Sammanfattning

Abstract.

Nyckelord: keywords

Denna uppsats är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Johan Selberg

Johannes Bandgren

Godkänd,

Handledare: Kerstin Andersson

Examinator: Exam

Tacksägelser

Thanks.

Karlstad Universitet, 9 april 2018 Johan Selberg och Johannes Bandgren

Innehåll

Figurer

Tabeller

Kapitel 1

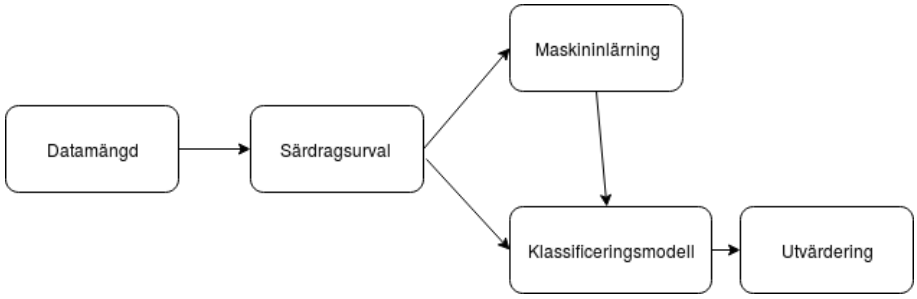
Introduktion

Kapitel 2

Bakgrund

Syftet med studien är att utvärdera tre stycken klassificeringsmodeller inom maskininlärning och hur bearbetningen samt märkningen av en datamängd påverkar en klassificeringsmodells prestanda. Maskininlärningsalgoritmerna som kommer att användas för att få fram klassiciferingsmodellerna är Naive Bayes (NB) och Support Vector Machine (SVM). Den tredje klassificeringsmodellen som tas fram kommer att vara en djuplärande modell som representeras av ett neuronnät, mer specifikt ett Convolutional neural network (CNN). Figur 2.1 visar en förenklad bild av hur processen för att utvärdera klassificeringsmodellerna kommer att se ut. Datamängden består av en mängd indata som är märkt med förväntad utdata, som går igenom en bearbetningsprocess där t.ex. data som inte tillför något värde tas bort och där mängden av data delas upp i tränings- och valideringsdata. Bearbetningsprocessen diskuteras mer ingående i 3.1. På den bearbetade datan utförs ett särdragsurval, där dataspecifika särdrag plockas ut... MER .. som diskuteras vidare i 3.2. Träningsmängden används av maskininlärningsalgoritmen för att skapa klassiciferingsmodellen. När klassificeringsmodellen är framtagen används valideringmängden för att utvärdera prestandan av modellen.

Studien kommer att använda maskininlärning för att utföra sentimentanalys (SA) d.v.s utläsa huruvida en text uttrycker någonting positivt eller negativt. Uppdragsgivaren, CGI, betraktar sentimentanalys som en viktig pusselbit för framtida lösningar man vill erbjuda sina kunder. Lösningar skulle kunna vara chatbotar där



Figur 2.1: Utvärderingsprocessen av maskininlärningsmodeller

sentimentanalys används så att chatboten ändrar sitt språk utefter svaren från slutanvändare. Eller att det kan användas för trendanalys där ett företag vill veta vad allmänheten tycker före och efter att man har släppt en ny produkt eller efter att kvartalsrapporten har släppts.

I avsnitt 2.1 förklaras vad SA är och olika klassificeringstekniker inom SA berskrivs kortfattat. I avsnitt 2.2 ges en kort introduktion till maskininlärning och framstegen som gjorts under den senaste tiden. Dessutom beskrivs NB och SVM samt hur inlärningsprocessen ser ut. I avsnitt 2.3 ges en överblick över hur en datamängd ser ut och vilka märkningsmetoder som används för twittersentimentanalys (TSA). Avsnittet beskriver även de valda datamängderna. I avsnitt 2.4 ges en introduktion till TSA och vilka utmaning samt problem TSA har. Slutligen i avsnitt 2.5 förklaras vilka hjälpmedel som kommer att användas för att utvärdera algoritmernas prestanda.

2.1 Sentimentanalys

SA används för att studera människors åsikter, attityder och känslor mot olika entiteter. En entitet kan vara ett ämne, en händelse eller en individ. Målet med SA är att identifiera känslan som är uttryckt i en text för att därefter analysera den. Processen delas upp i tre steg: att hitta entiteter, identifiera känslan för de entiteterna och slutligen klassificera dessa entiteter. [1].

Inom SA appliceras klassificeringen på 3 olika nivåer: dokument-, menings- och aspektnivå [1]. SA på dokumentnivå klassificerar om ett helt dokument uttrycker en

positiv eller negativ åsikt, exempel på dokument kan vara produktrecensioner eller nyhetsartiklar. Medan på meningsnivå klassificeras varje mening i ett dokument. Slutligen nere på aspektnivå analyseras de möjliga aspekterna av en entitet. En mening kan behandla olika aspekter av en entitet. Både positiva och negativa åsikter kan delges om en entitets olika aspekter. Ett exempel på det är meningen "Ölen var väldigt god, men tyvärr alldeles för dyr", som innehåller både en positiv och en negativ åsikt om en entitet som i det här fallet är ölen.

2.1.1 Klassificeringstekniker

De olika klassificeringsteknikerna som i nuläget används för SA delas upp i 3 olika kategorier: maskininlärningsmetoder, lexikonbaserade metoder samt hybrida metoder [1].

Maskