Abstract

This is a thesis about how to model and predict stockmovements. The overall problem statement this thesis will try to solve is to study if a stockprice can be modelled and what models will give the best profitablity. It starts with some stocktheories about the efficent market hypothesis and some theories which will disclaim that theory by showing certain patterns in the stocks, which you can predict prices of. One of the theories, Farma&French, will be taken into the analysis to show how it performs when the theory is tried on real data, but that's later on in the thesis. When the foundation of the stocktheory is explained it go on with the theory of Deep Learning and neural network which will be made into a model too. These DeepLearning models are designed to make a model which will recognize certain patterns in all kinds of data. The main idea of that is to show some of the data, and then it has to predict the other part of the data and then increase learning. The third model is designed almost the same way as the neural network-model, and it uses the same data but with 180 more variables. These variables is 180 words collected from articles at Wall Street Journals and is made from 800.000 articles. The thoughts of this model is that, we always get distracted by news and when that happens it has to have an impact on the stockprice. The last model is a normal ARIMA-model from the thoery of econometrics. With all these six methods I model the data from 2001 to 2015 and forecast them to 2021. The results will show that the ARIMA-model can't model the price in the trainingset and neither the forecastpart. But the RNN- and Farma&French-model will give some good results regarding the forecast. When all the forecasts are made I make profitabilityanalysis to show how much return each model will make from 2015 to 2021. In the end of my thesis I give an assessment, in which I discuss whether my results is close to the efficient markethypothsis or not. Here I conclude that as I get a profit in five of the six models, the hypothesis is not as relevant to day as in the past. Today we can make econometric and other models, that can be used to predict future stockmovements, if we treat it with care. If we only treat it as it is, a model, we can take the conclusions from the models and use it but it can't stand alone.

Aktiemodellering

Speciale

Speciale

Skribent: Kasper Rønde Kongensgaard

Vejleder: Lasse Bork

Dato: Kl. 12 - 4. Juni 2021

Cand.oecon

Institut for Økonomi og Ledelse

Det samfundsvidenskabelige fakultet

Aalborg universitet

Institut for økonomi og ledelse
Det samfundsvidenskabelige Fakultet
Aalborg Universitet

Fibigerstræde 2
9220 Aalborg
Danmark
Phone: +45 99 409940

AALBORG UNIVERSITET
STUDENTERRAPPORT

Femte studieår ved
Det samfundsvidenskabelige Fakultet
Fibigerstræde 2
9220 Aalborg
Danmark
Phone: +45 99 409940

Titel:

Aktiemodellering

Projekt:

Speciale

Projektaflevering:

Kl. 12 - 4. Juni 2021

Projektgruppe:

Gruppe 11

Deltager:

Kasper Rønde Kongensgaard (20156060)

Vejleder:

Lasse Bork

Eksamenstidspunkt:

25. Juni 2021

Formål:

Dette speciale er lavet med det formål at kunne sammenligne almindelige metoder for aktiemodellering også deeplearning-metoder. Det er der kommet seks modeller ud af, som er vidt forskellige, både i data og metode. Der er blevet lavet analyser, som har haft til formål at vise sammenhængen mellem aktieprisen og almindelige karakteristika omkring markedet generelt og det specifikke omkring aktiers karakteristika, om deres profitabilitet og investeringer eksempelvis. Så er der lavet en almindelig tidsserie med en ARIMA, og disse er der så prøvet at finde alternativer til med en model, som modellerer ud fra pris og avisoverskrifter. Hvor meget fylder nyhedsstrømmen i aktiers fluktueringer? Den sidste model bruger deeplearning, i form af RNN- og LSTM-modeller. Formålet med disse vidt forskellige metoder er at lave en komparativ analyse og undersøge på forskellige parametre hvilken, der er mest profitabel.

Sidetal for opgaven: 88
Anslag med mellemrum: 130.076 (Antal sider: 55)
Afsluttet: Kl. 12 - 4. Juni 2021

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med r-script) må kun ske efter aftale med forfatteren.

Forord

Dette speciale er udarbejdet af en studerende på 10. semester, på Økonomi-uddannelsen, på Aalborg Universitet. Modellering af aktiesturkturer er det overordnede tema for projektet. Forudsætningerne for at læse rapporten er et vis kendskab til økonometrisk metode og analyse.
Der rettes stor tak til vejleder Lasse Bork for inspirerende vejledning og konstruktiv kritik.

Læsevejledning

Der vil gennem rapporten fremtræde kildehenvisninger, og disse vil være samlet i en kildeliste bagerst i rapporten. I rapporten anvendes der kildehenvisning efter APA-metoden, så i teksten refereres en kilde med [Efternavn, År]. Denne henvisning fører til kildelisten, hvor bøger er angivet med forfatter, titel, udgave og forlag, mens internetsider er angivet med forfatter, titel og dato. Figurer er nummereret i henhold til kapitel, dvs. den første figur i kapitel 7 har nummer 7.1, den anden har nummer 7.2 osv. Forklarende tekst til figurer findes under de givne figurer.
Projektet starter med en gennemgang af teori, der er anvendt i projektet, først med aktieteori, som skal skabe en grobund for, hvordan aktier skal analyseres og forståes. Kapitel 2 og 3 vil beskæftige sig med økonometriske og andre modelleringsværktøjer såsom RNN-modeller i området for DeepLearning. Derefter vil der i kapital 4 blive forklaret datagrundlaget for analyserne. I kapitel 5 vil der blive foretaget en analyse på baggrund af de resultater der er lavet, og hvordan modellerne er blevet til, og der vurderes de i forhold til hinanden ud fra forskellige økonometriske metrics. Til sidst vil der foreligge en vurdering af resultaterne i kapitel 6.

Kasper Rønde Kongensgaard

Indhold

Figurer

Tabeller

´

Indledning

Aktiemarkedet har altid været det vilde vesten. På et kort øjeblik kan man se prisen stige, og dermed få et afkast, man ikke havde regnet med, men man kan også lynhurtigt risikere, at alle de tanker man har haft om den fremtidige pris, har været forkerte, og dermed have tabt alle sine penge. Dermed er aktiemarkedet ubarmhjertigt. Kan man på nogle måder have nogle pejlemærker, når man ikke har den spåkugle, som havde gjort det hele nemmere? Er der nogle metoder, som kan give nogle gode estimeringer for, hvordan aktiekursen er om en dag, om en uge, om en måned eller måske med en længere forecasthorisont? Hvis man finder noget, der kan nærme sig forudsigelser, der kan give profitabilitet, vil det altid være en god tilføjelse til den almindelige aktieteori, som også har nogle gode værktøjer til at bestemme prisen. Det er det, som denne afhandling handler om; at prøve at finde alternative metoder, der kan estimere prisen, også holde det op mod en almindelig metode til at forecaste prisen, her tænkes eksempelvis på Farma&French og ARIMA. Med de nye metoder tilføjes der deep learning, som allerede bruges på aktier, men som er forholdsvist nyt, og derudover skal der ses på, om avisartiklers overskrifter fra Wall Street Journal kan tilføje noget forklaring til mainstream aktieforecast-metoder. Lige præcis de nye metoder er nogle, som er den primære motivation til at skrive dette speciale. Når DeepLearning i fremtiden, og på sin vis allerede, skal være drivkraften bag selvkørende biler mv. så vil det være mærkeligt, hvis de ikke også kan modellere et tilfredsstillende forecast på aktier. Den anden metode er ideen om, at vi i dag bliver påvirket af medierne i alle mulige retninger hver dag. Det er medierne, som sætter dagsordenen til en vis grad for, hvilke emner der florerer på dagligdagens emner i samfundet. Derfor ville det også være mærkeligt, hvis det ikke også påvirker investorers investeringsstrategier fra dag til dag. I disse modeller afgrænses der til, at såsom dividende og transaktionsomkostninger ikke er med i specialet. Projektet er bygget op om at finde gode muligheder for at forecaste aktiepriser, og se hvilke metoder der er bedst til lige netop det, og deraf kommer følgende problemformulering:

Problemformulering: *Kan aktiepriser modelleres, og hvilke modeller, af lineære og ikke-lineære, giver de mest præcise forecast, der kan give den bedste profitabilitet?*

Metode

Dette projekt forsøger at lave en komparativ analyse af forskellige økonometriske analyser ved både lineære og ikke-lineære metoder for at bestemme en fremtidig aktiepris. Problemformuleringens ontologiske antagelse er, at der ved forskellige modelleringsmetoder kan dannes grobund for forecasts af aktiepriser, der kan bruges til profitabilitet på aktiemarkedet. Epistemologien sker gennem flere metoder til analyse af tidsserie, og derefter en vurdering af disse analyser ved forskellige nøgletal/metrics, der kan bestemme den egentlige performance for de forskellige analyser. Men det er svært at lave sådanne analyser, så de virker, da aktiemarkedet er styret af de generelle, universelle mekanismer, som styrer individernes økonomiske handlinger på baggrund af, at der antages, at individerne er egoister, der maksimerer deres egen individuelle nytte. Det mente Menger, at når det er fastlagt, så kan vi bedre modellere og estimere, hvilke mekanismer der vil komme (Ingemann, 2014, p. 50). Dette speciale er et empirisk projekt, som bruger en kvantitativ metode til at lave en deduktiv analyse af nogle forskellige teorier, der testes på virkeligheden, og dermed undersøge om de forskellige mekanismer i teorierne kan bruges til modellering af aktier. Derfor er genstandsfeltet de forskellige modelleringsmetoder, hvor statistik og aktieteori er fagniveauet, og at aktiemarkedet kan modelleres ved forskellige metoder, er specialets grand theory. Ud fra Poppers tre-trins-metode (Jespersen, 2009, p. 54) ses der i verden 1, det reale niveau, på, hvad der sker i virkeligheden, hvor der bliver beskrevet forskellige mekanismer, som der er blevet opdaget for aktiemarkedet, eksempelvis Januar-effekten og andre forskellige mekanismer, som er blevet opdaget, og som derfor er udvandet sammen med publiceringen. Derudover bliver der vist, hvordan S&P500-indekset observeres i perioden. Herefter går vi over i det analytiske niveau, verden 2, hvor der analyseres med forskellige metoder for, hvordan aktiemarkedet kan modelleres (i det fatuelle stratum) gennem resultater ved statistiske metoder, og laver hermed empiriske test mellem resultaterne og virkeligheden (i det empiriske stratum). Hele tiden gåes der frem og tilbage gennem det erkendelsesmæssige slør, indtil der nåes endelige resultater. I verden 3 ses der så på, om der objektivt kan drages konklusioner mellem virkeligheden og teorien gennem de empiriske test, og om de kan bidrage til teoriernes gyldighedsområde. Kan der skabes profitabilitet på aktier med disse metoder.

KAPITEL 1

Aktieteori

I dette afsnit vil der blive præsenteret forskellige teorier for aktier, som skal give grobund til en vis forståelse for, hvordan aktier agerer indenfor forskellige mønstre og nøgletal. Der startes med at fortælle om teorien om den efficiente markedshypotese, og hvor der så kommer teorier, der argumenterer i mod den efficiente markedshypotese for så at komme med nogle sammenhænge, som forskellige parametre kan være med til at forklare aktieprisen for en given aktie.

1.1 Den efficiente markedshypotese

I 1970 kom Fama Eugene med en ny teori om aktiemarkedet, som hedder den efficiente markedshypotese. Ifølge den skal et marked være efficient, hvis virksomheder skal kunne lave investeringsbeslutninger for produktionen, og investorer kan vælge mellem aktier, der repræsenterer ejerskab af virksomheden under antagelse af, at aktierne hele tiden repræsenterer fuldt ud alt information, der er tilgængelig. Ud fra dette kommer han med følgende definition: (Fama, 1970, p. 384)

Definition 1: Effcient

Et marked, hvor priserne altid fuldt ud reflekterer alt tilgængelig information, er efficient

Det vil altså sige, at det er umuligt at lave økonomisk profit ved at handle på basis af den information, der er til rådighed (Granger, 1992, p. 3). Men (Fama, 1970, p. 384) stiller selv spørgsmålet om, hvad "fuldt ud reflekterer" betyder. Her vil han prøve at finde ligevægtsprisen for en given aktie, hvor ligevægten i markedet vil blive set som det forventede afkast. På basis af den tilgængelige information er ligevægtsafkastet på en given aktie en funktion af dens risiko. Problemet er, at der er flere forskellige definitioner af risiko. Alle der har en tanke om aktier, har deres egen definition af risiko, og når alt tilgængelig information

på tidspunkt t er inkorporeret i aktien på tidspunkt t, kan informationen ikke bruges til at forecaste en akties værdi i fremtiden. Hermed siger han, at gennemsnittet af afkastet z på aktie j til tiden t+1 på basis af informationen Φ_t er lig 0, og dermed et "fair game", se ligning 1.1 (Fama, 1970, 384-385).

$$E(z_{j,t+1}|\Phi_t) = 0 \tag{1.1}$$

I det tidlige stadie af denne model var der tanker om, at det at en pris fuldt ud reflekterede alt tilgængelig information, var antagelsen om, at prisændringerne (selv på kun en periode) er uafhængige, og dermed var ændringerne også set som identisk fordelt. Disse to antagelser medfører, at en akties sti følger en random walk. Så gennemsnittet af det forventede afkast på tidspunkt t+1 er uafhængig af informationen på tidspunkt t, og dermed er hele fordelingen uafhængig af informationen på tidspunkt t. Der bliver argumenteret for, at random walk-modellen er en udvidelse af fair game, da den model bare siger, at markedsligevægten kommer frem ved det forventede afkast, og dermed kommer det af en stokastisk process, som genererer afkastene. En random walk kommer ind i modellen, når en investor ser udvikling, og der hele tiden kommer ny information, som hele tiden prøver at finde ligevægt, og hvor afkastfordelingen gentager sig selv gennem tiden (Fama, 1970, p. 386-387).

1.2 Teorier imod den efficiente markedshypotese

I afsnittet om den efficiente markedshypotese, afsnit 1.1, bliver markedet drevet af alt tilgængelig information på rekordtid, men der er dog nogle teorier om mønstre på aktiemarkedet, der går i mod den efficiente markedshypotese, som bliver publiceret, og nogle af dem bliver der redegjort for i dette afsnit.

1.2.1 Regimeskiftende modeller

Hvis en metode er fundet, vil en akademiker nok hellere få profit ud af det end at publisere ideen. Men (Granger, 1992) mente dog, at den eneste måde at tjene på aktiemarkedet var at skrive en bog om det. Han giver nogle eksempler på nogle modeller, som virkede godt, men hvor nogle af dem forsvandt, når først alle kendte dem (Granger, 1992, p. 4). Dem han taler om, er:

- Evne til forecast af aktier med lav volatilitet
- Den mindste markedsværdi
- Sæsoneffekter
- Prisændringer

Den første model han gengiver, er evnen til forecast af aktier med lav volatilitet. Her er der taget et ugentligt afkast fra Standard and Poor 500-indekset for perioden 1946-1985. Han tager volatiliteten af aktien som variable, og her deles volatiliteten op i kvartiler, hvor der bliver taget 1/5 af aktierne, som er i den laveste kvartilrække. Disse aktier blev delt op i to lige store dele, hvor den første halvdel blev brugt til at estimere modellen, og den sidste halvdel blev brugt til validering af resultatet. Dette gav en forbedring af modellen på $3,1\%$. Ingen af de andre kvartilsæt fandt nogen forbedring, hvor de bare blev til en random walk. Men senere forskning af samme forsker af modellen har ikke fundet denne evne (Granger, 1992, 4-5).

Den næste model, den mindste markedsværdi, blev brugt på aktier i New York eller American Stock Exchange i perioden 1951 til 1986. Her er der blevet lavet porteføljer baseret på markedsværdien, i denne model "size", og fortjeneste til pris, E/P-ratioen, og derefter blev der beregnet det månedlige afkast. Hvert år på datoen d. 31/3 blev alle aktier rangeret efter markedsværdien, og de 10% lavest rangerede kom ind i den mindste portefølje, de næste 10% i den næste portefølje og så fremdeles. Porteføljen blev ændret pr. år, og der blev beregnet et gennemsnitligt månedligt afkast. Det viste sig, at de porteføljer med de aktier, der var mindst i markedsværdi, havde det højeste gennemsnitlige afkast, og de porteføljer med den højeste E/P havde et bedre afkast end dem med de laveste E/P (Granger, 1992, p. 5).

Den tredje model, der bliver nævnt, er sæsoneffekter, hvor Januar-effekten bliver fremhævet. Her er det igen de virksomheder, der har den mindste markedsværdi og højeste E/P, der tages udgangepunkt i. I januar fik disse porteføljer et afkast på $7,46\%$ i gennemsnit men kun et gennemsnitligt afkast på $1,39\%$ i de andre måneder. I perioden fra 1926 til 1986 var der kun et tab i januar, for disse virksomheder, i syv ud af de mulige 61 år (Granger, 1992, p. 6). Dette kan også ses med en mandags-effekt, hvor afkastene er højere om mandagen, og når en måned skifter til en anden måned, er der også høje afkast (Malkiel, 2003, p. 64).

Den fjerde model forklarer, at der er en tendens til, at retningen i afkast for aktierne er invers over tid. Aktier, der gør det godt i den ene periode, har en tendens til at gøre det dårligt i den næste og omvendt for aktier med omvendt fortegn. Her blev der valgt 200 tilfældige handledage i perioden januar 1974 til januar 1984. Her blev der noteret de aktier med det største procentvise tab, og over de næste 10 handledage, blev der tjent 3.6% på dem, og dem der havde klaret sig godt i første periode, kom ud med et tab på $1,8\%$ over de 10 dage. Dette er forsøgt en del gange, og de viser en vis stabilitet (Granger, 1992, p. 6).

1.2.2 Mønstre i aktiekurser

I starten var (Malkiel, 2003) enig i betragtningen om, at aktiemarkedet ikke var til at forudsige. Som han giver som eksempel i en af hans tidlige bøger, så kan en chimpanse med bind for øjnene, der kaster med dartpile mod de aktier, der skulle vælges i sin portefølje, klarer sig lige så godt som eksperternes portefølje. Men i starten af det 21. århundrede blev hypotesen om det efficiente marked mindre brugbar. Mange økonomer og statistikere begyndte at tro på, at aktiemarkedet delvist er forudsigelig, hvis man tilførte nogle psykologiske og adfærdsmæssige elementer i porteføljevalget. Her bliver der vist nogle af de modeller, der angriber den efficiente markedshypotese. Dog slutter han af med nogle undersøgelser om de bobler, der har været i tidens løb, og konkluderer, at markedet måske er mere efficientt, end nylige undersøgelser vil have os til at tro. Men disse konklusioner vil ikke være i dette afsnit (Malkiel, 2003, p. 59-60).

Kortsigtet momentum

Når det gælder afkast på kort sigt, er der blevet lavet nogle undersøgelser for seriekorrelation for afkastet, som disse undersøgelser finder ud af, ikke er 0. For mange af aktierne, bevæger de sig i den samme retning, og derfor kan de afvise hypotesen om, at aktiepriser bevæger sig som random walks. Der er også fundet undersøgelser om, at ikke-parametriske modeller i nogle tilfælde kan være med til at forudsige en aktiekurs (Malkiel, 2003, p. 61).Hvis økonomer og psykologer bruges i adfærdsøkonomi, som tidligere nævnt, vil sådanne kortsigtede momentummer være konsistent med psykologiske feedback-mekanismer. Der bliver givet to forklaringer, hvor den ene er fra psykologer, og som handler om, at uanset om vi er enige i udviklingen af kursen eller ej, så følger vi trenden med en bandwagon-effekt. Den anden kommer fra adfærdsforskere, som har en anden forklaring på kortsigtede momentummer, og som mener, at der er en tendens til, at investorer ikke reagerer nok overfor ny information. Hvis hele indholdet af nyheden kun er taget for en lille tidsperiode, vil aktiepriserne følge nyhedens konklusioner, uanset om den er god eller dårlig, og nyheden vil dermed kunne se sig afspejle i aktiekursen. Alt dette er momentum-investorer, som handler ud fra, hvad der sker i øjeblikket, men som ikke altid reagerer kraftigt nok på nyheden, mener adfærdsforskere. Men studier viser, at disse momentum-investorer ikke får de store afkast, da de betaler en stor del af det til transaktionsomkostninger, og derfor tror mange investorer også, at buy-and-hold-strategien stadig skal beholdes. Selv i perioder hvor denne momentumstrategi burde virke, gik det værre for momentum-investorer end for dem, der har buy-and-hold-strategien (Malkiel, 2003, p. 61-62). Disse kortsigtede momentummer forsvinder ofte, efter de er blevet publiceret. Januar-effekten, eksempelvis, forsvandt kort tid efter, den blev publicerret, fordi

når de ser ud til at udfordre hypotesen om det efficiente marked, bliver ideen udvandet af, at alle nu bruger ideen (Malkiel, 2003, p. 63).

Langsigtede afkastændringer

Når aktier har kørt i dage og uger, kommer der en negativ og positiv seriekorrelation, som går i mod hypotesen om markedsefficientiteten. Der bliver argumenteret for, at dette er tegn på overreaktion. Investorer giver bølger af enten for meget optimisme eller for meget pessimisme, der får priserne til at gå fra sine normale værdier for så senere at konvergere til sit normale gennemsnitsniveau. Dette kan ses som en overreaktion fra investorerne, hvor de hele tiden har en overdreven selvsikkerhed for egne evner til at forecaste prisen. Dette kører efter en contrarian-strategi, hvor man køber de aktier, der på tidspunktet bløder, og sælger dem, der har klaret sig godt, som kan ses som modellen for prisændringer i afsnit 1.2.1 (Malkiel, 2003, p. 63).
Nogle andre studier viser, at volatiliteten for aktier kan ses med volatiliteten for renter, og at tendensen for renter konvergerer mod gennemsnittet. Når renter går op, går prisen for både obligationer og aktier ned og omvendt. Derfor kan dette bruges til at sige, at når renterne konvergerer mod deres gennemsnit, vil dette mønster også generere aktier til at konvergere mod deres gennemsnit (Malkiel, 2003, p. 63-64).

1.2.3 Forudsigelige mønstre ud fra værdibaserede parametre

Der er blevet konstateret et mønster for, at det fremtidige aktieafkast kan forudsiges ved værdibaserede parametre, som er dividendeudbytte og price-earnings (P/E-ratio).

Fremtidig afkast ud fra dividendeudbytte

Der er blevet undersøgt, hvorvidt dividendeudbyttet (som er ratioen af hvor meget dividenden udgør af aktien) kan bruges til at forecaste fremtidig afkast. Undersøgelser har vist, at, afhængig af forecasthorizonten, op til 40% af variansen for det fremtidige afkast for hele markedet kan forklares ved dividendeudbyttet for markedsindekset. Undersøgelser har vist, at aktier med relativt høje og relativt lave dividender giver et højere afkast end dem midt i mellem. Men der bliver argumenteret for, at disse mønstre ikke har kunnet give noget brugbart til forecast siden midten af 1980'erne, og er derfor ikke brugbare i dag. En mulig forklaring skal findes ved, at virksomheder hellere vil lave tilbagekøb af aktier end at give dividende. Senere undersøgelser har også vist, at søger man afkast via dividendestrategier, har det givet et dårligere resultat i forhold til markedet generelt (Malkiel, 2003, p. 65).

Fremtidig afkast ud fra Price-Earning-ratio

Her er der også blevet undersøgt, hvorvidt store afkast kan tjenes på grupper af aktier med relativt lave P/E-ratios. Artikler har vist, at også denne ratio har vist at kunne forklare 40% fra variansen for fremtidig afkast. De konkluderer, at det har være forudsigeligt til en vis grad i før i tidenn. Forklaringsgraden for P/E-ratios er dog faldet, ligesom dividende-forklaringen, men i stedet for ned til 3%, som ved dividende, er P/E kun faldet til omkring 20%. Der har den lagt nogenlunde stabilt i de perioder, der er blevet undersøgt for. Disse resultater ses i artiklen som, at forecast af aktieafkast på denne metode er nogenlunde stabil (Malkiel, 2003, p. 66-67).

1.3 Fama&French - Fundamental analyse

I dette afsnit vil der blive forklaret, hvilke parametre (Fama and French, 2014) mener, der skal bruges for at kunne estimere en aktiekurs. I (Fama and French, 2014) bygger de videre på deres eget arbejde fra 1993 om tre-faktors model, som er bygget videre fra CapM. I dette afsnit vil der blive forklaret de fem parametre. De tre modeller fra (Fama and French, 2014) vises i ligning 1.2, 1.3 og 1.4.

$$R_{it} - R_{Ft} = a_i + b_i(R_{Mt} - R_{Ft}) + e_{it} \quad (1.2)$$

$$R_{it} - R_{Ft} = a_i + b_i(R_{Mt} - R_{Ft}) + s_i SMB_t + h_i HML_t + e_{it} \quad (1.3)$$

$$R_{it} - R_{Ft} = a_i + b_i(R_{Mt} - R_{Ft}) + s_i SMB_t + h_i HML_t + r_i RMW_t + c_i CMA_t + e_{it} \quad (1.4)$$

Ligning 1.2 er den almindelige CapM, som er afkastet for aktiv eller porteføljе i minus afkastet for det risikofrie aktiv, som er afkastet på den en-månedlige amerikanske statsobligation (Fama and French, 2014). Ligning 1.2 har så en konstant a plus en konstant b, som er ganget på markedsafkastet minus afkastet på det risikofrie afkast, hvor der så er tilføjet et fejlled. Det vil sige, at det forventede afkast er lig med den risikofrie rente plus betydningen af markedets risikopræmie. b viser dermed, hvor meget risiko investeringen vil tilføje til markedet. Hvis aktien er mere risikobetonet end markedet, vil b>0. Hvis b<0, er aktien mindre risikobetonet end markedet (Kenton, 2021).
Den næste ligning, 1.3, er lavet af (Fama and French, 1993) som en videreudvikling af CapM med to ekstra faktorer, SMB (SmallMinusBig) og HML (HighMinusLow). Til sidst er der fem-faktorsmodellen, som også er lavet af Fama og French (Fama and French, 2014) men med tilføjelse af to ekstra variable, RMW (Forskellen mellem afkastet for porteføljer med robust og svag profitabilitet) og CMA (Forskellen mellem afkastet på proteføljer med aktier med lave og høje investeringsvirksomheder).

Den første variabel efter CapM er SMB, som er de virksomheder med en lille markedskap minus dem med en stor. Markedskap er antallet af aktier udestående gange prisen på en aktie på et givet tidspunkt. De opdeles i to kategorier, store og små markedskaps, som deles ved 50%. Resten af variablerne deles i tre ved grænserne 30% og 70%. Her deles SMB op i 12 grupper, hvor der er porteføljer med seks størrelser OP (Overskuddet divideret med egenkapitalen). Så er der porteføljer med seks størrelser INV (Årlig stigning i totale aktiver). Til sidst deles de op i seks grupper af B/M, som er værdien af virksomheden i forhold til dens markedskap, hvor værdien af virksomheden beregnes som dens aktiver over dens forpligtigelser. Det hele ses i tabel 1.1. Ud fra tabel 1.1 kan der nu laves de variable, som

Tabel 1.1: Opdeling af variable til Farma&French

Små	Book-to-market (B/M)	Høj (SH)
		Neutral (SN)
		Lav (SL)
	Profitabilitet (OP)	Robust (SR)
		Neutral (SN)
		Svag (SW)
	Investeringer (INV)	Konservative (SC)
		Neutral (SN)
		Aggressive (SA)
Store	Book-to-market (B/M)	Høj (BH)
		Neutral (BN)
		Lav (BL)
	Profitabilitet (OP)	Robust (BR)
		Neutral (BN)
		Svag (BW)
	Investeringer (INV)	Konservative (BC)
		Neutral (BN)
		Aggressive (BA)

er nødvendige til at lave modellen fra (Fama and French, 2014). Hvordan disse beregninger så laves, kan ses i ligningerne 1.5 til 1.11 (Erdinç, 2017).

$$SMB_{B/M} = (SH + SN + SL)/3 - (BH + BN + BL)/3 \tag{1.5}$$

$$SMB_{OP} = (SR + SN + SW)/3 - (BR + BN + BW)/3 \tag{1.6}$$

$$SMB_{Inv} = (SC + SN + SA)/3 - (BC + BN + BA)/3 \tag{1.7}$$

$$SMB = (SMB_{B/M} + SMB_{OP} + SMB_{Inv})/3 \tag{1.8}$$

$$HML = (SH + BH)/2 - (SL + BL)/2 \tag{1.9}$$

$$RMW = (SR + BR)/2 - (SA + BA)/2 \tag{1.10}$$

$$CMA = (SC + BC)/2 + (SA + BA)/2 \tag{1.11}$$

1.4 Nøgletal

Når en investor skal vide, om en aktie er værd at investere i, kan der ses på, hvordan virksomheden i det hele taget klarer sig indtjeningsmæssigt, og om investorerne får noget af overskuddet. Derfor er det en fordel at kende nogle forskellige nøgletal, som viser disse parametre, og her skrives der om nøgletallene dividendeudbytte og P/E-ratio.

1.4.1 Dividendeudbytte

På aktiemarkedet ses dette ofte som dividendeudbytte, som er forholdet mellem, hvad en virksomhed hvert år betaler i dividende ud af den aktuelle aktiepris, som opgøres i procent. Det bliver brugt af investorer, der vil have en større sikkerhed for deres investering. Hvis to virksomheder betaler det samme i dividende hvert år, udregnet i kroner, og den første virksomheds aktiekurs er dobbelt så høj som den anden, vil den enkelte investor tage den virksomhed med den laveste kurs, for der får investoreren en højere procentuel dividende ud af hver aktie. Dermed får man en højere dividende for hver krone, der er investeret. Der er dog en ulempe ved det, for for hver krone, der bliver givet til dividender, jo færre investeringer kan virksomheden lave, og dermed kan det gøre aktiekursen lavere på lang sigt, fordi man ikke har lavet de nødvendige investeringer til den fremtidige konkurrence.
Men der kan også være spekulation fra virksomhedens side ved at betale høje dividender. En høj dividende kan være et udtryk for en aktiepris' "tankning". Udbyttet vil ofte stige, når prisen falder, som er set som "værdifælden". Det kan være, at virksomheder lider, og betaler højere dividende for, at markedet forhøjer prisen på aktien, så derfor skal der altid tjekkes motivet for at udbetale dividender (Little and Mansa, 2020).

1.4.2 Price-to-earnings-ratio og Book-to-market

Price Earnings Ratio (P/E Ratio) er forholdet mellem aktieprisen og fortjeneste pr. aktie, som giver en indikation af, hvad værdien af virksomheden er. Det er en vigtig parameter, da det giver information om, hvor profitabel virksomheden er nu og i fremtiden. Men som en enkelt værdi for en enkelt virksomhed er den ikke meget værd, da den skal sammenlignes med andre selskaber fra samme industri eller med værdierne i de foregående år i samme virksomhed.
Hvordan P/E-ratio'en beregnes, kan ses i ligning 1.12.

$$P/E = \frac{Aktiepris}{Indtjening\ pr\ aktie} \tag{1.12}$$

Fortjenesten er det, der er tjent på en aktie over 12 måneder. P/E-ratio er et nøgletal, som kan bruges til at bestemme, om der betales en fair pris for aktien i forhold til andre

virksomheder i samme branche. En høj P/E er givet ved en vækstaktie, som indikerer en positiv fremtidig indtjening, og investorer har dermed højere forventninger til denne, men de har også ofte en højere volatilitet, og kan ses som mere risikabel, ligesom den også kan være vurderet for højt. Deraf kan en lav P/E ses som en værdiaktie, som også ses som en undervurderet aktie, fordi den er lavere end konkurrenternes. Denne forkerte pris vil hurtigt gøre, at investorer køber den, inden markedet korregerer for den lavere pris, og dermed er der tjent penge (Corporate Finance Institute, nd). Værdiaktier har en tendens til at give et højere afkast end en vækstaktie, og det mener adfærdsøkonomer, kommer af, at investorer er for selvsikre ved deres egne evner, og ender med at betale for meget for vækstaktier. Et nøgletal, som fortæller det samme som P/E-ratio er price-to-book, og ses i ligning 1.13. En price-to-book-værdi over en ses som en aktie, der er overvurderet i forhold til bogværdien (Malkiel, 2003, 68-69).

$$^{P}/_{B} - ratio = \frac{Markedspris\ pr.\ aktie}{Bogværdi\ pr.\ aktie} \tag{1.13}$$

$$Bogværdi\ pr.\ aktie = \frac{Aktiver - Forpligtelser}{Antal\ aktier} \tag{1.14}$$

(Mcclure, 2021)

KAPITEL 2

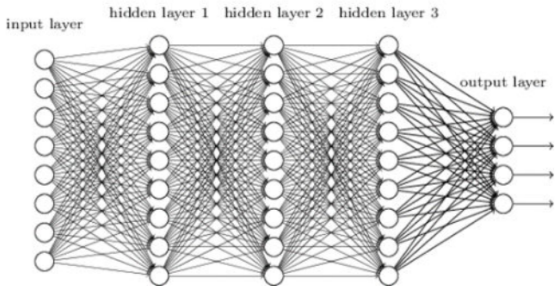
Neurale netværk

I dette afsnit vil der blive forklaret, hvad neurale netværk er, hvordan det opererer, og herefter vil der blive forklaret mere dybdegående, hvad de forskellige elementer gør. Forklaringerne vil i store træk være matematisk, og der bliver forklaret hvordan algoritmen og tabsfunktionen fungerer inde bag systemet. Til sidst vil der blive forklaret, hvordan det neurale netværk kan forbedres.
Neurale netværk er en teknik, som bruges til at finde et system i det data, der arbejdes med. Det kan bruges til at classificere eller forecaste nyt data, som modellerne aldrig har set før ud fra lignende data, den er blevet trænet på. Den er lavet til følgende problemstillinger:

- Binær klassificering
- Multiclass-klassificering
- Regression

(Chollet and Allaire, 2018a, p. 50-51)

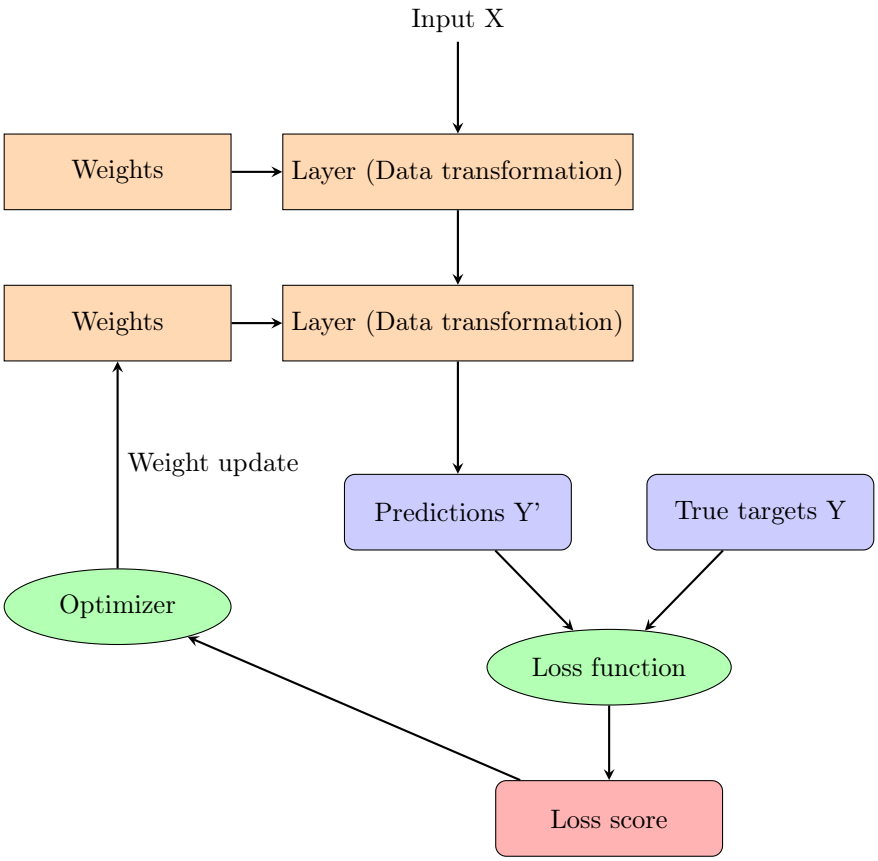
2.1 Det neurale netværks anatomi



Figur 2.1: Deep neural netværk - Hentet fra (Kyrykovych, 2020)

Det neurale netværk er inspireret af den menneskelige hjernes neuroner. Den indeholder et input-layer, hvor data'et, der skal analyseres på, kommer ind i modellen, og hvor data'et

så bliver analyseret i de hidden-layers, der er i modellen for at lave nogle vægte, der gør, at det klassificeres rigtigt i modellens output-layer (Kyrykovych, 2020).
Hver hidden-layer indeholder små noder, som sammenlignes med neuronerne i en menneskehjerne. Jo flere hidden-layers, der er i modellen, jo mere kompleks, deep, bliver modellen, og dermed mere og mere som vores hjerne. Det neurale netværk er designet til at genkende stemmestyring, lyd og grafik, lave en ekspertvurdering over bestemte emner og en hel masse andet, der kræver forudsigelse, og kreativ og analytisk tænkning (Kyrykovych, 2020).
For at forstå hvordan den gør, som den gør, se figur 2.2. Modellen får givet noget data som



Figur 2.2: Forhold mellem de forskellige elementer i en Deep Learning-model - fra (Chollet and Allaire, 2018a, p. 51)

input, som så bliver givet til modellens layers, som arbejder sammen for at give nogle vægte, der får modellen til at forstå mønstrene i data'et. Det vil efterfølgende blive lavet til en forudsigelse Y', som bliver sammenlignet med det rigtige data i loss-funktionen, som producerer en loss-værdi. Loss-værdien vil blive brugt i optimizeren til at opdatere vægtene, så de bedre repræsenterer data'et. Denne proces vil stoppe, når loss-værdien ikke kan minimeres mere. Så modellens loss-funktion er størrelsen, der skal minimeres under træningen, og optimizeren fastsætter metoden for, hvordan netværket bliver opdateret ud fra loss-funktionen

(Chollet and Allaire, 2018a, p. 51-52).

2.2 Systemet bag det neurale netværk

I det foregående afsnit er der blevet forklaret meget kort, hvordan et neuralt netværk fungerer, men for det kan bruges, skal matematikken og systemet bag være defineret. I dette afsnit bliver der forklaret et netværk, perceptron, der har et binært output, og hvor der er input x_j for det j'te input, og hvor vigtig hver input er for outputtet, vægten w_j for det j'te input. Hvordan det beregnes, ses i ligning 2.1, hvor det første udtræk er i ligningsform og andet udtryk er i matrix-form.

$$output = \begin{cases} 0 & hvis \ \sum_j w_j x_j \leq Grænseværdi \\ 1 & hvis \ \sum_j w_j x_j > Grænseværdi \end{cases} = \begin{cases} 0 & hvis \ w * x + b \leq 0 \\ 1 & hvis \ w * x + b > 0 \end{cases} \tag{2.1}$$

I andet udtryk i ligning 2.1 ses det samme udtryk i matrix-form som det første, men grænseværdien er kommet om på den anden side, som et bias, $b \equiv -Grænseværdi$, hvor b er et tal for hvor let det er for den enkelte perceptron at give dens output. Er b positiv hælder den mod 1, og hvis den er negativ hælder den mod 0.
I de følgende afsnit vil der blive forklaret, hvordan det fungerer med algoritmen backpropagation, der skal få vægte og bias til at minimere tabsfunktionen C med ændring af hele netværket ud fra Sigmoid-funktionen (Nielsen, 2015, Chapter 1).

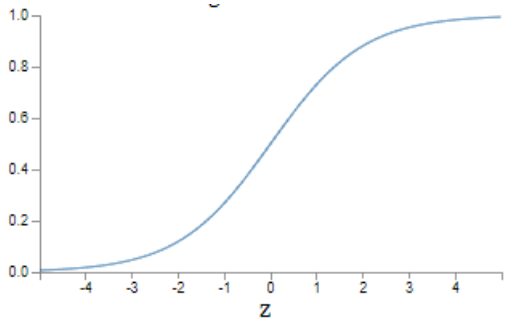
2.2.1 Sigmoid og Softmax

Sigmoid

Problemet som Sigmoid-funktionen vil løse, er, at når der normalvis bliver ændret Δw til w vil hele netværket ændres, og outputtet for alle de andre inputs vil også ændres. Det gør det svært at ændre netværket en lille smule uden at ændre den del af netværket, der allerede minimerer C. Det kan gøres ved at definere en ny neuron, en Sigmoid neuron, som er bedre til at få outputtet til at ændre sig mindre ved den samme ændring af nogle få vægte og bias et sted i netværket. I stedet for, at output kun kan være 0 eller 1, altså binært, så kan Sigmoid tage en værdi mellem 0 og 1. Det gør, at hver led i netværket har nuancer for, hvilket output den producerer, i stedet for den kun producerer binært. Sigmoid-funktionen kan defineres som ligning 2.2.

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-\sum_j w_j x_j - b}} \tag{2.2}$$

I figur 2.3 ses, hvordan Sigmoid-funktionen agerer grafisk. Det giver et bedre overblik over, hvad ligning 2.2 gør. Når z er et stort positivt tal, så er $e^{-z} \approx 0$ og $\sigma \approx 1$. Hvis $e^{-z} \to \infty$ så



Figur 2.3: Sigmoid-funktion

$\sigma(z) \approx 0$. Så når z er meget negativ eller meget positiv, agerer den som en perception. Det er kun, når z er lidt i midten, at der er forskel fra perception og Sigmoid, men det gør, at nuancerne er til stede. Så funktionen i Sigmoid gør, at en lille ændring i w, Δw_j, og i bias, Δb, vil kun give små ændringer i output, som ses i ligning 2.3 (Nielsen, 2015, Chapter 1).

$$\Delta output \approx \sum_j \frac{\partial output}{\partial w_j} \Delta w_j + \frac{\partial output}{\partial b} \Delta b \tag{2.3}$$

Softmax

Her defineres en anden type output-layer for det neurale netværk, Softmax-funktionen. I ligning 2.4 ses Softmax-funktionen til z_j^L med aktiveringen a_j^L af j's outputneuron, hvor $z_j^L = \sum_k w_{jk}^L a_k^{L-1} + b_j^L$.

$$a_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}} \tag{2.4}$$

Ligning 2.4 er en output-funktion, som ikke kun kan tage to men flere output-neuroner, og derfor gør det, at outputtet ikke længere er binært. Hvis der er j output-neuroner skal der gælde, at disse aktiveringer summerer til 1 som i ligning 2.5.

$$\sum_j a_j^L = \frac{\sum_j e^{z_j^L}}{\sum_k e^{z_k^L}} = 1 \tag{2.5}$$

Ligning 2.5 siger altså, at stiger den ene aktivering a_j^L, så skal de andre falde, og ligning 2.5 holder også for alle andre aktiveringer. Grundet eksponentialet sikrer det også, at funktionen er positiv. Så når dette Softmax-layer er et sæt af positive tal, der summerer til 1, kan det ses som en sandsynlighedsfordeling. Derfor kan output-aktiveringen a_j^L være et estimat for en sandsynlighed for, at inputtet x, der klassificeres som j, er rigtigt klassificeret.
Nu hvor Softmax er defineret, kan der nu defineres den tabs-funktion, som er anderledes fra Sigmoid-funktionen, hvor tabs-funktionen for den Sigmoid vises i afsnit 2.2.3. Tabs-funktionen for Softmax er her defineret som en log-likelihood-tabsfunktion. Her vil x være et

træningsinput til netværket, og y vil være det valgte output til x. Dermed kan log-likelihood-tabsfunktionen defineres som ligning 2.6.

$$C \equiv -ln(a_y^L) \tag{2.6}$$

Hvis ligning 2.6 har den største sandsynlighed for det rigtige output y, vil sandsynligheden a_y^L være tæt på 1 og tabsfuntionen $-ln(a_y^L)$ vil være lille, mens det vil forholde sig omvendt, hvis den estimerer forkert (Nielsen, 2015, Chapter 3).

2.2.2 Backpropagation

Backpropagation er en algoritme, som skal minimere tabs-funktionen C, som beskrives i afsnit 2.2.3. Den minimere C ved at have et input x_j, hvor der ses på det sidste layer l-1 før netværkets outputlayer. I det layer ændres aktiveringen a ved en lille ændring i vægten w for at ændre aktiveringen i output-layer for så at få outputtet klassificeret rigtigt, og dermed minimeret C. Når det sidste layer l-1 har fået de værdier, der minimerer C mest, laves den samme procedure for layer l-2 hele vejen tilbage til input-layer. Denne procedure, kaldet backpropagation, kan skrives som ligning 2.7, som giver ændringerne i C ved ændringerne i aktiveringerne, $a_j^l, a_q^{l+1}, ..., a_n^{L-1}, a_m^L$, gennem hele netværket med den specifikke rute.

$$\Delta C \approx \frac{\partial C}{\partial a_m^L}\frac{\partial a_m^L}{\partial a_n^{L-1}}\frac{\partial a_n^{L-1}}{\partial a_p^{L-2}}...\frac{\partial a_q^{l+1}}{\partial a_j^l}\frac{\partial a_j^l}{\partial w_{jk}^l}\Delta w_{jk}^l \tag{2.7}$$

For at klassificere ruten i ligning 2.7 for alle ruter, bliver alle mulige valg for de mellemliggende neuroner på ruten nu summeret i ligning 2.8.

$$\Delta C \approx \sum_{mnp...q}\frac{\partial C}{\partial a_m^L}\frac{\partial a_m^L}{\partial a_n^{L-1}}\frac{\partial a_n^{L-1}}{\partial a_p^{L-2}}...\frac{\partial a_q^{l+1}}{\partial a_j^l}\frac{\partial a_j^l}{\partial w_{jk}^l}\Delta w_{jk}^l \tag{2.8}$$

Herefter beregnes raten for en andring i C med respekt for en vægt w_{jk}^l i netværket i ligning 2.8. Det som ligning 2.8. siger, er, at hver sti mellem to neuroner er associeret med en ratefaktor, som er den partielle derivat af en neurons aktivering med respekt for den andens neurons aktivering. Ratefaktoreren for den ene sti er så produktet af ratefaktorerne gennem stien. Den totale ændringsrate for $^{\partial C}/_{\partial w_{jk}^l}$ er så summen af ratefaktorerne for alle stier fra den bestemte vægt til C. Denne rate kan beregnes som i ligning 2.9 (Nielsen, 2015, Chapter 2).

$$\frac{\partial C}{\partial w_{jk}^l} = \sum_{mnp...q}\frac{\partial C}{\partial a_m^L}\frac{\partial a_m^L}{\partial a_n^{L-1}}\frac{\partial a_n^{L-1}}{\partial a_p^{L-2}}...\frac{\partial a_q^{l+1}}{\partial a_j^l}\frac{\partial a_j^l}{\partial w_{jk}^l} \tag{2.9}$$

De fire antagelser bag backpropagation

For at bestemme de fire antagelser, vil der blive defineret fejlen i j's neuron i l's layer, hvor backpropagation vil relatere δ_j^l til $^{\partial C}/_{\partial w_{jk}^l}$ og $^{\partial C}/_{\partial b_j^l}$. Proceduren er, at der laves en lille ændring Δz_j^l til neuronens vægtede input, så neuronens output bliver $\sigma(z_j^l + \Delta z_j^l)$ i stedet for $\sigma(z_j^l)$. Denne ændring skal gøre C mindre ved $\frac{\partial C}{\partial z_j^l}\Delta z_j^l$. Dermed er δ_j^l defineret som vores fejlled, og ses i ligning 2.10.

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l} \tag{2.10}$$

Da fejlledet er defineret, defineres den første antagelse ved ligningen for fejlen i netværkets outputlayer, δ^L, som ligning 2.11.

$$\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L) = \nabla_a C \circ \sigma'(z^L) \tag{2.11}$$

Ligning 2.11 har to led, hvor det ene led definerer hvor hurtigt C ændres af en funktion for j's outputaktivering, og det andet led er hastigheden for, hvor hurtig aktiveringsfunktionen σ ændres ved z_j^L. Efter lighedstegnet er det samme udtryk i matrix-form, da det kræves i backpropagation.

Den anden antagelse siger, at ligningen for fejlen δ^l i de næste layers δ^{l+1} er som ligning 2.12.

$$\delta^l = ((w^{l+1})^T\delta^{l+1} \circ \sigma'(z^l) \tag{2.12}$$

Ligning 2.12 siger, at hvis fejlen δ^{l+1} kendes, så bruges $(w^{l+1})^T$ til at bevæge fejlen tilbage af gennem netværket, der giver en fejl på netværkets outputlayer. Det sidste led får fejlen til at bevæge sig baglæns til aktiveringsfunktionen i layer l, som giver fejlen δ^l i det vægtede input til layer l.

Den tredje antagelse er for ændringsraten for tabsfunktionen med respekt for ethvert bias i netværket, som ses i ligning 2.13.

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \rightarrow \frac{\partial C}{\partial b} = \delta \tag{2.13}$$

Fejlen δ_j^l er lig med ændringsraten $^{\partial C}/_{\partial b_j^l}$, og antagelserne i ligning 2.11 og 2.12 siger, hvordan δ_j^l skal beregnes, som siger, at δ kan beregnes ved den samme neuron som ligningens bias b.

Den sidste antagelse siger, at ændringsraten for C med respekt for en hver vægt w i netværket skrives som ligning 2.14.

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l \tag{2.14}$$

Dette fortæller, at når aktiveringen er lille, vil $^{\partial C}/_{\partial w}$ også være lille, så vægtene vil lære i et langsomt tempo, så den ikke ændres meget i gradient descent (Nielsen, 2015, Chapter 2).

2.2.3 Gradient descent og den kvadratiske tabs-funktion

I tidligere dele i dette afsnit er der forklaret, hvordan netværket kører ved at ændre på vægtene og aktiveringerne gennem backpropagation, og der er tidligere nævnt tabsfunktionen C, hvor målet er, at det skal minimeres. Her defineres den første tabsfunktion C i ligning 2.15, som kaldes for den kvadratiske tabsfunktion, eller mean squared error (MSE), hvor x er et træningsinput, og y(x) er en vektor med det valgte output, mens n er antal input, w og b er henholdsvis samlingen af alle vægte og bias i netværket, og a er en vektor af outputs fra netværket.

$$C(w,b) \equiv \frac{1}{2n}\sum_x ||y(x)-a||^2 \rightarrow C = \frac{1}{n}\sum_x C_x \rightarrow C = \frac{1}{n}\sum_x \frac{||y(x)-a||^2}{2} \tag{2.15}$$

Ligning 2.15 har tre udtryk, hvor det andet og tredje udtryk bruges senere, men for nu bruges kun det første, og det er en ikke-negativ funktion, som nærmer sig 0, når output y(x) nærmer sig output $a \forall x$. Så målet er at finde værdier for w og b, som får C til at nærme sig 0, og det gøres med algoritmen gradient descent.

Da vægtene i vektorerene w og b netop ikke kun er et enkelt tal i hver, men millioner af variable i hver, så vil det være svært at finde minimum med calculus. Til det bruges algoritmen gradient descent, som er en algoritme, der vil prøve at finde ekstremum blandt alle de mange variable. Lad C være en funktion af m variable v, så er ændringen i ΔC, lavet af Δv, bestemt som ligning 2.16.

$$\Delta C \approx \nabla C * \Delta v \tag{2.16}$$

Her defineres C som en vektor med partial derivater med gradient-vektoreren ∇C, som ses i ligning 2.17.

$$\nabla C \equiv \left(\frac{\partial C}{\partial v_1}, ..., \frac{\partial C}{\partial v_m}\right)^T \tag{2.17}$$

For variablerne v, vil Δv være defineret som ligning 2.18.

$$\Delta v = -\eta \nabla C \rightarrow \Delta C \approx -\eta \nabla C * \nabla C = -\eta ||\nabla C||^2 \tag{2.18}$$

Hvor η er defineret som learning rate, og er en lille, positiv parameter, og ud fra ligning 2.16 kan det omskrives til det sidste udtryk i ligning 2.18. Når $||\nabla C||^2 \geq 0$, vil $\Delta C \leq 0$, og C vil altså altid falde og aldrig stige, når v ændres i ligning 2.18. Så når det er antaget, skal der beregnes en værdi for Δv, værdien for hvor meget v skal ændres med hele tiden for at finde et globalt minimum, som ses i ligning 2.19.

$$v \rightarrow v' = v - \eta \nabla C \tag{2.19}$$

Ligning 2.19 er dermed gradient descent-algoritmen, som hele tiden ændrer positionen for v med en lille mængde Δv for at finde minimum for C. Ligning 2.19 kan så blive lavet om til to ligninger, som repræsenterer gradient descent-algoritmen for vægtene w og bias b i ligning 2.20 og 2.21.

$$w_k \rightarrow w'_k = w_k - \eta \frac{\partial C}{\partial w_k} \tag{2.20}$$

$$b_l \rightarrow b'_l = b_l - \eta \frac{\partial C}{\partial b_l} \tag{2.21}$$

Ligning 2.20 og 2.21 kan dermed bruges til at finde et minimum for C. Der er dog nogle problemer ved dette, hvor andet og tredje udtryk i ligning 2.15 nu bruges, da det er et gennemsnit af tabsfunktionen for alle individuelle input x. Dermed skal alle individuelle input x beregnes ved gradienten ∇C_x, hvor der så tages gennemsnittet, og når n bliver stor, kan det tage rigtig lang tid, og læringen bliver langsom.

En måde at gøre algoritmen hurtigere på er ved at bruge stokastisk gradient descent. Det handler om, i stedet for at beregne hele ∇C, så beregnes små randomly bestemte stikprøver af træningsinput ved ∇C_x. Ved at tage det gennemsnitlige resultat af alle de små resultater, kommer der et godt estimat for ∇C, og som også er hurtigere.

De random-udvalgte træningsinputs $X1, X2, ...X_m$ ses som mini-batches og størrelsen af stikprøverne defineres m, og defineres som stor nok til at være et godt estimat. Ligningen ses i ligning 2.22.

$$\frac{\sum_{j=1}^{m} \nabla C_{X_j}}{m} \approx \frac{\sum_x \nabla C_x}{n} = \nabla C \tag{2.22}$$

For at kunne bruge det i neurale netværk, antages w_k og b_l at være vægte og bias i netværket. Så vil den stokastiske gradient descent hele tiden udvælge tilfældigt udvalgte mini-batches. Derefter tages summen af alle træningsekemplerne X_j i det udlagte mini-batch, og bliver ved med at udvælge tilfældige mini-batches, indtil alle træningsinputs er gået igennem en enkelt epoch. Det samme som i ligning 2.20 og 2.21 gøres i ligning 2.23 og 2.24, bare med mini-batches i stedet.

$$w_k \rightarrow w'_k = w_k - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_k} \tag{2.23}$$

$$b_l \rightarrow b'_l = b_l - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l} \tag{2.24}$$

Nu hvor den kvadratiske tabsfunktion kan bruges i et neuralt netværk med backpropagation laves nu to antagelser om tabsfunktionen.

Den første antagelse er, at tabs-funktionen C, ligning 2.15, bliver skrevet som et gennemsnit af tabs-funktionen C_x for individuelle input x. Det kan ses ved C_x i ligning 2.25

Den anden antagelse er, at tabs-funktionen C kan skrives som en funktion af outputs fra det neurale netværk, som kan skrives som ligning 2.25 (Nielsen, 2015, Chapter 1).

$$C = C_x = \frac{1}{2}||y - a^L||^2 = \frac{1}{2}\sum_j (y_j - a_j^L)^2 \tag{2.25}$$

2.3 Teknikker for at undgå overfitting

I de tidligere afsnit i dette kapitel er der forklaret, hvordan modeller i deep learning fungerer matematisk, men når der er lavet en model for det givet træningsdata, så ender det mange gange med, at modellen ikke kan forudsige mønstret i lignende data. Det vil altså sige, at modellen har lært strukturen i træningsdataet så godt, at den ikke generalisere på andet data, og dermed er modellen ikke noget værd. Tabs-funktionen kan dog blive ved med at falde, selvom accuracy stagnerer ved en bestemt epoch, og dermed siges det, at der er overfit ved den epoch, hvor der ikke sker en forbedring længere. Nogle eksempler på overfit kan ses i nedenstående punkter.

- Accuracy forbedres ikke ved en bestemt epoch
- Tabsfunktionen for valideringsdataet bliver forbedret indtil en bestemt epoch, men begynder så at stige. Men forbedres accuracy for både træning og validering, skal tabs-funktionen ikke bruges
- Accuracy er meget højere end valideringsaccuracy, så modellen husker resultaterne mere, end den generaliserer.

For at undgå overfit kan der blive brugt forskellige teknikker, som der her kort bliver forklaret, som kan være at bestemme hyperparameters som antal epochs, læringsraten og netværkets arkitektur. I modeller opereres der med tre datasæt, som er træning, test og validering, og hyper-parametrene skal sættes ud fra valideringsdataet i stedet for testdataet fordi, hvis hyper-parametrene sættes ud fra test-dataet, vil der blive en god accuracy for testdataet, men det er svært at teste, om den generaliserer nok overfor andet data. Hvis det er sat ud fra valideringsdata, og det så også går godt for testdataet, er der en god indikation for, at den generaliserer godt. Denne metode kaldes hold out-metoden, fordi valideringssættet holdes væk fra trænings- og testsættet (Nielsen, 2015, Chapter 3).

2.3.1 Regularization

Hvis der er overfit i modellen, og netværket og træningssættet er fast, kan der blive lavet regularization-teknikker. Den første teknik, der forklares, hedder vægthenfald eller L2-regularization, som handler om at tilføje en ekstra term til tabs-funktionen, som ses i ligning

2.26, hvor C_0 er en given tabsfunktion.

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2 \tag{2.26}$$

I ligning 2.26 tilføjes summen af de kvadrerede vægte i netværket til C, hvor vægtene er skaleret med en faktor, $\lambda/2n$, hvor regularization-parameteren $\lambda > 0$. λ er en parameter, som skal vælge, om tabs-funktionen skal søge at finde små vægte eller minimere den originale tabs-funktion. Når λ er lille, skal den originale tabs-funktion minimeres, fordi der her er tale om, at den hverken er god på træning eller validering. Når λ er stor, skal der findes nogle små vægte for at forhindre overfit. Ligesom tabsfunktionen er lavet om, skal læringsraten for gradient descent og stokastisk gradient descent-algoritmen for vægtene w, ligning 2.20 og 2.23, også laves om. Ligningerne for b, 2.21 og 2.24 bliver ikke lavet om, og forbliver de samme. Ligning 2.20 og 2.23, opdateringsreglen, bliver lavet om i ligning 2.27 og 2.28.

$$w \to w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta\lambda}{n} w = \left(1 - \frac{\eta\lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w} \tag{2.27}$$

$$w \to \left(1 - \frac{\eta\lambda}{n}\right) w - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial w} \tag{2.28}$$

Ligning 2.27 og 2.28 er de samme som ligning 2.20 og 2.23 bortset fra en tilføjelse ved $1 - \eta\lambda/n$, som er L2-regularization-faktoren. Det udtryk fortæller sammenhængen mellem datastørrelsen og λ. Hvis datastørrelsen forstørres, stiger med en faktor 50, skal λ også stige med 50, for at reguleringen ikke bliver udvandet, og det er tilfældet, hvis læringsraten η er konstant.

Grunden til bias ikke bliver ændret, er fordi en stor bias ikke gør en neuron følsom overfor dens input på samme måde som vægtene gør. Derudover vil store biases også gøre, at neuronerne bliver mættet, som den skal.

En anden regularization-teknik er L1-regularization, som handler om at tilføje summen af de absolutte værdier til en given tabs-funktion. Dette ses i ligning 2.29.

$$C = C_0 + \frac{\lambda}{n} \sum_w |w| \tag{2.29}$$

I ligning 2.29 er der tilføjet et led til tabsfunktionen, som skal bruges til at få netværket til at ønske små værdier, ligesom ved L2. Hvordan netværket med en L1-regularization fungerer, kan ses i ligning 2.30, hvor leddet tilføjes til den partielle derivat af tabs-funktionen for w.

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} sgn(w) \tag{2.30}$$

I ligning 2.30 blev der tilføjet leddet $\frac{\lambda}{n} sgn(w)$, og sgn(w) siger, at hvis w er positiv, så er w lig 1, og er w negativ, er w lig -1. Det opdaterer så opdateringsreglen, ligning 2.23, til ligning 2.31.

$$w \to w' = w - \frac{\eta\lambda}{n} sign(w) - \eta \frac{\partial C_0}{\partial w} \tag{2.31}$$

I både ligning 2.28 og 2.31 er intentionen at minimere vægtene ved de store vægte. Men metoderne er forskellige, for ved L1 bliver vægtene mindre med en konstant, indtil det når 0, og ved L2, minimeres vægtene med en mængde, der er proportionel med w.
En tredje regularization-teknik er at bruge dropouts. Her gælder det ikke om at forbedre tabs-funktionen, men at forbedre hele netværket. Hvis der er træningsinputtet x og det valgte output y, køres dataet normalt først igennem med forward-propagation også backpropgation bagefter. Men ved dropout vil der randomly og midlertidigt blive fjernet en procentdel af hidden neuroner i netværket. Herefter forward-propagate inputtet gennem netværket, også laves backpropagation gennem det nye netværk. Til sidst kommer de fjernede neuroner tilbage, og der bliver nu lavet den samme procedure med et nyt sæt randomly-valgte neuroner, der fjernes midlertidigt, som kører samme procedure. Men når de kommer tilbage, er netværket dobbelt så stort, fordi netværket har lavet nye neuroner, hvis dropout er 50%, så vægtene halveres til sidst. De forskellige netværk vil på skift give overfit i forskellige scenarier, men med et gennemsnit af de forskellige netværk, der er kørt, skulle overfitting blive fjernet. Det skal fjerne overfitting fordi der i et normalt netværk, vil være en tendens til, at en neuron ikke kan være afhængig af specifikke tidligere neuroner, og er derfor tvunget til at lære mere robuste detaljer, som er brugbar ved foreningen af de mange random subsets af neuronerne.
En fjerde regularization-teknik er at udvide træningssættet. Med flere input vil netværket kunne lære flere og anderledes egenskaber ved data'et. Hvis det blev plottet for accuracy mod sætstørrelse for træningssættet, er der bevis for, at en større mængde data vil gøre accuracy højere. Hvis der ikke er mere data til rådighed, kan man prøve at lave mere data ved at tilføje noget af det allerede eksisterende, men hvor det er ændret en lille smule (Nielsen, 2015, Chapter 3).

2.3.2 Hvordan hyper-parameters vælges til det neurale netværk

Det kan være meget svært at finde det rigtige sæt af hyper-parametre, men der er nogle teknikker, der kan bruges for at finde et godt sæt parametre. Den brede strategi, der skal startes med, er at gøre netværket så simpelt som muligt. Hvis der er mange mulige outputs for inputtene, vil det være en fordel at fjerne nogle af outputtene, så det kun er et binært valg mellem to klasser. Det vil gære dataet meget mindre, og det vil gøre træningen hurtigere. Det vil give indsigt i, hvordan netværket agerer på dataet. Derefter kan antallet af mulige

outputs stige, til det fulde data er implementeret. Det samme kan også gøres for antallet af træningsinput generelt uden at fjerne nogle klasser. Ved hver gennemgang af netværket, kan der hele tiden blive forbedret værdier for læringsraten η og L2-parameteren, λ, og netværket kan herefter blive mere kompleks.

En hyper-parameter som læringsraten η kan blive valgt ved at plotte læringsraten ved tab mod epochs, og her vælges en læringsrate, der gør, at tabet falder. Er η for stor, så vil de steps i algoritmen være for store i ligning 2.19, og algoritmen kommer op af minimumspunkterne igen. Men at vælge en for lille η vil gøre den stokatiske gradient descent-algoritme for langsom og tabs-funktionen bliver ikke minimeret. Så metoden for at finde en god værdi er at tage en værdi, og køre det igennem. Hvis tabet falder, skal der prøves med højere værdier af η, indtil lige før den begynder at stige. Hvis η stiger, skal der prøves med lavere værdier. Men η må aldrig være større end grænseværdien, og en god regel er, at den skal minimeres med en faktor 2 af grænseværdien. Er der fire mulige output, skal den så være under 0,25 eller med faktor 2: 0,125. Dette vil gøre, at der trænes med mange epochs uden, læringen bliver langsom. Ved dette vil der blive lavet en konstant læringsrate. Men den kan også variere mellem et interval. Metoden er at køre netværket for nogle få epochs, også plotte cost mod læringsraten. I det interval, hvor der ses det største fald i tab-funktionen, vil være det interval, læringsraten skal være i mellem.

En anden hyper-parameter er at bruge early-stopping. Det handler om at beregnes accuracy på valideringsdataet. Ved den epoch, hvor accuracy stopper med at blive forbedret, skal modellen stoppe. Det afholder også modellen fra overfit. Men dette er når, de andre parametre er blevet fastsat. Men det kan være, at accuracy for nogle få epochs ikke forbedres, men så forbedres ved senere epochs. Derfor bliver der lavet en parameter, patience, som sikrer, at den stopper, når modellen ikke er forbedret indenfor antallet af epochs, patience-parameteren har givet af tid.

En tredje hyperparameter, regularizationsparametren λ, er igen ved at prøve sig frem. Der skal startes med $\lambda = 0$, og her kan der først vælges en god læringsrate η. Derefter kan $\lambda = 1$, og her få det til at stige eller falde med en faktor 10, indtil modellen er forbedret så meget som muligt. Til sidst skal η så optimeres igen, når λ er forbedret så meget som muligt (Nielsen, 2015, Chapter 3).

KAPITEL 3

Lineære modeller og evalueringsmetricses

Dette kapitel har til formål at give et overblik over den lineære model ARIMA, som skal bruges til at forecaste aktieprisen. Heri vil der blive forklaret, hvilke kriterier modellerne vælges ud fra, hvordan forecastene laves og evalueres. Til sidst vil der blive forklaret forskellige nøgletal, som skal bruges til at sammenligne på tværs af, ikke bare de lineære modeller, men mellem alle modellerne, der bruges i dettes speciale. Det vedrører både nøgletal på træningsdelen, forecastene, retning og markedstiming m.m.

3.1 ARIMA-modeller

ARIMA står for AutoRegressive Integrated Moving Average, og ses som en ARIMA(p,d,q). De første to bogstaver, AR, fortæller, at funktionen på tidspunkt t er afhængig af tidligere perioder med en konstant. De sidste to, MA, betyder, at fejlledet på tiden t er ligeledes afhængig af fejlledet på tidligere tidspunkter med en konstant. Til sidst betyder I, at den er differenced I gange for at gøre den stationær, som behandles i ligning 3.3. En ARIMA-model kan ses som ligning 3.1 (Enders, 2015, p.50-51).

$$y_t = a_0 + \sum_{i=1}^{p} a_i y_{t-i} + \sum_{i=0}^{q} \beta_i \varepsilon_{t-i} \tag{3.1}$$

For en ARMA-model, ligning 3.1, er brugbar, skal den være stationær. En ARMA-model skal altid være stationær, som betyder, den har et konstant gennemsnit, varians og kovarians. Men kravene er forskellige for AR- og MA-delen i modellen. For MA-delen skal være stationær skal i, antal lags, være endelig, altså definerbar. Så for AR-delen er stationær, skal der gælde, at alle de karakteristiske rødder skal være indenfor enhedscirklen, så $1-\sum a_i > 0$, og dermed gælder ligning 3.2, og ARMA-modellen er stationær (Enders, 2015, p. 55-60).

$$Ey_t = Ey_{t-s} = {}^{a_0}/_{(1-\sum a_i)} \tag{3.2}$$

Hvis ARMA-modellen ikke er stationær, skal den gøres stationær ved at difference den d gange, indtil den er stationær. Måden at difference modellen på er ved ligning 3.3, hvor det ikke er ARMA-modellen længere men forskellen mellem hver periode. Dette gøres, indtil den er stationær.

$$z_t = y_t - y_{t-1} = \Delta y_t \rightarrow z_t = (a_0 + a_1 y_{t-1} + \beta_1 \varepsilon_t + \beta_2 \varepsilon_{t-1}) - (a_0 + a_1 y_{t-2} + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2}) \tag{3.3}$$

3.1.1 Kriterier for modelvalg

Når der er lavet forskellige ARIMA-modeller, vil spørgsmålet så være hvilken model, der har den rigtige mænge af p, d, og q, der passer dataet bedst. Derfor bliver der her defineret to kriterier for valg af den bedste model, som hedder Akaike Information Criterion (AIC) og Schwartz Bayesian Criterion (SBC), hvor formlerne er i ligningerne i 3.4 og 3.5, hvor n er antal estimerede parametre, som er p, q og en kontant, og T er antallet af anvendte observationer.

$$AIC = T * ln(sum\ of\ squares\ residuals) + 2n \tag{3.4}$$

$$SBC = T * ln(sum\ of\ squares\ residuals) + n * ln(T) \tag{3.5}$$

For de to kriterier i ligning 3.4 kan bruges, skal T i modellerne være den samme, for ellers er kriterierne for hver model ikke sammenlignelige. Den bedste model er den model, der har den laveste AIC og SBC, som kan nå $-\infty$. Hvis der er en model, der har den laveste værdi for begge parametre, vælges den, men har den ene model den laveste AIC og en anden model har den laveste SBC, så bliver det lidt sværere. SBC har en tendens til at vælge den mest sparsommelige, og her skal der tjekkes, om fejlledene er white noise. AIC har en tendens til at vælge en model, der har for mange parametre, og derfor skal der tjekkes, om hver parameter er signifikant. Er den model, som SBC vælger, en model der har flere signifikante parametre, skal den vælges (Enders, 2015, p. 69-70).

3.1.2 Forecast

Hovedformålet med en ARIMA-model er at lave et forecast, en forudsigelse ud i fremtiden, ud fra det data der er til rådighed. Derfor er det nødvendigt at forklare, hvordan sådan et forecast bliver til, og hvad der skal være opmærksomhed på, når der bliver lavet et forecast. For ligningen $y_t = a_0 + a_1 y_{t-1} + \varepsilon_t$ kan regnes for tiden t+1 og fremadrettet ses ligning 3.6, hvor $E_t y_{t+1}$ er forventningen af y_{t+1} betinget af informationen på tiden t.

$$y_{t+1} = a_0 + a_1 y_t + \varepsilon_{t+1} \rightarrow E_t y_{t+1} = a_0 + a_1 y_t \rightarrow E_t y_{t+2} = a_0 + a_1 E_t y_{t+1} \tag{3.6}$$

Fra ligning 3.6 kan det så følge, at den generelle ligning, forecast-funktionen, kan ses som ligning 3.7, som er j step ud i fremtiden ud fra informationen på tid t.

$$E_t y_{t+j+1} = a_0 + a_1 E_t y_{t+j} \rightarrow E_t y_{t+1} = a_0(1 + a_1 + a_1^2 + ... + a_1^{j-1}) + a_1^j y_t \tag{3.7}$$

Herfra kan det betingede gennemsnit af disse forecasts i en ARIMA-model, når $j \rightarrow \infty$ konvergere mod det ubetingede gennemsnit $a_0/1 - a_1$. Dermed kan der konkluderes, at jo større j bliver, jo større bliver fejlledet i forecastet, og den betingede varians, som ses i ligning 3.8, bliver også dermed større.

$$var[e_t(j)] = \sigma^2[1 + a_1^2 + a_1^4 + a_1^6 + ... + a_1^{2(j-1)}] \tag{3.8}$$

Den betingede varians, når $j \rightarrow \infty$ konvergerer mod den ubetingede varians, som er $\sigma^2/(1 - a_1^2)$. Til sidst kan der defineres et konfidensinterval. Hvis der antages, at fejlledet ε_t er normalfordelt, kan der laves et konfidensinterval omkring forecastet. Et forecast for t+1 er defineret som ligning 3.9 (Enders, 2015, p. 79-81).

$$a_0 + a_1 y_t \pm 1.96\sigma \tag{3.9}$$

Når forecastet så er lavet, kan der så spekuleres i, hvor godt forecastet er. En måde at løse dette problem er ved at holde noget data tilbage, holdback-perioden, og lave et forecast med det resterende. Herefter kan der vurderes, hvor godt forecastet fanger holdback-perioden. Her kan de forskellige modeller, der er estimeret blive vurderet op mod hinanden om, hvilken model der fanger holdback-perioden bedst ved at se hvilken model, der har den laveste forecastfejl. Dette kaldes in-of-sample-forecast. Måden at evaluere er ved mean square prediction error (MSPE), som ses ligning 3.10, hvor e er forecasterrors (Enders, 2015, p. 82-88).

$$MSPE = \frac{1}{H} \sum_{i=1}^{H} e_{1i}^2 \tag{3.10}$$

3.1.3 Evaluering af forecasts

Når der er flere forskellige forecast, ville en sammenligning af de to forecasts MSPE, ligning 3.10, være at godt bud til at sammenligne de to forecast. Men forecastene kan fejle på forskellige tidspunkter af forecastet, og der skal dermed tjekkes, om forecastene er signifikant forskellige. Her skal der bruges Diebold-Mariano test for at teste, om forecast f og g er signifikant forskellige fra hinanden, hvor forecastet er lavet af tidsserien y. Lad e og r være fejlledene i forecastene, se ligning 3.11 og 3.12. Antagelsen for Diebold-Mariano er, at d skal

være stationær (Zaiontz, 2021).

$$e_i = y_i - f_i \tag{3.11}$$

$$r_i = y_i - g_i \tag{3.12}$$

Lad så d være tabsforskellen i enten MSE eller MAE, se afsnit 3.2 for definition. Tabsforskellen d er defineret i ligningerne 3.13 til 3.15. Ligning 3.15 er autokovariansen ved lag k.

$$d_i = e_i^2 - r_i^2 \; eller \; d_i = |e_i| - |r_i| \tag{3.13}$$

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \; \mu = E[d_i] \tag{3.14}$$

$$y_k = \frac{1}{n}\sum_{i=k+1}^{n} (d_i - \bar{d})(d_{i-k} - \bar{d}) \; for \; n > k \geq 1 \tag{3.15}$$

Diebold-Mariano-testen kan så defineres for forecasthorisonten $h \geq 1$ som ligning 3.16.

$$DM = \frac{\bar{d}}{\sqrt{[y_0 + 2\sum_{k=1}^{h-1} y_k]/n}} \tag{3.16}$$

Er DM normalfordelt $DM \; N(0,1)$ er der en signifikant forskel mellem forskellene. Så $H0$ er, at der er en signifant forskel mellem forecastene (Zaiontz, 2021).
Ifølge (Zaiontz, 2021) er det generelt en god ide at lave forecastet på 1/3-del af observationerne. Er det tilfældet, at der er for lille et datagrundlag, er det bedre at bruge HLN-testen, da DM-testen ellers afviser nulhypotesen for tit. HLN-testen kan ses i ligning 3.17 (Zaiontz, 2021).

$$HLN = DM\sqrt{[n + 1 - 2h + h(h-1)]/n} \tag{3.17}$$

3.2 Evalueringsmetrics

De tre evalueringsmetricses er i ligning 3.18, 3.19 og 3.20. I ligningerne bliver variablerne defineret ved, at y_{true} er den rigtige værdi af målvariablen, $y_{predict}$ er den modellerede værdi af målvariablen, $\bar{y}$ er gennemsnittet af målvariablens rigtige værdi for alle observationer, og m er antallet af observationer. I ligning 3.21 er n antal observationer, og k er antal uafhængige variable (Chugh, 2020).
Ligning 3.18 er Mean Absolutte Error (MAE), og er gennemsnittet af den absolutte forskel mellem den rigtige og modellerede værdi i modellen. Den beskriver gennemsnittet af fejlleddene. Ligning 3.19 består af to værdier (Chugh, 2020).

$$MAE = \frac{1}{m}\sum_{i=1}^{m} \left| y_{true} - y_{predict} \right| \tag{3.18}$$

Ligning 3.19 er Mean Squared Error (MSE), og er gennemsnittet af den kvadrerede forskel mellem den rigtige og modellerede værdi, og ses som variansen af fejlledene. Samtidig er der Root Mean Squared Error (RMSE), og er standardafvigelsen af fejlledene (Chugh, 2020).

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_{true} - y_{predict}\right)^2} \tag{3.19}$$

Ligning 3.20 repræsenterer delen af variansen for den afhængige variabel, der er forklaret af modellen, og vil være mindre end 1 (Chugh, 2020).

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = 1 - \frac{\sum_i \left(y_{true} - y_{predict}\right)^2}{\sum_i \left(y_{true} - \bar{y}\right)^2} \tag{3.20}$$

Ligning 3.21 er en modificeret version af ligning 3.20, da den er justeret for antallet af uafhængige variable, og vil altid være mindre eller lig R^2 (Chugh, 2020).

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n-1)}{n-k-1}\right] \tag{3.21}$$

3.3 Test af retningen ved forecast

Når der er lavet et forecast, uanset hvilken model, der er defineret, kan der testes for, hvor godt forecastet rammer den præcise pris. Men der kan også testes for, om forecastet bare har retningen rigtig. Det kan der testes for ved (Pesaran and Timmermann, 1992). Der testes dermed for, om modellen binært, op eller ned, rammer rigtigt. Generelt vil den ikke-parametriske test af præstationen for forudsigelsen af x_t kan være generaliseret af ligning 3.22.

$$S_n = \frac{\hat{P} - \hat{P}_*}{\sqrt{v\hat{a}r(\hat{P}) - v\hat{a}r(\hat{P}_*)}} \sim N(0,1) \tag{3.22}$$

hvor de forskellige parametre er defineret som i ligningerne 3.23 til 3.28.

$$\hat{P} = n^{-1}\sum_{i=1}^{n} Z_t = n^{-1}\sum_{i=1}^{n} Pr(y_t x_t > 0) \tag{3.23}$$

$$\hat{P}_* = \hat{P}_y\hat{P}_x + (1-\hat{P}_y)(1-\hat{P}_x) \tag{3.24}$$

$$\hat{P}_y = \sum_{i=1}^{n} {}^{Y_t}/_n = \bar{Y} \tag{3.25}$$

$$\hat{P}_x = \sum_{i=1}^{n} {}^{X_y}/_n = \bar{X} \tag{3.26}$$

$$v\hat{a}r(\hat{P}) = n^{-1}\hat{P}_*(1-\hat{P}_*) \tag{3.27}$$

$$v\hat{a}r(\hat{P}_*) = n^{-1}(2\hat{P}_y - 1)^2\hat{P}_x(1-\hat{P}_x) + n^{-1}(2\hat{P}_x - 1)^2\hat{P}_y(1-\hat{P}_y) + 4n^{-2}\hat{P}_y\hat{P}_x(1-\hat{P}_y)(1-\hat{P}_x) \tag{3.28}$$

Ligning 3.23 fortæller andelen af de gange, at retningen har været rigtigt estimeret. Den fortæller sandsynligheden for, at den forecastestimerede værdi og den rigtige værdi har samme retning. Jo tætter $\hat{P}$ er på en, jo mere præcis er forecastet. Det kan dermed ses, om retningsvariablen er statistisk signifikant (Pesaran and Timmermann, 1992).

3.4 Markedstiming

Testen om markedstiming er lavet af (Pesaran and Timmermann, 1994), og giver en test, som kan analysere markedstimingen ud fra to forskellige strategier for modellering af, i dette tilfælde, aktieprisen. Den undersøger den statistiske signifikans for korrellationen mellem forecastet og de aktuelle værdier for afkastene på aktier. Lad y_t være de virkelige værdier for afkastene for en given portefølje, og lad x_t være forecastet af y_t. Deraf kan de fælles

sandsynligheder defineres i tabel 3.1 (Pesaran and Timmermann, 1994).

Rigtige værdier	Modellerede værdier		
	$x_t < 0$	$x_t \geq 0$	
$y_t < 0$	$P_{11}(n_{11})$	$P_{12}(n_{12})$	$P_{10}(n_{10})$
$y_t \geq 0$	$P_{21}(n_{21})$	$P_{22}(n_{22})$	$P_{20}(n_{20})$
	$P_{01}(n_{01})$	$P_{02}(n_{02})$	$I(n)$

Tabel 3.1: Markedstiming - Hentet fra (Pesaran and Timmermann, 1994)

Den ikke-parametriske test er baserede på de betingede sandsynligheder $P_1 = P_{11}/P_{10}$ og $P_2 = P_{22}/P_{20}$. En betingelse for et rationelt forecast er at have en positiv værdi, hvor $P_1(t) + P_2(t) > 1$. Nulhypotesen er så, at der ikke er nogen markedstiming, som ses i ligning 3.29 (Pesaran and Timmermann, 1994).

$$H_0 : \frac{P_{11}}{P_{10}} = \frac{P_{21}}{P_{20}} \rightarrow H_0 : \frac{P_{11}}{P_{10}} + \frac{P_{21}}{P_{20}} = 1 \tag{3.29}$$

3.5 Sharpe Ratio

Sharpe Ratio er et nøgletal, som bruges til at se på, hvordan en investering har performet justeret for dens risiko. Ratioen beskriver hvor meget afkast, der hentes ved ekstra volatilitet for at holde et mere risikobetonet aktiv. Nøgletallet beregnes som i ligning 3.30.

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p} \tag{3.30}$$

hvor R_p er afkastet på den pågældende portefølje, R_f er den risikofrie rate, typisk fra en statsobligation (U.S. Treasury rate), og σ er volatilliteten af porteføljen. Jo større Sharpe-ratio, jo mere attraktiv er det risikojusterede afkast, og jo mere attraktiv er aktivet, da man altid vil have større afkast for mindre risiko (Fernando, 2021).

KAPITEL 4

Data og modeller

I dette afsnit vil hele datagrundlaget blive fremvist i forskellige afsnit, alt efter hvilken analyse, det enkelte datasæt hører til. Først vil der blive præsenteret selve aktiekursen (eller indekser), der skal estimeres, og derefter dataet til de underliggende variable og valg af modeller.

4.1 S&P500

Her præsenteres variablen, der skal modelleres og forecastes på. Det er $S\&P500$-indekset, som er et aktiemarkedsindeks, der repræsenterer de 500 største amerikanske virksomheder, der har deres aktier på det regulære aktiemarked. For at være i indekset skal man have en ujusteret market cap på mindst 8,2 mia. dollars, mindst 50% af aktierne skal være tilgængelig for offentligheden, prisen skal mindst være 1 dollar pr. aktie, mindst 50% af deres faste aktiver og overskud skal være i USA, og til sidst skal de have mindst fire kvartalers positiv indtjening (Amadeo, 2020).

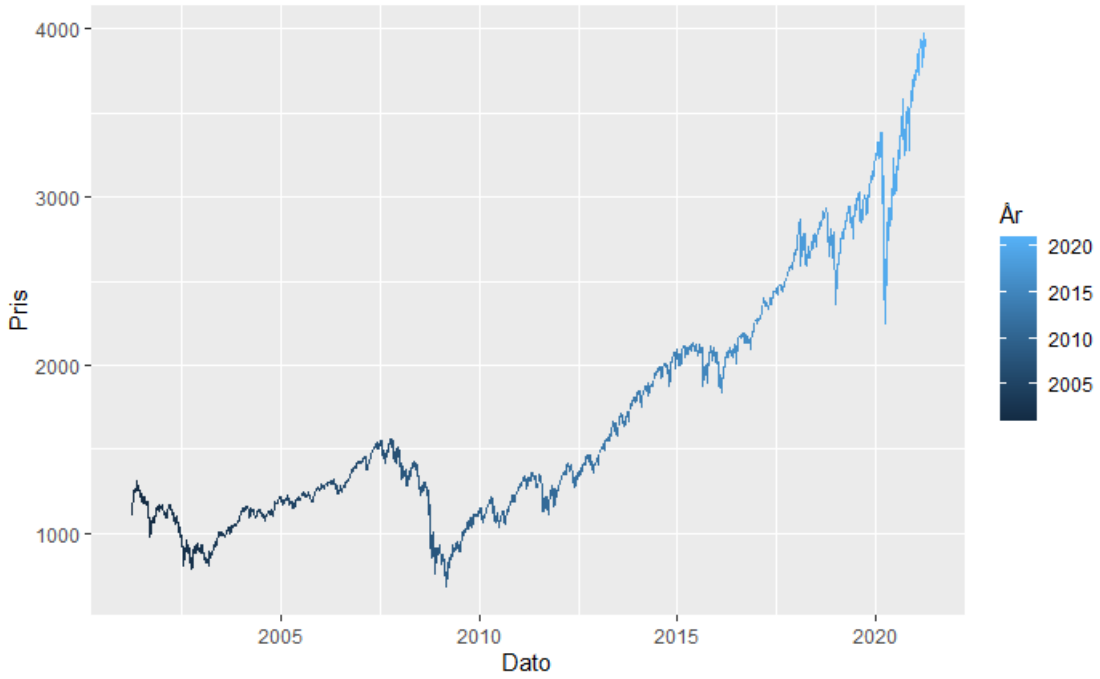
Hvor meget dataet strækker sig over, varierer efter analyse, fordi når der skal laves en ana-

Tabel 4.1: Statistik for SP500 for hele perioden

Statistikvariabel	Procentvis ændring	Prisændring
Min.	-11.98405%	-230.38
1st Qu	-0.44723%	-8.895
Median	0.06688%	-1.02
Mean	0.03219%	-0.553
3rd Qu	0.57288%	6.8
Max.	11.58004%	324.89

lyse om, hvorvidt ord fra artikler kan være med til at forklare variationerne fra periode til periode for S&P500-indekset, bliver perioden lidt kortere. Forklaringen findes i afsnit 4.4. Prisen for S&P-500 i perioden 2001 til 2021 kan ses i figur 4.1.

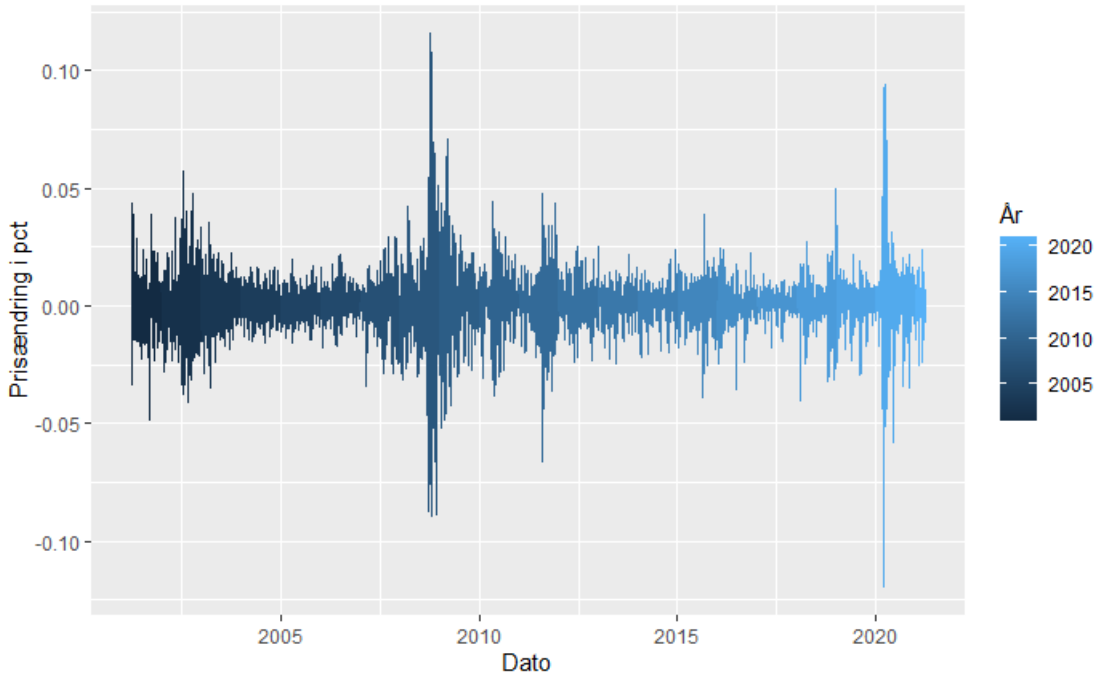
Variablen S&P500-indekset er fra dag til dag, hvor tidsperioden i det længste tilfælde er



Figur 4.1: Pris for S&P500 for hele perioden

fra juli 2001 til april 2021. Den tidsperiodes statistiske parametre er i tabel 4.1, som viser deskriptiv statistik, og fortæller, at S&P500 for hver dag i gennemsnit stiger med 0,03% med et topafkast på 11,5% og -11,6% som største afkasttab. Disse ses som outliere, og vurderes til at være ved finanskrisen i 2009 og corona i 2020. Normaltvis svinger de med en lille varians, da første og tredje kvartal er tæt på gennemsnittet, se også figur 4.1. Noget andet er den almindelige prisændring, som har en større grad volatilitet, som også ses i figur 4.1. Dette forklarer godt, at modelleringen må ske ved den procentvise ændring, for på den måde, følger dataet mere en normalfordeling og med en mindre varians.

Dataet bliver i tilfældet med RNN-modellen lavet til at splitte tre gange, hvor træningssættet slutter, på den dato der er 30% tilbage af datasættet, som er dateret til d. 30. Marts 2015, hvor testsættet starter. Ved RNN-modellen laves også et split ved de sidste 10% til validering. Grunden til, splittet er ved 30% er fordi, da afsnittet om foreevalueringen blev lavet, se afsnit 3.1.3 (Zaiontz, 2021), blevet det vurderet, at splittet skulle være der.



Figur 4.2: Den procentvise ændring fra dag til dag for hele perioden

4.2 DeepLearning

I modellen, hvor DeepLearning skal bruges, modelleres dataet med Recurrent Neural Network (RNN), som er en model, som er baseret på en rækkefølge (sequence). Disse baserer sig på, at observation i er en fortsættelse i rækkefølgen af obseration i-1. Derfor bruges de ofte, når der skal laves en model for tidsserie eller textmining, som betyder, at den tager elementerne i sammenhæng (Chollet and Allaire, 2018a, p. 180-182). Men RNN-layers er for simpel at bruge, da den kun bruger tid t-1 til at forudsige t, så derfor bruges LSTM-layers, som gemmer information fra alle tidligere layers (Chollet and Allaire, 2018a, p. 186-188). Denne form tager længere tid, men er også nødvendig, da en aktie ikke kun afhænger af prisen dagen inden. For dette kan lade sig gøre i praksis, kan dataet ikke kun være en vektor, men skal laves til en matrice, som har de foregående dage, som leder op til den pågældende dag i matricen. Dette vises i figur 4.3. Herefter bliver dataet transformeret, ikke til procentvis

$$\begin{pmatrix} t-3 & t-2 & t-1 & t \\ t+1-3 & t+1-2 & t+1-1 & t+1 \\ t+2-3 & t+2-2 & t+2-1 & t+2 \end{pmatrix}$$

Figur 4.3: Tidsrække

ændring, men der bliver taget kvadratroden af hele tidsserien, og derefter normaliseres det med ligning 4.1.

$$X' = \frac{X - \mu}{\sigma} \tag{4.1}$$

Ligning 4.1 er en z-score-normalisering, hvor værdierne for gennemsnit og standardafvigelse er i tabel 4.2, og indikerer hvor langt standardafvigelsen er fra gennemsnittet (Ciaburro and Venkateswaran, 2017, p. 180). Dette gøres i stedet for at modellere den procentvise ændring. Normaltvis ville det være mere nyttigt med min-max-normalisering, da dette gør, at dataet ligger mellem 0 og 1, men der er opnået bedre resultater med en z-score normalisering.

Center	40.3251
Scale	8.330657

Tabel 4.2: Preprocesing for deep learning

4.3 ARIMA

I dette afsnit skal der laves den bedst mulige ARIMA-model for hele perioden. Til forskel fra DeepLearning-modellen, bliver der brugt den procentuelle ændring til modellering, som vises i figur 4.2. Som før nævnt er der splittet ved 30% af dataet, sådan modellen kun modelleres på de første 70, som så udgør træningssættet. Den bedste model vælges ud fra værdien AIC, som er forklaret i kapitel 3, og hvor resultaterne vises i tabel 4.3. Her skal der vælges den model med den mindste AIC-værdi, som er en ARIMA(5,0,3) med et gennemsnit, der ikke er lig 0. Modellen står med fed i tabellen.

Model	AIC
ARIMA(2,0,2) with non-zero mean	-20799.71
ARIMA(0,0,0) with non-zero mean	-20753.96
ARIMA(1,0,0) with non-zero mean	-20785.72
ARIMA(0,0,1) with non-zero mean	-20781.39
ARIMA(0,0,0) with zero mean	-20755.78
ARIMA(1,0,2) with non-zero mean	-20789.93
ARIMA(2,0,1) with non-zero mean	-20793.64
ARIMA(3,0,2) with non-zero mean	-20804.11
ARIMA(3,0,1) with non-zero mean	-20801.33
ARIMA(4,0,2) with non-zero mean	Inf
ARIMA(3,0,3) with non-zero mean	-20804.64
ARIMA(2,0,3) with non-zero mean	-20804.75
ARIMA(1,0,3) with non-zero mean	-20789.92
ARIMA(2,0,4) with non-zero mean	-20802.75
ARIMA(1,0,4) with non-zero mean	-20789.69
ARIMA(3,0,4) with non-zero mean	-20806.16
ARIMA(4,0,4) with non-zero mean	-20804.21
ARIMA(3,0,5) with non-zero mean	-20803.8
ARIMA(2,0,5) with non-zero mean	-20806.15
ARIMA(4,0,3) with non-zero mean	-20806.47
ARIMA(5,0,3) with non-zero mean	**-20813.31**
ARIMA(5,0,2) with non-zero mean	-20809.7
ARIMA(5,0,4) with non-zero mean	-20801.49
ARIMA(5,0,3) with zero mean	Inf

Tabel 4.3: Modeludvælgelse af ARIMA

Når den mest præcise model er fundet, skal der testes for, om den bedre kan modelleres gennem en ARCH-model. Der bliver testet for homoskedasticitet ved fejlledene i den stationære proces. Resultatet er i tabel 4.4. Som der står, er nul-hypotesen, at der ingen ARCH-effekter er, og den kan ikke forkastes med den lille p-værdi. Derfor skal den ikke estimeres med GARCH.

χ^2	DF	Nul-hypotese	Pværdi
1520.2	12	Ingen ARCH-effekter	<2.2e-16

Tabel 4.4: ARCH-test for fejlledene

4.4 Avisoverskrifter som underliggende variabel

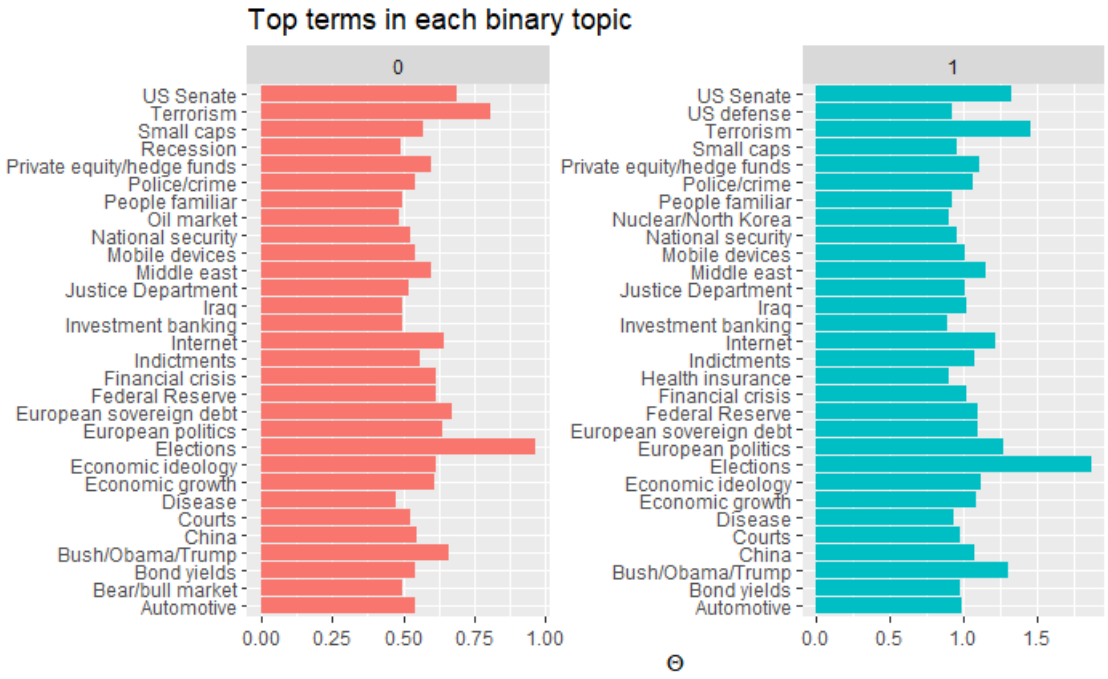
Her er det ideen at bruge avisartikler som underliggende variabel til at kunne modellere en aktiekurs. Det data, der bliver brugt i denne rapport, kommer af Bryan Kelly, som er professor i finans på Yale School of Management, og bl.a. forsker han i, hvordan textmining kan bruges til at modellere fluktueringer i økonomiske nøgletal (Kelly, 2021). På (Kelly, 2021)'s hjemmeside kan der blive fundet data af overskrifter til avisartikler, som han har lavet textmining ud af. Avisartiklerne kommer fra Wall Street Journal, og indeholder data fra januar 1984 til juni 2017 på månedlig basis. Derfor er der 800.000 forskellige artikler derfra, og det munder ud i 180 ord/emner. Det er regnet ud med variablen θ som viser, hvor meget hver emne/ord k vægter på et tidspunkt t. Dataet er kun på månedlig basis, og der bliver brugt data, der er fra 2001 og frem til 2017 i denne rapport. Variablen θ er regnet som i ligning 4.2, og er opmærksomheden på emne k på måned τ (Kelly et al., 2020).

$$\theta_{\tau,k} = \frac{\sum_{t\epsilon\tau}\sum_{i=1}^{N_i} I(z_{t,i}\hat{} = k)}{\sum_{t\epsilon\tau}\sum_{q=1}^{K} I(z_{t,i}\hat{} = q)} \tag{4.2}$$

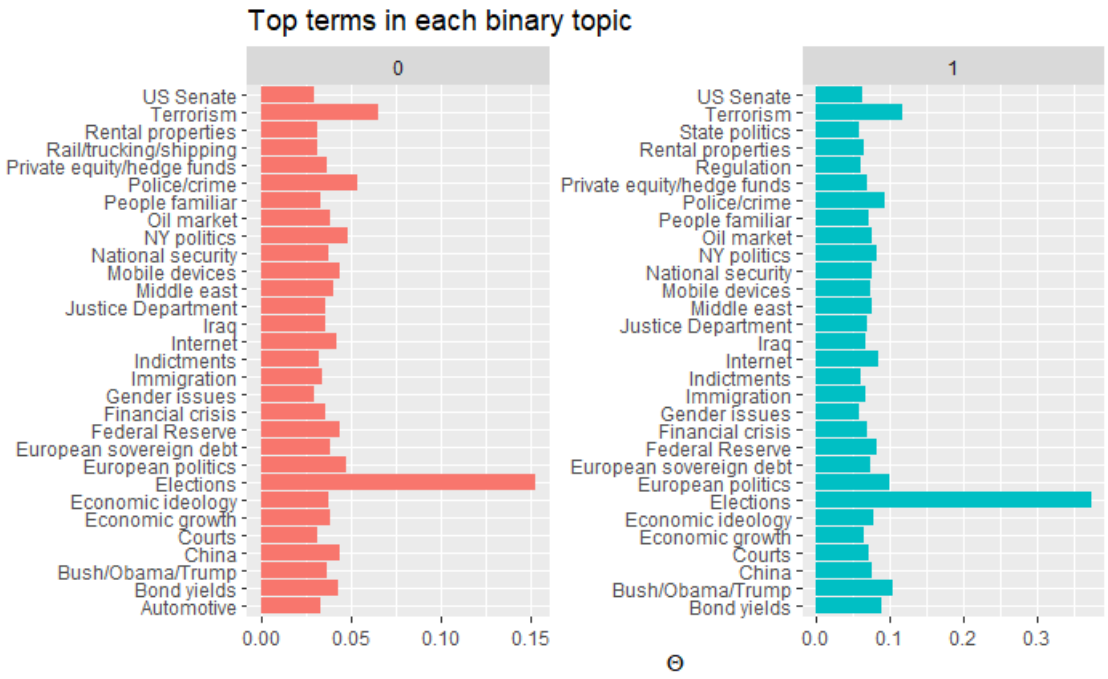
Her kan θ bruges til at se, om der er en sammenhæng mellem termernes θ og aktiekursens fluktueringer. Det bruger (Kelly et al., 2020) til at kunne modellere vækst i industriproduktion og vækst i beskæftigelse med en R^2- værdi på henholdsvis 0.22 og 0.59 med supervised machinelearning.

I figur 4.4 og 4.5 vises der, hvilke ord der har betydning for, når kursen er gået ned, 0, og når den er gået op, 1. Det kan være nemt at se i firgur 4.5, at 2016 er et valgår, men den helt store sammenhæng mellem kursens retning og θ er ikke til at få øje på. Nogle få ord er dog kun i den ene af 0 og 1 i figur 4.4, og ud fra dette vil health insurance og us-defence være at finde på listen, når indekset går op, og oilmarket og recession vil være at finde, når det går ned. Det kan der være en sammenhæng i, da amerikanerne altid ser deres forsvar som noget positivt, og når der bliver investeret i forsvaret, kommer der også flere arbejdspladser. Hvis der så ses på ordet recession, som er på listen, når det går ned, så har ordet sikkert været brugt mange gange omkring finanskrisen, og derfor giver det også god mening, at det bruges, når det går ned.

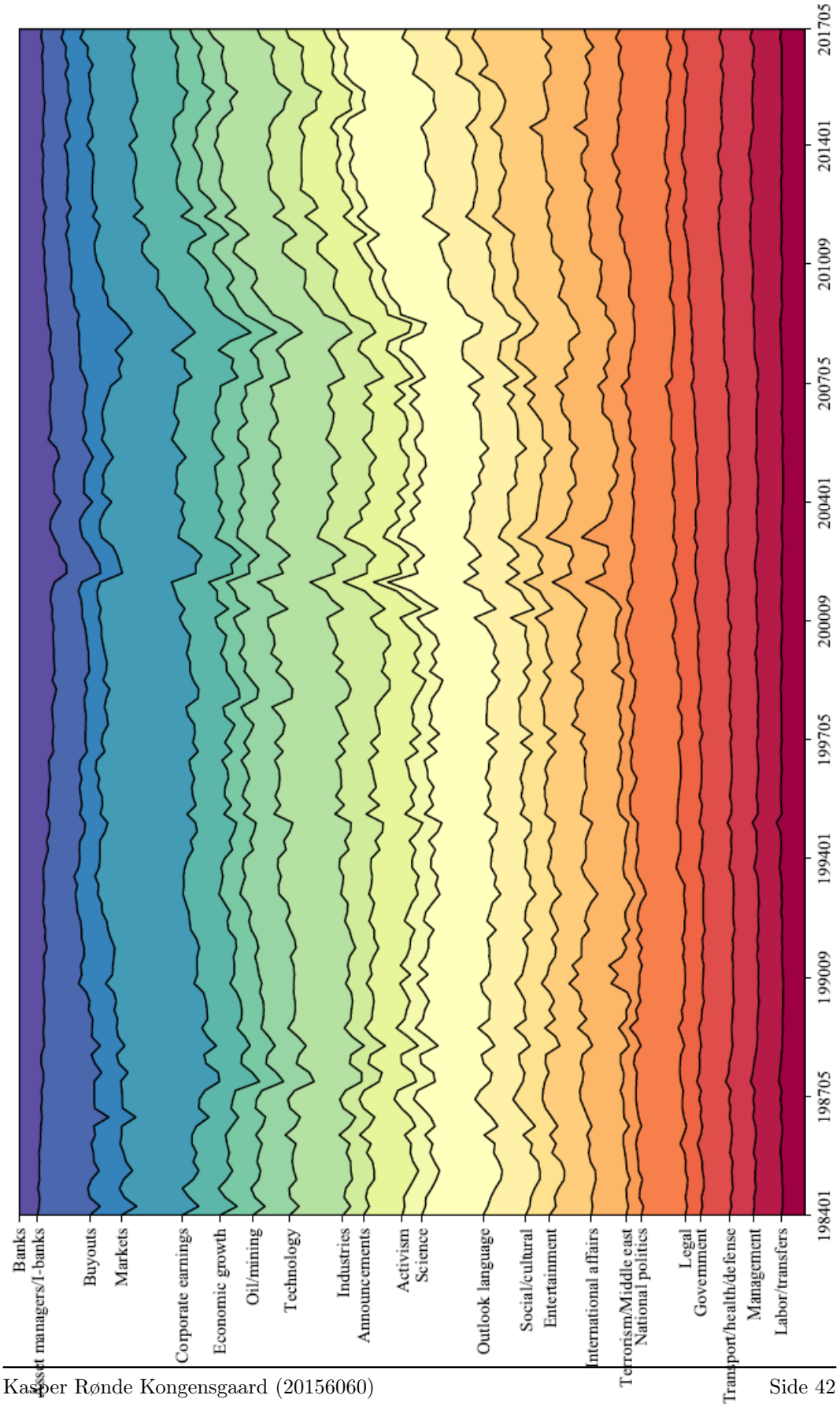
I artiklen (Kelly et al., 2020) bliver der brugt andre metoder til at gøre det mere overskueligt med at vise dataet. Det vises i figur 4.6, og her bruges hierarchical clustering fra unsupervised machinelearning til at samle de 180 ord til 23 metatopics. Hierachical cluste-



Figur 4.4: Ords θ i hele perioden



Figur 4.5: Ords θ i 2016



Figur 4.6: Metatopics baseret på clustering - Fra (Kelly et al., 2020)

ring går kort ud på, at alle variablerne bliver samlet som et datapunkt, også samles de emner, der ligger tættest på hinanden, indtil alle er samlet i et cluster (Chauhan, 2019). Det gør, at computeren selv vælger, hvordan de forskellige emner passer bedst sammen uden indblanding fra mennesker. Ud fra figur 4.6 så ligger emnerne meget stabile over tid med små fluktueringer. Men der er dog nogle ting, som bliver mindre, eksempelvis olie og minedrift, som giver god mening, da det er blevet mere udfaset over årene med den nye klimadagsorden. Et emne, som er blevet større over årene er Mellemøsten og terrorisme, som er et enkelt emne. Det fyldte ikke det store i 80'erne, men fik et boost i starten af 00'erne med World Trade Center, og stille og roligt efter boostet er et større emne blevet større.
Metoden for at modellere dataet er stort set identisk med metoden for den almindelige modellering af tidsserien med RNN- og LSTM-modeller. Her bruges igen illustrationen fra figur 4.3, men her tilføjes alle de 180 ord som forklarende variable til hver måned. Derfor vises også tabel 4.5 for at vise tallene for skaleringen, som er kommet af samme metode, ligning 4.1.

Center	37.201674
Scale	5.339898

Tabel 4.5: Preprocesing for Avis deep learning

4.5 Farma&French

Her fortsættes afsnittet om Farma&French-modellen fra afsnit 1.3, som vil modellere aktieindekset S&P500. De ligninger, der er i det nævnte afsnit, er ikke beregnet i dette speciale, da forfatteren til disse modeller, har lavet beregningerne på hjemmesiden `https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`. Derfor skal tallene kun downloades, også kan det bruges til at beregne de tre modeller i ligning 1.2, 1.3 og 1.4, som er henholdsvis CapM, Tre-faktor-modellen og Fem-faktor-modellen. Perioden er valgt fra d. 1 juli 2001 til 2 februar 2020, og forskellige nøgletal for hver variabel kan ses i tabel 4.6. Variablerne er blevet forklaret i afsnit 1.3, men navnene her er lidt anderledes, for R_excess

Tabel 4.6: EDA for de forskellige variable ved Farma&French

	Gennemsnit	Median	Standardafvigelse	Minimum	Maksimum
Afkast	0.0002	0.001	0.012	-0.090	0.116
MKT_RF	0.0003	0.001	0.012	-0.090	0.113
SMB	0.0001	0.0002	0.006	-0.034	0.045
HML	0.00004	0	0.006	-0.044	0.048
RMW	0.0002	0.0001	0.004	-0.026	0.033
CMA	0.0001	0	0.003	-0.026	0.020
RF	0.0001	0.00003	0.0001	0	0.0002
R_excess	0.0002	0.0005	0.012	-0.090	0.116

er afkastet af S&P500 fra dag til dag minus afkastet på den risikofrie rente RF. Derudover er MKT_RF lig med markedsafkastet minus den risikofrie rente RF. Resten hedder det samme som i afsnit 1.3. Resultaterne kommer i kapitel 5.

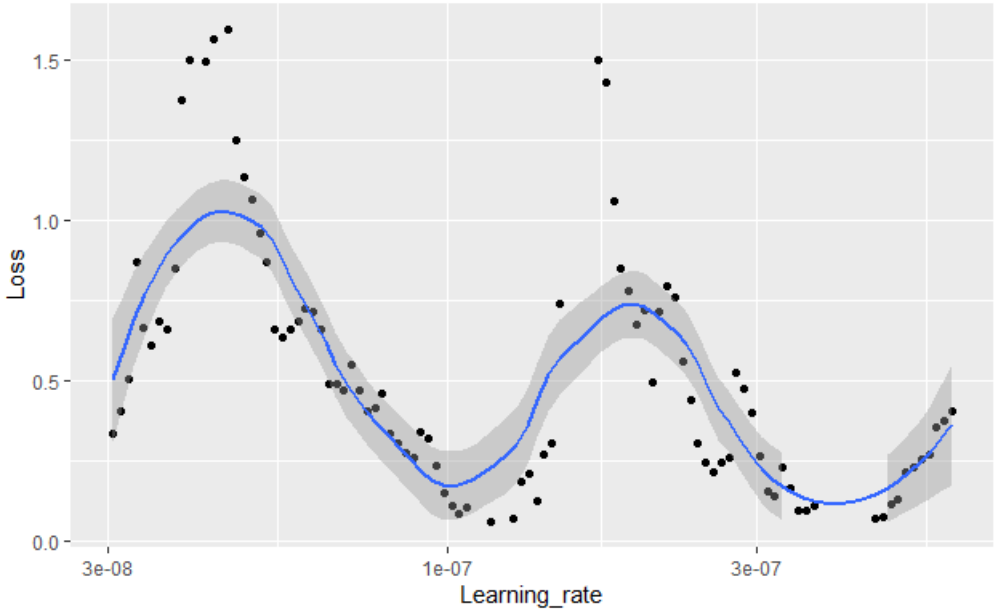
KAPITEL 5

Empiriske resultater

I dette kapitel vises alle de empiriske resultater for hver model, og hvilke yderligere overvejelser, der er gjort for at forbedre modellerne, som ikke er medtaget i kapitel 4. Det bliver gjort i et afsnit for hver model. De sidste tre afsnit i dette kapitel har til formål at evaluere på forecastene, og sammenligne de forskellige modeller mod hinanden på både størrelse af forecastfejl, retning og markedstiming. Alle parametrene er forklaret og defineret i kapitel 3.

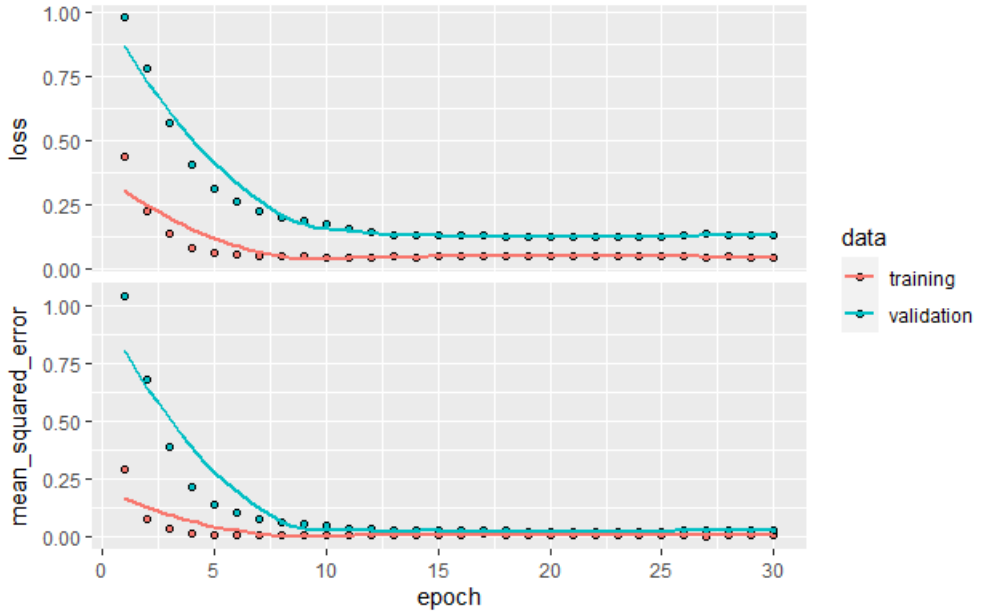
5.1 RNN-model

Her i dette afsnit forklares, hvordan DeepLearning-modellen laves, og hvilke resultater, den har produceret. Der er brugt LSTM-layers, da RNN-layers er layers kun er beregnet til data, som afhænger af det forrige i sættet. Det blev forklaret tidligere, og før den skal køre dataet i gennem endeligt, skal der først bestemmes et interval, som læringsraten skal holde sig indenfor. Det gøres, som forklaret i afsnit 2.3.2, hvor modellen kører i et enkelt epoch, hvor resultatet for læringsraten vises i figur 5.1. I den figur skal der vælges det interval, som har den stejleste kurve nedad, og i modellen er der valgt intervallet mellem 6e-08 til 1e-07, som viser en stejl nedgang i tabsfunktionen.



Figur 5.1: Tabs-funktion mod læringsrate

Herefter giver det en callback til modellen, som bestemmer, at læringsraten skal være inde i det interval. Hvis læringsraten er for stor, vil algoritmen risikere at overse et lokalt minimum, som kan være mindre end det minimum, den allerede har fundet, og er den for lille, vil modellen være for langsom til at modellere, og man kan vente evigt med at få et resultat. Derfor er et interval bestemt. Det leder frem til figur 5.2, som viser, hvordan tabs-funktionen og MSE falder for både validering og træning, så den ikke er i nærheden af at overfitte. En anden callback-funktion kaldet early-stopping er også brugt, og den bruges til at stoppe modellen, hvis der ikke er forbedring for modellen indenfor det antal epochs, der kaldes patience. Patience er her sat til 7, og holder øje med MSE og ikke loss-funktionen. Men med et maks antal epochs på 30, har patience ikke været nødvendig. Resultatet af algoritmen ses i tabel 5.1, og viser værdierne for, hvordan modellen har virket på de enkelte dele for trænings-, validerings- og testdelen. Selvom figur 5.2 ikke viser nogen tegn på overfitting, fanger modellen ikke testsættet lige så godt som træningssættet, så der er en smule overfit, men den lille mængde, der er, er ret lille. I hver af de layers, der er i modellen, er der inkorporeret en dropout-rate på 1%, men jvf. afsnit 2.3 kunne der være blevet brugt L2-regularization eksempelvis, men det er prøvet uden den store forbedring.



Figur 5.2: Historyplot

Metric	Datamængde	Værdi
Loss	Træning og validering	0.0469
	Træning	0.0402
	Validering	0.1322
	Test	0.5561
Mean squared error	Træning og validering	0.009055
	Træning	0.0033
	Validering	0.0293
	Test	0.3671

Tabel 5.1: Metrics for RNN-modellen

Til sidst er der lavet nogle grafer for, hvordan modellen klarer sig, prismæssigt i figur 5.3 og procentuel i 5.4. I disse modeller for deep learning er valgt prisen i stedet for prisændringen i modelleringen, da resultaterne blev bedre i modelleringerne, se resultaterne i afsnit 5.7. I figur 5.3 ses det, at modellen klarer det super godt i træningsperioden, men at der kommer nogle problemer, så snart forecastet er cirka 2 år inde, men ifølge figur 4.2 sker der også nogle store udsving, som modelleringen også får med, men udsvingene er bare ikke store nok. Forecastet starter som nævnt d. 30. marts 2015.

Figur 5.3: Modellingsgraf af prisen

Figur 5.4: Modelleringsgraf

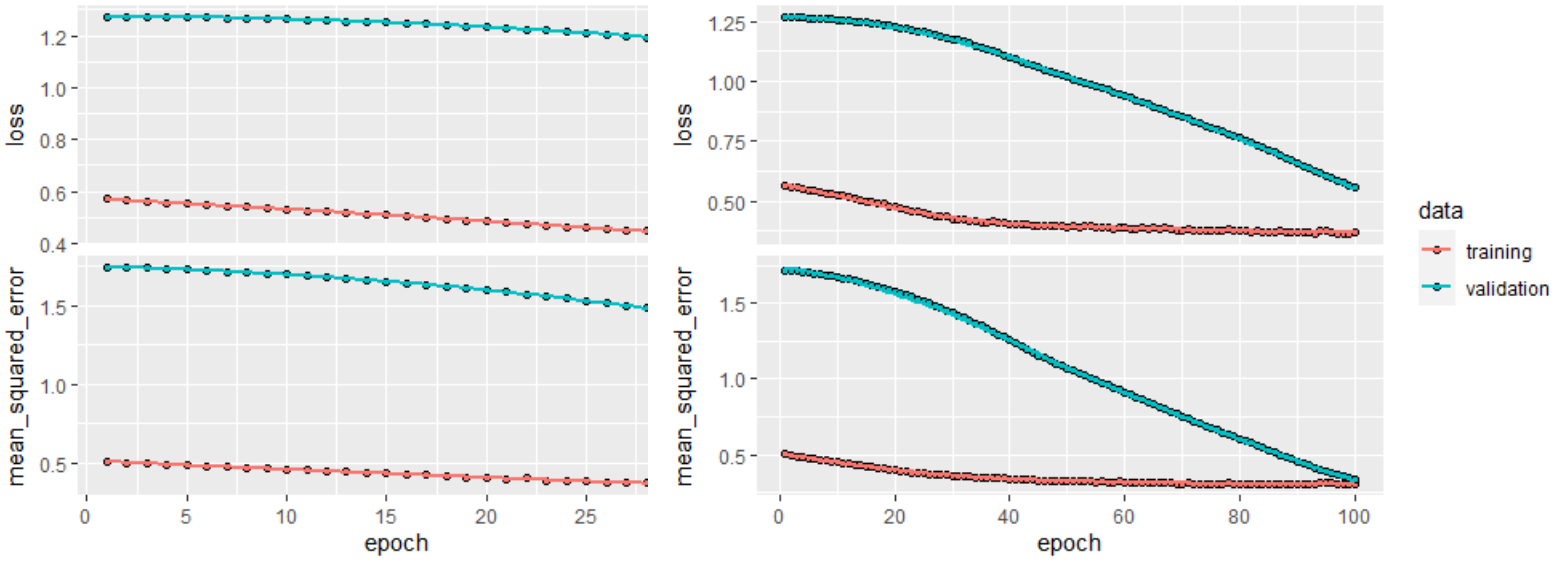
5.2 Tekstmining af avisoverskrifter

Avisoverskrifter er nyheder, som kommer fra dag til dag, men jævnfør afsnit 4.4 så er dette data kun tilgængelig på månedsbasis, og selvom der er 180 datapunkter og tidsserien, må det konstateres, at resultaterne er noget dårligere end ved almindelig tidsserie i RNN-modeller, jvf. afsnit 5.1. Det kan konstateres ved tabel 5.2, som har nogle væsentlig højere værdier end den traditionelle RNN-model på alle parametre, jvf. tabel 5.1. Men der er ikke tegn på overfit, når der ses på træning og validering, men der kan dog godt argumenteres for det, hvis der ses på testsættet, da det er lidt højere, men det er allerede forbedret, jvf. figur 5.5a. Værdierne i tabel 5.2 er de værdier, der kommer af en stigning af antallet af epochs, jvf. figur 5.5a og 5.5b, så værdierne er fra figur 5.5b.

Metric	Datamængde	Værdi
Loss	Træning og validering	0.3694
	Træning	0.3640
	Validering	0.5546
	Test	0.7025
Mean squared error	Træning og validering	0.3112
	Træning	0.3056
	Validering	0.3339
	Test	0.5855

Tabel 5.2: Metrics for Avis-modellen

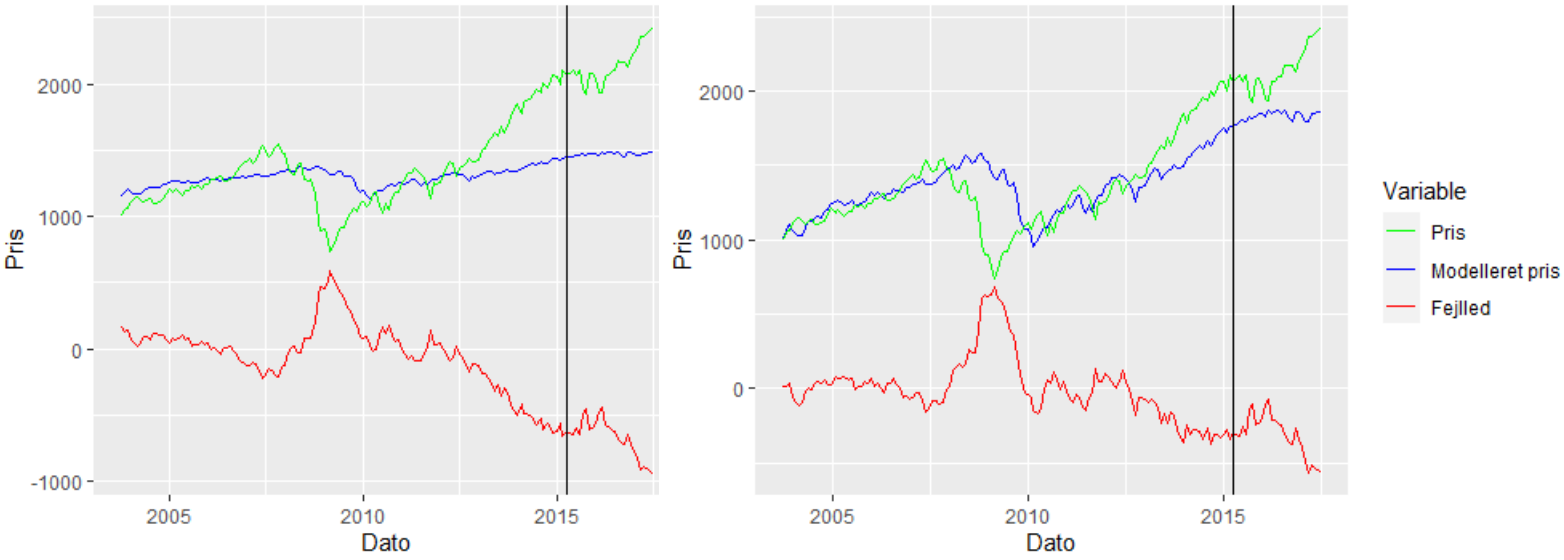
Ud fra afsnit 2.3 så skal MSE og valideringsmse forbedres over hele perioden, men mse er meget højere end valideringen, og derfor er et af kriterierne opfyldt for overfit, se figur 5.5a. Men afsnit 2.3 fortæller også om nogle metoder for at undgå dette overfit, og det kan bl.a. gøres ved at forhøje antallet af epochs fra 30 til 100, og det kan ses i figur 5.5b. I figur 5.5b falder både mse og valideringsmse, og smelter sammen, når de rammer epoch 100. Derfor var det ikke nødvendig med andre tiltag end at forhøje antallet af epochs.



(a) Historyplot af avismodel (b) 2. Historyplot af avismodel

Figur 5.5: De to historyplot for avismodellen

Resultaterne af de 70 ekstra epochs kan også ses tydeligt i figur 5.6a og 5.6b, som forbedrer sig markant. Fra næsten bare at være en kontant modellering i figur 5.6a til at følge fluktueringerne i figur 5.6b, så ses det som en klar optimering af modellen, som kun krævede at forhøje antallet af epochs. Det skal så også siges til dette, at selvom dette er på månedsbasis, og den almindelige RNN-model var på dagsbasis, så er antallet af datapunkter højere, da der for den almindelige tidsserie havde cirka 22 datapunkter pr. måned, så har denne model et datapunkt pr. måned plus de 180 datapunkter, så denne model har flere datapunkter, men til gengæld har den almindelige tidsserie også priserne på dagsbasis, og ud fra tabel 5.1 og 5.2, så er modelleringen af dagspriserne stadig bedre. Men det vurderes, hvilken model der er bedst senere, jvf. afsnit 5.5, 5.6 og 5.7.



(a) 1. Visualisering af modelestimeringen

(b) 2. Visualisering af modelestimeringen

Figur 5.6: Visualisering af modelestimering

(a) 1. Modellering af de procentvise svingninger

(b) 2. Modellering af de procentvise svingninger

Figur 5.7: Modellering af de procentvise svingninger

5.3 ARIMA

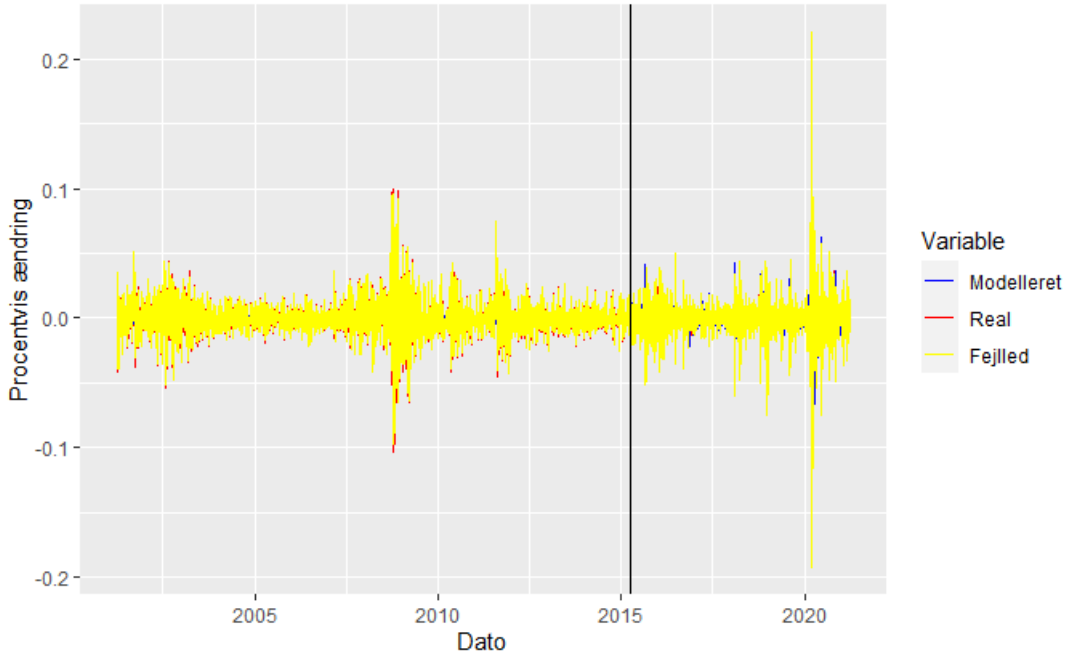
Der er allerede forklaret i afsnit 4.3, at ARIMA-modellen skal modelleres ved ARIMA(5,0,3) med et gennemsnit, der ikke er 0. Deruodver var der ingen GARCH-effekter, og resultatet af det valg, er modellen bestemt til at være det, der vises i tabel 5.3. Der er nogle, ar5 og ma2, der er signifikante på alle tre værdier, og derudover er ar2 signifikant på 5 og 10%, og resten er overhovedet signifikant på noget niveau. Men til trods for de to tidligere modeller, RNN-modellerne, skal resten modelleres ved afkastene fra dag til dag. I disse modeller vises ingen tegn på sæsonbetoning, da det er almindelige AR- og MA-effekter, der vises i tabellen. Det er til gengæld godt, da der så ikke skal laves yderligere tiltag for at fjerne sæsonbetoning, da en model der ignorerer disse sæsoneffekter vil have en høj varians (Enders, 2015, p.96). Derudover kan der også konstateres stationaritet, da en minus summen af parametrene i

Tabel 5.3: Resultater fra ARIMA-modellen

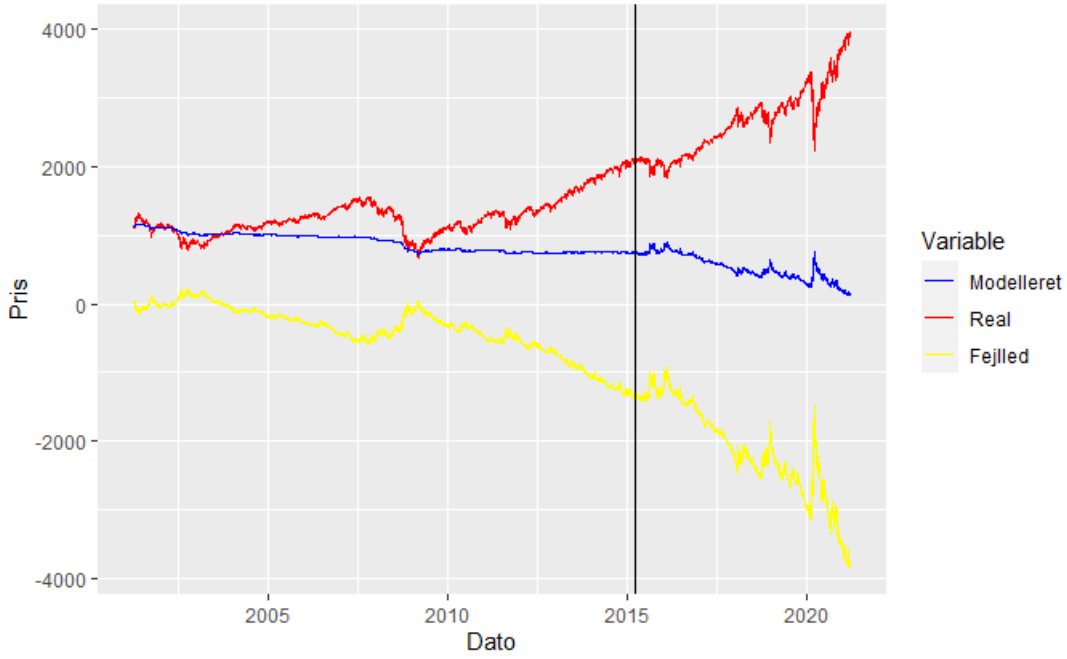
	Dependent variable:
ar1	-0.088
	(0.366)
ar2	0.340^{**}
	(0.150)
ar3	-0.325
	(0.301)
ar4	-0.018
	(0.030)
ar5	-0.089^{***}
	(0.018)
ma1	-0.002
	(0.368)
ma2	-0.392^{***}
	(0.136)
ma3	0.378
	(0.308)
intercept	-0.0001
	(0.0002)
Note:	*p<0.1; **p<0.05; ***p<0.01

modellen er over 0.

Men selvom der er nogle signifikante variable, så er det ikke gode takter, der findes i figur 5.8. Fejlledet fylder store dele af figuren, som egentlig skulle fylde mindst i figuren. Det samme billede gør sig gældende i figur 5.9, som viser med stor tydelighed, at den ikke fanger den opadgående trend, som prisen viser, der er. Ud fra figur 5.9 viser den mere en konstant trend, som også er lidt nedadgående. I afsnit 5.5, 5.6 og 5.7 bliver der vurderet på forskellige parametre, hvordan ARIMA-modellen klarer sig i forhold til de andre modeller.



Figur 5.8: Modellering af indekset for hele perioden med ARIMA



Figur 5.9: Modellering af indekset for hele perioden med ARIMA

5.4 Farma&French

I dette afsnit vises resultaterne fra Farma&French-modellerne, både som CapM, tre-faktor og fem-faktor. Ved det første øjekast på tabel 5.4 ses det, at alle parametrene er signifikante på alle niveauer i alle modellerne. Der kan også ses, at r^2 og den justerede r^2 begge stiger, når der bliver tilføjet de to faktorer og udgør tre-faktor-modellen, men den stiger dog ikke med tilføjelse af de sidste to parametre. Ses der på konstanten er den i minus, men så tæt på 0, at den godt kan vurderes som 0. Konstanten er indeksets α, og betyder at der efter omkostninger, så præsterer den lige så godt som benchmarken i markedet.

Tabel 5.4: Resultater for CapM, Tre-faktor- og Fem-faktormodellerne

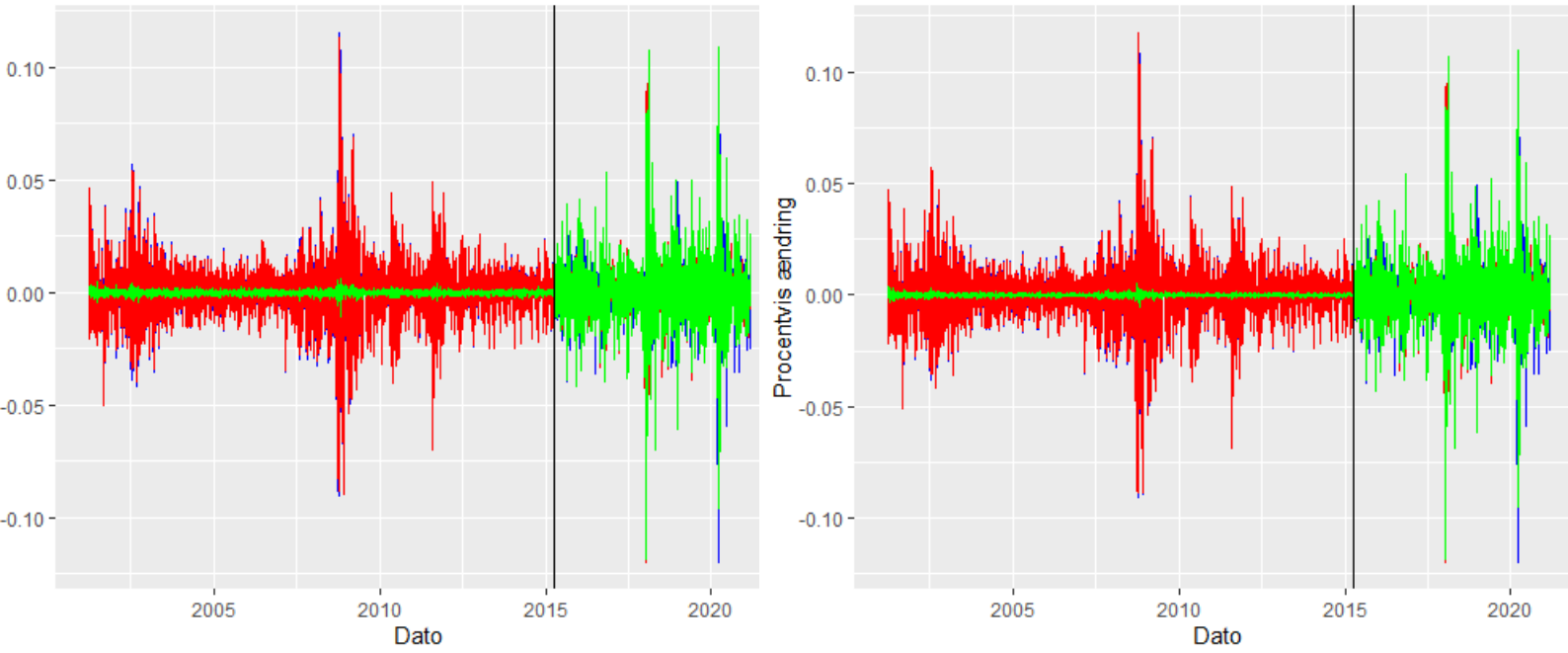
	Dependent variable:		
	R_excess		
	(1)	(2)	(3)
MKT_RF	0.998***	1.006***	1.016***
	(0.001)	(0.001)	(0.001)
SMB		−0.140***	−0.134***
		(0.002)	(0.002)
HML		0.020***	0.020***
		(0.002)	(0.002)
RMW			0.062***
			(0.002)
CMA			0.037***
			(0.003)
Constant	−0.0001***	−0.0001***	−0.0001***
	(0.00002)	(0.00001)	(0.00001)
Observations	3,518	3,518	3,518
R^2	0.994	0.998	0.998
Adjusted R^2	0.994	0.998	0.998
Residual Std. Error	0.001 (df = 3516)	0.001 (df = 3514)	0.001 (df = 3512)
F Statistic	551,347.600*** (df = 1; 3516)	512,352.000*** (df = 3; 3514)	480,445.400*** (df = 5; 4668)
Note:			*p<0.1; **p<0.05; ***p<0.01

Ses der på den første model, CapM, er den under 1, men også kun lige. Det betyder, den vurderer, at markedet er mere risikofyldt end aktivet, der er valgt i denne opgave. Men fortsætter man til de næste to modeller for denne parameter, stiger den, og kommer over 1, og dermed vurderer modellen at det pågældende aktiv er mere risikobetonet end markedet

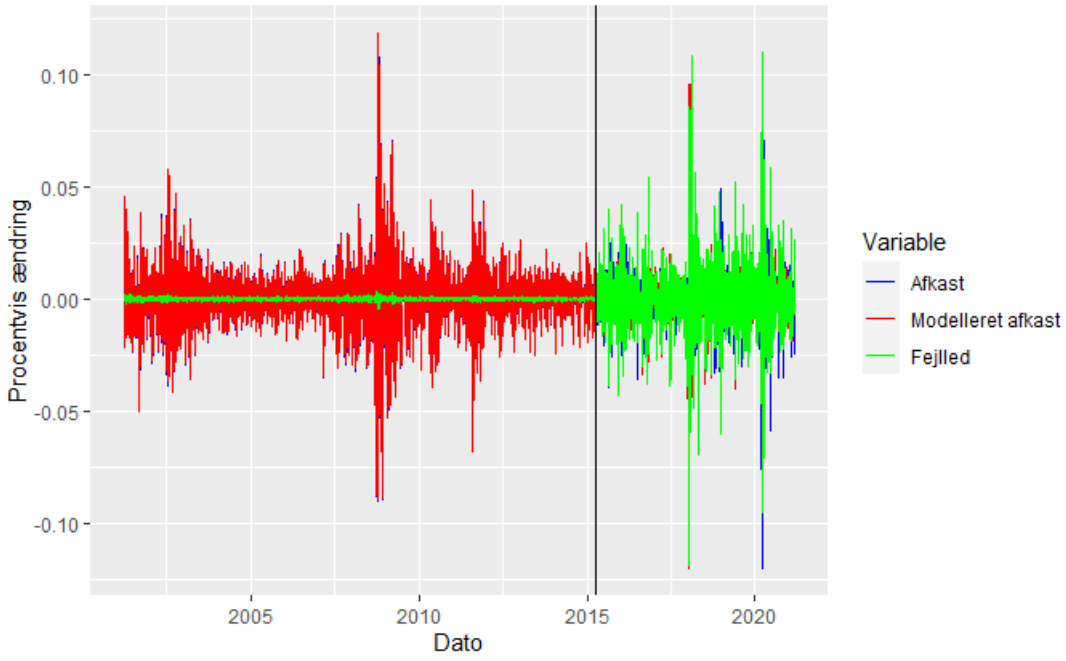
(Belyh, 2019). Der kan dog argumenteres for, at den i alle tilfælde er knap 1, hvilket giver god mening, da det pågældende aktiv er S&P-500-indekset, som fylder en stor del af markedet, og er dermed også en stor del af denne parameter.
Videre til tre-faktor-modellen, så har den parametrene SMB (SmallMinusBig) og HML (HighMinusLow), og ifølge afsnit 1.3 så skal de små outperforme de store virksomheder, når der tales om størrelse. Det viser det historiske afkast, som små virksomheder har givet over de store, den såkaldte size-effekt. Men da den parameter er i minus i begge af modellerne, viser den den modsatte effekt, store outperformer små virksomheder, så størrelseseffekten er her modsat. Der er større afkast ved store virksomheder. Hvis der fortsættes til den næste parameter, HML, er værdifaktoreren, som siger, at værdivirksomheder (med høj B/M) skal outperforme vækstvirksomheder (med lav B/M). Værdiaktier giver større afkast i fremtiden. Den parameter er også signifikant i begge modeller, og er positiv. Det betyder dermed, at værdiaktier gør det bedre end vækstaktier (Belyh, 2019), og at de billige aktier performer bedre end dyre (Musarurwa, 2019).
Hvis der bliver bevæget over til fem-faktor-modellens to sidste resultater er det RMW og CMA. Den første er RMW, som er de mest profitable minus de mindst profitable virksomheder. Det betyder, at virksomheder, der reklamerer med et større fremtidig afkast, altså større profitabilitet, giver større afkast i fremtiden. Det er også tilfældet i modellen, da den er positiv. Til sidst er der CMA, conservatively minus aggresively, som er forskellen mellem dem, der investerer forsigtigt og dem der investerer aggressivt. Dem der investerer aggressivt, og giver proffitten til store projekter vil give tab på aktiemarkedet (Belyh, 2019). Da den parameter er positiv, er det netop tilfældet.
Modellerne er estimeret med OLS for, hvordan indekset udvikler sig procentuelt, og det kan ses i figurene i 5.10, som viser estimeringerne for CapM (5.10a), Tre-faktor (5.10b) og Fem-faktor (5.10c). De tre modeller ser meget ens ud, både når der ses på fejlledet og estimeringerne på både trænings- og testsiden, som igen er placeret d. 1.april 2015. Men lige så meget, som de passer på træningssættet, har det svært ved at modellere på testsættet, altså forecastet, hvor det er fejlledet, der dominerer. Men hvis der ses på figur 5.11 ser det meget ens ud, hvordan de performer, når de er regnet tilbage til den originale pris. I den figur ses der, at den meget godt modellerer fluktueringerne, men at den vurderer for højt. Derfor skal der vurderes, hvor godt den performer. Det ses der nærmere på i afsnit 5.5, 5.6 og 5.7, hvordan de performer i forhold til hinanden, modellerne i mellem.

(a) Modellering af indekset for hele perioden med CapM

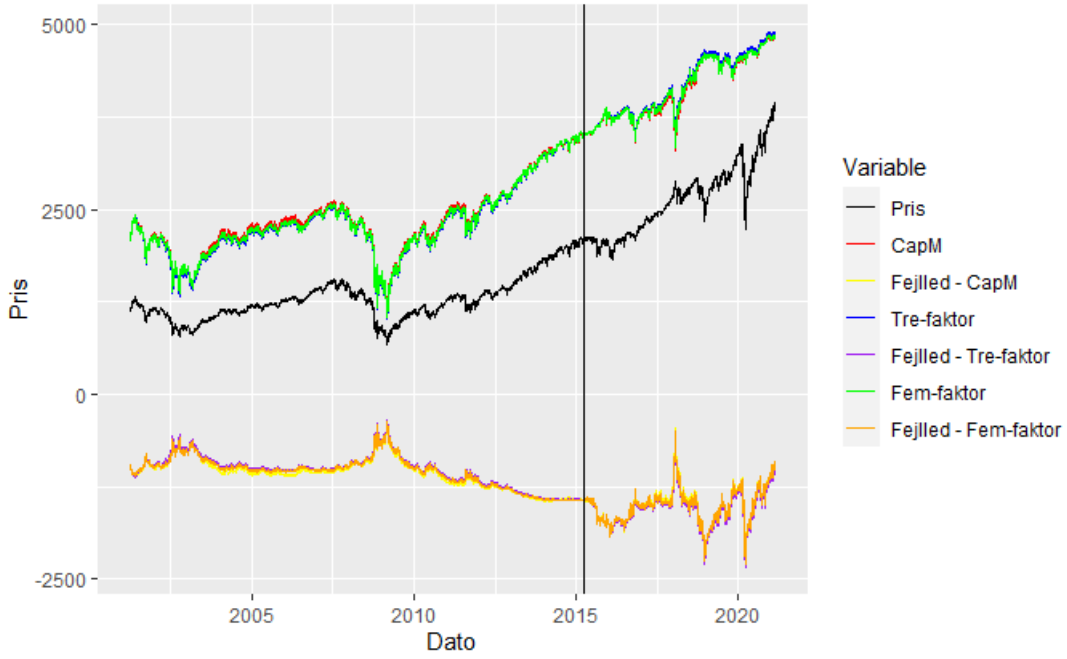
(b) Modellering af indekset for hele perioden med FF3



(c) Modellering af indekset for hele perioden med FF5



Figur 5.10: Forecasts for Farma&French



Figur 5.11: Sammenligning af modellernes evne til at følge virkeligheden

5.5 Evaluering af retningen

I tabel 5.5 vises resultaterne ved resultaterne fra (Pesaran and Timmermann, 1992), som viser, om modellerne i overhængende grad har retningen rigtig. Der er en værdi på både trænings- og forecastsættet, og som tabellen viser, bliver hver model alt andet lige dårligere fra trænings- til forecastsættet, bortset fra avis-modellen, som holder sig konstant, så den er bedst på forecastdelen overfor alle de andre modeller. Selvom ARIMA-modellen er bedre end RNN-modellen på træningsdelen, så er det stadig forecastdelen, der skal bruges, og det er så RNN-modellen, men de to modeller er meget tæt på hinanden.

Tabel 5.5: Test for retning - Pearson og Timmermann

Del	Model	Test	DirACC
Træning	DL	PT	0.5079745
Forecast	DL	PT	0.5006649
Træning	Capm	PT	0.9582149
Træning	Tre-faktor	PT	0.9752700
Træning	Fem-faktor	PT	0.9769756
Forecast	Capm	PT	0.4932796
Forecast	Tre-faktor	PT	0.5060484
Forecast	Fem-faktor	PT	0.5026882
Træning	ARIMA	PT	0.5279909
Forecast	ARIMA	PT	0.4731254
Træning	Avis	PT	0.5714286
Forecast	Avis	PT	0.5714286

5.6 Evaluering af markedstiming

Når der skal vurderes på markedstiming, bruges Henriksson-Merton-modellen, som forklares i afsnit 3.4. I tabel 5.6 ses resultaterne fra netop den model, som skal bruges i vurderingen af markedstiming, γ. Denne parameter er tilknyttet den dummyvariabel, som er 0 ved nedadgående tidspunkter og positive ved opadgående tidspunkter. Er γ positiv, er der markedstiming (Breaking Down Finance, nd). Når der ses på tabel 5.6, er det ikke overraskende, at det giver bedre værdier for træningssættet end for forecastet. Ses der på, hvordan de klarer sig i forhold til hinanden, så går det klart bedst for RNN-modellen, og det går så godt, at markedstimingen i forecastet for RNN-modellen er bedre end nogle af de andre i træningssættet med undtagelse af ARIMA-modellens træningssæt. Det er også overraskende, at Farma&French-modellerne egentlig bliver dårligere fra CapM til five-factor-modellen på træningssættet. På forecast-delen er det det netop dem, der gør det dårligst, for selv modellen med overskrifterne gør det bedre end dem, men de er i minus ligesom forecast-delene med Farma&French og ARIMA, så de har altså ikke markedstimings-evner.

Tabel 5.6: Markedstiming - Merton og Henriksson

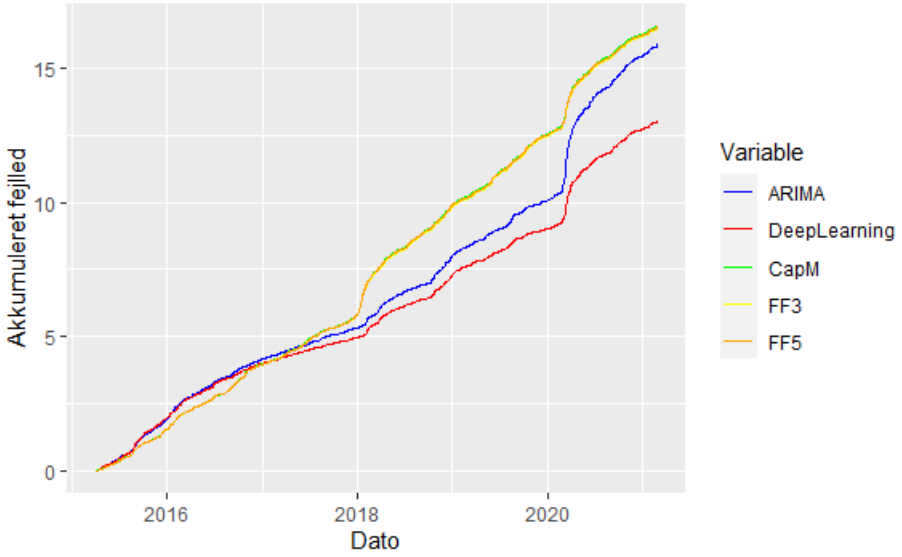
Del	Model	Test	Gamma
Træning	DL	HM	0.17514072
Forecast	DL	HM	0.05160519
Træning	Capm	HM	0.010643930
Træning	Tre-faktor	HM	0.002199227
Træning	Fem-faktor	HM	0.002102047
Forecast	Capm	HM	-0.115525153
Forecast	Tre-faktor	HM	-0.112256637
Forecast	Fem-faktor	HM	-0.115377032
Træning	ARIMA	HM	0.0557
Forecast	ARIMA	HM	-0.495
Træning	Avis	HM	-0.084971914
Forecast	Avis	HM	-0.059570434

5.7 Evaluering af forecast

I dette afsnit skal der evalueres på, hvordan modellerne har klaret sig i forhold til hinanden. Tallene i tabel 5.7 er regnet ud fra ligningerne i afsnit 3.2, og viser kort de forskellige metrics for både trænings- og forecastsættet. Der skal lige noteres, at textmining er på månedsbasis, og resten er på dagsbasis. Når der ses på alle tallene overordnet, er det tydligt at se, at RNN-modellen vinder på alle parametre, både på trænings- og på forecast-delen. Det samme forklares af r^2, som næsten er på 100% for både trænings- og forecastdelen, dog med et mindre fald fra træning til forecast, men ikke noget af betydning i forhold til de andre modeller. På den dårlige side, findes ikke overraskende modellen med avisoverskrifter, som er dårlig i både træning og forecast, og ikke overraskende er ARIMA-modellen også noget bagud, men kun i træningsdelen. Det der er overraskende er, at den er bedre i forecast-delen end i træningsdelen, når der ses på r^2. Umiddelbart ser det meget mærkeligt, men ses der på figur 5.12, så præsterer den ikke så dårlig i forecastfejl i forhold til de andre i perioden. Men det er mærkelige tal for ARIMA, og har endnu ikke kunnet finde forklaringen på den høje r^2 i forecastdelen.

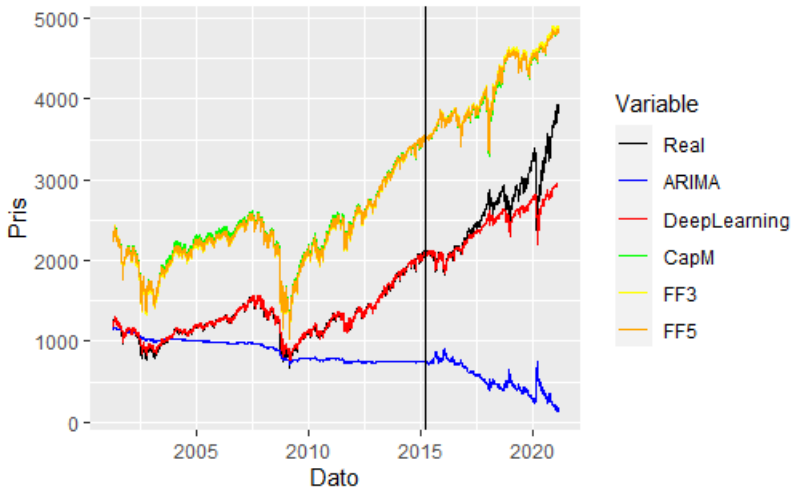
Tabel 5.7: Metricstabel

Del	Metric	ARIMA	DeepLearning	CapM	FF3	FF5	Textmining
Træning	RMSE	536.5131730	31.7682985	1089.8815407	1053.804334	1068.8469190	214.5412918
	MAE	410.9269061	22.9604241	1070.5766203	1032.561615	1048.6090106	146.1176087
	r^2	0.1706275	0.9896045	0.9572947	0.963806	0.9613951	0.4474316
Forecast	RMSE	2200.5532978	303.4644696	1518.8821736	1553.7997144	1527.4349491	342.83405733
	MAE	2098.2621644	213.8689080	1499.4785382	1536.9253084	1509.7648578	313.11217857
	r^2	0.9686404	0.9468024	0.7506517	0.7779073	0.7723028	0.02346202



Figur 5.12: Akkumuleret fejlled for forecastperioden

Ses der så på figur 5.13, som viser modelleringerne for alle modeller, undtagen modelleringen af avisoverskrifter, da det er på månedsbasis, og ikke passer med de andre, så kan det ses, at ARIMA-modellen overhovedet ikke flugter med de andre eller den rigtige pris. Noget der også blev konstateret med tabel 5.6 for markedstiming for forecast-delen.



Figur 5.13: Prisgraf for alle modeller undtagen avis-modellen

I tabel 5.8 ses resultaterne for Diebold-Mariano-testen, og nulhypotesen er, at de har samme forecastaccuracy, og alternativhypoteserne i tabel 5.8 er henholdsvis, at de har forskellig accuracy, at de er mindre præcise, eller til sidst om de er mere præcise end benchmark-modellen, som er RNN-modellen. Forecasthorisonten er 1489 dage, fra 1. april 2015 til d. 26.

februar 2021. Når benchmarkmodellen testes i mod ARIMA og tre-faktor-modellen, gives en p-værdi på 100% og 50%, når der testes for, hvilken der er større. Det bevirker, at der rent statistisk ikke er grundlag for, at den ene har bedre forecast end den anden. Herefter testes der med textmining overfor benchmark. Da det giver en p-værdi på 100%, så accepteres nulhypotesen, og konkluderes, at benchmarken har lavere forecastingsfejl end avis-modellen. Ved beregning af tallene for textmining, er resultaterne fra deep learning lavet om, så det kun er den sidste dag i måneden, der bliver handlet, så de kan sammenlignes.

Stat	Alternativ hypotese	ARIMA	FF3	Textmining
P-værdi	DL=Model	1	1	<2.2e-16
	DL<Model	0.5	0.5	<2.2e-16
	DL>Model	0.5	0.5	1

Tabel 5.8: Diebold-Marino med LSTM-modellen som benchmark

Til sidst skal der vurderes, om modellerne rent faktisk kan tjene penge, når det kommer til forecastperioden. Det skal forudsættes, at disse modeller blev lavet d. 1. april 2015, og satte den i gang med at købe de dage, modellerne mente, at de næste dage ville have en opadgående trend, og solgte de dage, den vurderede, at næste dage ville prisen falde. Hvis prisen steg eller faldt flere dage i træk, blev der ikke foretaget noget. Resultaterne for hver model ses i tabel 5.9. Der er dog ikke taget transaktionsomkostninger såsom kurtage med i dette. Den viser antal køb og salg, profit for hele perioden og andre statistiske parametre. Der kan der ses på positivprocenten, som viser den del af handlerne, der er tjent penge, og her er det avis-modellen, som har fået den højeste procentsats, men kigges der på, at den kun har foretaget syv handler, er det ikke prængende, men den har også kun haft på månedsbasis, og den har da også det største gennemsnitlige afkast til trods for, at det blev vist i tabel 5.8, at RNN-modellen havde statistisk lavere forecastfejl. Men ses der kun på afkast, så vinder RNN-modellen over avis-modellen, da den har et højere afkast over hele perioden. ARIMA-modellen har til gengæld fejlet stort, da den har haft 397 handler, men har en positivprocent på 0. Det kan der også ses på tallene, hvor det er minus hele vejen ned. Det er til trods for, at den har været noglelunde tæt på de andre modeller, når det handler om test af retningen, se tabel 5.5 og forecastnøjagtighed i tabel 5.8. Ses der på de sidste tre modeller, de tre fra Farma&French, så vurderes de til at være gode, og de har også næsten lige så mange handler som RNN-modellen men til trods for det har de et lavere afkast for hele perioden, har et lavere gennemsnitligt afkast for hver handel, har en højere standardafvigelse og en lavere sharperatio. For Farma&French's vedkommende ligner de

meget hinanden. Men den bedste af de tre fra Farma&French er tre-faktor-modellen med et højere afkast, både for perioden og fra dag til dag, den har en højere sharperatio, og har en højere positivprocent. Så ARIMA-modellen fejler stort, også er det RNN-modellen, som er bedst, sammen med tre-faktor-modellen, til at tjene penge.

Tabel 5.9: Algoritmehandel for de seks modeller

	Deep learning	AvisTextmining	ARIMA	CapM	Tre-faktor	Fem-faktor
Antal køb	395.000	7.00	397.00000	390.000	390.000	391.000
Antal salg	395.000	7.00	397.00000	390.000	390.000	391.000
Antal transaktioner	790.000	14.00	794.00000	780.000	780.000	782.000
Profit i procent over hele perioden	38.528	10.52	-516.35227	17.984	30.767	24.661
Største tab	-22581.000	-13166.00	-43193.00000	-26592.000	-26592.000	-26592.000
Største tab i procent	-8.222	-6.68	-16.18897	-10.720	-10.720	-10.720
Største vinder	20993.000	16038.00	-2.00000	23816.000	23816.000	23816.000
Største vinder i procent	8.578	7.71	-0.00069	9.620	9.620	9.620
Gennemsnitligt beløbsafkast	315.818	3304.71	-3385.79597	144.308	245.005	199.719
Gennemsnitligt afkast i procent	0.098	1.50	-1.30064	0.046	0.079	0.063
Median-beløb	231.000	3722.00	-1634.00000	369.500	414.000	344.000
Median-afkast i procent	0.094	1.77	-0.59401	0.152	0.168	0.147
Standardafvigelse for afkastsbeløb	3933.978	8946.12	5345.09363	4174.046	4159.876	4018.218
Standardafvigelse for afkast i procent	1.485	4.40	2.04741	1.596	1.566	1.539
Sharpe-ratio	0.066	0.34	-0.63526	0.029	0.050	0.041
Antal vindende handler	212	5	0	222	225	224
Antal tabte handler	183	2	397	168	165	167
Positiv_procent	53.671	71.43	0.00	56.923	57.692	57.289

KAPITEL 6

Vurdering

Tilbage i aktieteorien blev der startet med et afsnit om den efficiente markedshypotese, 1.1, som mener, at priserne i markederne altid fuldt ud reflekterer alt tilgængelig information. Dermed menes der, at det gennemsnitlige afkast på tiden t+1 på basis af informationen Φ_t er lig 0, og at de opfører sig som random walks. Men senere teorier i kapitel 1 slår fast at der har været en del teorier, som er gået imod denne teori, og har givet nogle gode afkast. Det har de også, men som (Granger, 1992) pointerer, så bliver teorierne ubrugelige, når først de er publicerede, som han viser en del eksempler på i kilden. Så skal man lade være med at publicere og tjene pengene i stedet? Det kan være, at der er en, som kommer med samme ideer i en anden bog kort tid efter, også kan man hverken tjene penge på aktiemarkedet med sine resultater og analyser og heller ikke på salg af den nyligt udgivet bog. Så den bedst mulige måde at tjene penge er vel at udgive bogen, og tjene på det. At man ikke kan forudsige fremtiden, mente Keynes også i hans bøger og artikler. Her er fire citater fra ham, hvor han kommenterer på netop de emner (Shuler, 2020):

"Markets can remain irrational longer than you can remain solvent."

"If farming were to be organised like the stock market,
a farmer would sell his farm in the morning when it was raining,
only to buy it back in the afternoon when the sun came out."

"The markets are moved by animal spirits, and not by reason."

"Long run is a misleading guide to current affairs. In the long run we are all dead."

Disse citater fra Keynes fortæller om en mere i rækken, som ikke tror på, at der kan tjenes penge på aktiemarkedet, og at markedet er efficient. Egentlig tror Keynes mindre på det ud fra citaterne, da afsnit 1.1 mente, at investeringer dog godt kunne gå i 0 eller i plus, hvis man tog risikoen. Men Keynes går skridtet videre, og siger hold dig væk, ellers bliver man insolvent. Man skal holde sig til korte og sikre investeringer, eller holde sig helt væk, for

usikkerhed er ikke noget, der kan modelleres på lang sigt. Som han siger, på lang sigt er vi alle døde. Men er disse teorier overhovedet relevante mere, især når der bliver foretaget handler med statistiske modeller og algoritmer eller med DowTheory (Choice, 2020) som teknisk analyse?
I denne rapport er der netop prøvet at bruge statistik og økonometri til at forsøge at forecaste den fremtidige pris, og dermed gå imod den tese, at intet kan modelleres. Det er blevet til seks forskellige modeller, som er to, hvor der er brugt RNN- og LSTM-modeller til at modellere prisen, og hvor de andre er modelleret ud fra traditionel modellering med ARIMA og de tre fra Farma&French. I alle modeller er der konstateret, at det går bedre med at modellere træningssættet end for forecastsættet, hvilket ikke er ret mærkeligt, da det er det stykke, den modellerer ud fra.

6.1 Vurdering ud fra de forskellige metrics

Som nævnt så vil modellerne altid være bedst på træningssiden end på forecastsiden, og det er også tilfældet for de forskellige metricses, som bruges i denne rapport. Startes der med retningsvariablen, tabel 5.5, så bliver alles tilfælde i retningsvariablen dårligere fra trænings- til forecastsættet, og for de fleste modeller, modelleres retningen til at være omkring de 50% sikre med undtagelse af træningssættet for CapM, Tre-faktor og Fem-faktor, som er over de 95%, men falder så på niveau med de andre i forecastet. Fordi det er forecastet, som er vigtigst, så vil tre-faktor-modellen være bedst. Det er ikke tilfældet med markedstimingen,se tabel 5.6, som egentlig dumper de tre Farma&French med under 0. Udover RNN-modellen er det kun tilfældet for træningsdelen, hvor der vurderes at være markedstiming, i de tre Farma&French-modeller. Til gengæld er der en klar vinder med RNN-modellen i markedstiming, da både trænings- og forecastdelen klart er bedst. Så vurderes der udelukkende på markedstiming, vinder RNN-modellen klart. Det samme gør sig gældende i RMSE, MAE og r^2, se tabel 5.7, hvor RNN-modellen er meget bedre end alle de andre, både når det kommer til trænings- og forecastdelen. Selvom Farma&French-modellerne gør det meget godt med hensyn til r^2 i træningssættet, så dykker det meget, når der fortsættes til forecastdelen. Med hensyn til ARIMA-modellens tal ser det meget mærkeligt ud, at den præsterer så godt i forecast-delen, når der ses på r^2, både når den scorer så lavt i træningssættet, samtidig med både standardafvigelsen, MSE, og gennemsnittet, MAE, for fejlledene stiger så kraftigt fra træning til forecast. Derfor må der være sket en fejl et sted, som ikke har været muligt at finde.
Hvis der skal vurderes samlet set på disse tal, rangeres de på, hvordan de klarer sig overfor hinanden, indenfor hver metric i tabel 6.1. Tabellen kan ikke bruges til at konkludere, da

der kan være meget stor forskel på, hvordan de forskellige metrics er spredt, og derfor viser tabel 6.1 en meget forsimplet rangering af modellerne. Men ud fra tabellen, kan der alligevel godt vises et mønster for, at RNN-modellen, den ikke-lineære model, kan vurderes til at være en af de bedre modeller i denne rapport. Der er dog en enkelt parameter, som den ikke har kunnet modellere, og det er retningen. Det kan skyldes, at den ikke modellerer afkastene men prisen.

Tabel 6.1: Rangering af modellerne

Del	Metric	ARIMA	DeepLearning	CapM	FF3	FF5	Textmining
Træning	RMSE	3	1	6	4	5	2
	MAE	3	1	6	4	5	2
	r^2	6	1	4	2	3	5
	Retningsvariabel	5	6	3	2	1	4
	Markedstiming	2	1	3	4	5	6D
	SUM for træning	19	10	22	16	19	19
Forecast	RMSE	6	1	3	5	4	2
	MAE	6	1	3	5	4	2
	r^2	1	2	5	3	4	6
	Retningsvariabel	6	4	5	2	3	1
	Markedstiming	6D	1	5D	3D	4D	2D
	SUM for forecast	25	9	21	18	19	13
	Summen i det hele	44	19	43	34	38	32

6.2 Stemmer resultaterne overens med den efficiente markedshypotese?

Fra afsnit 1.1 og det første stykke i denne vurdering er der argumenteret for, at man vil ende op med et afkast på 0% medmindre der tages ekstra risiko. Den efficiente markedshypotese argumenterede også for, at alt information bliver inkorporeret i markedet på ingen tid. Derfor skulle det dermed ikke være muligt at kunne få et afkast med avisoverskrifter. Der skal selvfølgelig tages forbehold for, at denne model med avisartikler er på månedsbasis, og at den ikke har lavet så mange handler som de andre. Derfor kan der heller ikke blive sammenlignet en til en, fordi de har forskellige forudsætninger.

De seks modeller har dog også nogle store underskud med sig i nogle handler, men det her forecast er også rigtig lang. Sådan en længde vil ikke være det mest optimale, hvis de skulle bruges i virkeligheden. Det kan der specielt ses på figur 5.12, som ikke overraskende viser en større fejlled, som tiden går. Derfor vil det også være nemmere at afvise den efficiente markedshypotese, hvis der bliver lavet et mindre tidsperspektiv. Hvis man aktivt selv lavede alle disse forecast hver dag, og handlede ud fra dem, vil resultaterne nok også se bedre ud,

Tabel 6.2: Algoritmehandel for de seks modeller

	Deep learning	AvisTextmining	ARIMA	CapM	Tre-faktor	Fem-faktor
Antal køb	395.000	7.00	397.00000	390.000	390.000	391.000
Antal salg	395.000	7.00	397.00000	390.000	390.000	391.000
Profit i procent over hele perioden	38.528	10.52	-516.35227	17.984	30.767	24.661
Største tab i procent	-8.222	-6.68	-16.18897	-10.720	-10.720	-10.720
Største vinder i procent	8.578	7.71	-0.00069	9.620	9.620	9.620
Gennemsnitligt afkast i procent	0.098	1.50	-1.30064	0.046	0.079	0.063
Sharpe-ratio	0.066	0.34	-0.63526	0.029	0.050	0.041
Antal vindende handler	212	5	0	222	225	224
Antal tabte handler	183	2	397	168	165	167
Positiv_procent	53.671	71.43	0.00	56.923	57.692	57.289

og blev det gjort med den metode, kan disse forecast også bedre ses som de er, modelleringer. Modelleringer, som aldrig må veksles med sandheden, og altid skal bruges med den forsigtighed, der er nødvendig. Skal man aktivt sidde med disse modeller, så er der også større chance for, at man kan stoppe i tide, inden de store tab kommer. Men da der bliver lavet afkast på fem af dem, vil det være et godt argument at bruge og sige, at den efficiente markedshypotese ikke gælder i dag.

6.3 Kan avisoverskrifter forudsige aktieprisen?

Når der skal handles med aktier, skal man forstå, hvad der driver fluktueringerne i dem. Professor Bryan Kelly med flere har prøvet at forstå fluktueringer i økonomiske nøgletal med forretningsnyheder. I den forbindelse skriver han i konklusionen (Kelly et al., 2020, p. 39):

Our approach is motivated by the view that news text is a mirror of the state of the economy. The media sector is an information intermediary that meets information demand of consumers and investors with verbal descriptions of economics events and their interpretation. Of course, viewing news media as a verbal mirror of the economy, we must also regocnize that it comes with flawed refractions, often in the form of producer and consumer biases, erroneus inference, and unsubstantiated speculations.

Her snakker han om, at nyheder er et spejl på økonomien, og at medierne giver beskrivelser af økonomiske begivenheder, som de gerne vil have det til at fremstå, og som giver forkerte konklusioner på både virksomheders og forbrugeres side, og som giver grobund for spekulationer, der ikke har hold i virkeligheden. Men ikke desto mindre er det den virkelighed, der bliver fortalt. I hans konklusioner fremgår det, at nyheder er med til at bestemme den

fremtidige økonomiske virkelighed, og på samme måde er der prøvet i denne afhandling at skabe lys for, om de også kan forudsige fremtidig aktiekurs.
Denne model har været svær at vurdere, hvilken metode og hvilke værktøjer, der har skullet bruges i denne forbindelse. Skulle det være en VAR-, OLS- eller en RNN-model. Mulighederne har været mange, og det endte med en RNN-model, da den gav de bedste resultater for både træningsdelen og forecastet, men når den er lavet på månedsbasis, og de andre er lavet på dagsbasis, er det svært at give et entydigt svar på, om den kan være lige så god som de andre modeller. Man kan se på tabel 6.2, at de andre modeller også er nogenlunde enige om, hvor mange transaktioner, der skal være i forecastperioden, og det giver en nem måde at vurdere, hvilke metoder der er mest profitable. Man kunne selvfølgelig have lavet alle modeller på månedsbasis, men det havde så givet problemer for de andre modeller, da de så ville have meget mindre datagrundlag end avismodellen, da de andre kun har tidsserie af aktien at forholde sig til for RNN- og ARIMA-modellernes vedkommende og et til fem variable mere til Farma&French-modellerne, mens avismodellen har tidsserien plus 180 andre variable. Men når det så er pointeret, så giver avismodellen et positivt afkast, og derfor er den relevant i den forstand, den er profitabel. Det mest optimale havde været enten, at der var et datagrundlag, som havde avisoverskrifterne på dagsbasis og som rå data, i stedet for som nu, hvor artiklen (Kelly et al., 2020) kun giver et forarbejdet data til rådighed. Hvis det rå data havde været til stede og på dagsbasis, havde modellen haft tidsserien og sikkert med 10.000 variable af ord, som skulle danne grobund for en mere omfattende analyse af profitabilitetsanalysen for Avismodellen. Så havde der været grundlag for at bruge bl.a. embedding-layers og mange flere ord (Chollet and Allaire, 2018b). Men når datagrundlaget har været, som det er, er det gode takter, at den giver et positivt afkast, og ud fra tabel 6.2 er den profitabel, men er på et meget spinkelt grundlag i forhold til de andre modeller.

6.4 Kan modellerne stå alene?

Hvis man vil bruge modelleringsværktøjer til at forudsige aktiemarkedet, skal det kun være en del af værktøjerne. Hvis der ses på Farma&French-modellerne, viser det, at informationer på fortiden også kan bruges til at modellere fremtiden. Det er er netop et bevis på, at virksomheders generelle oplysninger kan bruges til at forudsige aktier. Men disse oplysninger er ikke de eneste, da ting såsom dividendeudbytte fra virksomhedens side kan være en strategi fra virksomhedens side til at forhøje aktieprisen, hvis virksomheden har brug for en højere pris, jvf afsnit 1.4.1. Dermed kan en dividendeudbytte også være en fælde, men uanset om det er eller ej, så vil den udbetaling ikke kunne vises på de forskellige modeller, da det ikke er inkorporeret. Der kan også ske noget uforventet, som kommer til at sende chokbølger

på aktiemarkedet, og får tingene til at falde. Derfor så kan en del af aktiepriserne godt modelleres, men der er en masse eksogene chok og små variable, der kan ændre kursens retning. Så modellerne kan bruges, men kun som en guideline for kursen.

Konklusion

I dette speciale er der forklaret forskellige aktieteorier, som starter med den efficiente markedshypotese, som handler om, at medmindre man tager overhængende stor risiko, så vil man få et gennemsnitligt afkast på 0, da alt information allerede er inkorporeret i markedet. Det er der kommet forskellige analyser ud af, hvordan man netop kan få et positivt afkast ud fra forskellige mønstre i form af nogle få og små aktieteorier og derefter økonometriske værktøjer, der har haft til formål at modellere aktieprisen for S%P-500-indekset fra 2001 til 2015, og som har skullet lave et in-of-sample-forecast på indekset for hver økonometrisk metode. I den forbindelse er der kommet seks forskellige analyser, hvoraf de fem af dem har været på dagsbasis, og den sidste har været på månedsbasis grundet et manglende datagrundlag. I den forbindelse er der blevet forklaret, hvordan hver model er blevet til, og hvilke værktøjer der har været til at forbedre modellerne, og derudover er der også vurderet på profitabilitetten på hver model. Her imponerer RNN-/LSTM- og Farma&French- modellen meget, som er blevet til på vidt forskellige måder. Farma&French er blevet til via en fundamental analyse, som har haft mellem et til fem parametre at forholde sig til, mens DeepLearning-modellen kun er blevet til ved indeksprisen dag for dag, og har skullet finde mønstre ud fra de værktøjer, der er i DeepLearning-universet. Derfor er de to modeller meget forskellige, men alligevel har de været gode til at forecaste prisen på en tidshorisont, som har været rigtig lang i forhold til den store volatilitet, der kan være på aktiemarkedet. Derfor skal man også passe på ikke at lægge for meget i dem, da det kun kan være guidelines, og man altid skal holde øjne og ører åbne overfor eksogene chok og bare små hverdagsforandringer, som kan give stød til aktiemarkedet. Det vil snarere være hjælpeværktøjer end spåkonetrolddom og forudsigelser, fordi verdenen er mere kompleks end man kan modellere, men man kan altid give et godt og kvalificeret bud. Når de forbehold er taget, er det RNN-modellen og Farma&French, der giver mest profitabilitet, når det er på dagsbasis.

Litteratur

Amadeo, K. (2020). The s&p500 and how it works. Retrieved from https://www.thebalance.com/what-is-the-sandp-500-3305888.

Belyh, A. (2019). The definitive guide to fama-french three-factor model. Retrieved from https://www.cleverism.com/fama-french-three-factor-model-guide/.

Breaking Down Finance (n.d.). Market timing. Retrieved from https://breakingdownfinance.com/finance-topics/performance-measurement/market-timing/.

Chauhan, N. S. (2019). What is hierachical clustering? Retrieved from https://www.kdnuggets.com/2019/09/hierarchical-clustering.html.

Choice (2020). Dow theory - a fundamental part of technical analysis. Retrieved from https://choicebroking.in/blog/dow-theory.

Chollet, F. and Allaire, J. J. (2018a). *Chapter 3 - Getting started with neural networks*, pages 50–83. Deep Learning with R. Manning, Shelter Island.

Chollet, F. and Allaire, J. J. (2018b). *Chapter 6 - Deep learning for text and sequences*, pages 164–218. Deep Learning with R. Manning, Shelter Island.

Chugh, A. (2020). Mae, mse, rmse, coefficient of determination, adjusted r squared - which metric is better? Retrieved from https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326

Ciaburro, G. and Venkateswaran, B. (2017). *Neural Networks with R*. Packt Publishing Ltd., Birmingham - Mumbai.

Corporate Finance Institute (n.d). Price earnings ratio. Retrived from https://corporatefinanceinstitute.com/resources/knowledge/valuation/price-earnings-ratio/ at 18/02/2021.

Enders, W. (2015). *Applied Econometric Time Series*. Wiley, University of Alabama, 4. edition.

Erdinç, Y. (2017). *Chapter 4 - Comparison of CAPM, Three-Factor Fama-French Model and Five-Factor Fama-French Model for the Turkish Stock Market*, pages 69–93. Financial Management from an Emerging Market Perspective. InTech, Rijeka, Kroatien.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal og Finance*, 25(2):383–417.

Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56.

Fama, E. F. and French, K. R. (2014). A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22.

Fernando, J. (2021). Sharpe ratio. Retrieved from `https://www.investopedia.com/terms/s/sharperatio.asp`.

Granger, C. W. J. (1992). Forecasting stock market prices: Lessons for forecasters. *International Journal of Forecasting*, 8(1):3–13.

Ingemann, J. H. (2014). *Videnskabsteori - For økonomi, politik og forvaltning*. Samfundslitteratur, Frederiksberg, København, 1. udgave, 2. opslag edition.

Jespersen, J. (2009). *Chapter 2 - Macroeconomic methodology: from a critical realist perspective*, pages 53–90. Macroeconomic Methodolgy - A Post-Keynesian Perspective. Edward Elgar Publishing Limitted, Cheltenham, UK; Northahmpton, MA, USA, 2011 edition.

Kelly, B. (2021). Retrieved from `https://www.bryankellyacademic.org/`.

Kelly, B., Bybee, L., Manela, A., and Dacheng, X. (2020). The structure of economic news.

Kenton, W. (2021). Capital asset pricing model (capm). Retrieved from `https://www.investopedia.com/terms/c/capm.asp`.

Kyrykovych, A. (2020). Deep neural networks. Retrieved from `https://www.kdnuggets.com/2020/02/deep-neural-networks.html`.

Little, K. and Mansa, J. (2020). Understanding dividend yield. Retrieved from `https://www.thebalance.com/understanding-dividend-yield-3140782` at 18/02/2021.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1):59–82.

Mcclure, B. (2021). Using the price-to-book (p/b) ratio to evaluate companies. Retrieved from `https://www.investopedia.com/investing/using-price-to-book-ratio-evaluate-companies/`.

Musarurwa, R. (2019). Fama french five factor asset pricing model. Retrieved from `https://blog.quantinsti.com/fama-french-five-factor-asset-pricing-model/`.

Nielsen, M. A. (2015). Neural networks and deep learning. Retrieved from `http://neuralnetworksanddeeplearning.com/index.html`.

Pesaran, H. and Timmermann, A. (1992). A simple nonparametric test of predictive performance. *Journal of Business and Economic Statistics*, 10(4):461–465.

Pesaran, M. H. and Timmermann, A. G. (1994). A generalization of the non-paramtric henriksson-merton test of market timing. *Elsevier Science*, 44(1-2):1–7.

Shuler, A. (2020). 7 insigthfull keynes quotes about economics and the stock market. Retrieved from `https://einvestingforbeginners.com/keynes-quotes-ansh/`.

Zaiontz, C. (2021). Diebold-mariano test. Retrieved from `https://www.real-statistics.com/time-series-analysis/forecasting-accuracy/diebold-mariano-test/`.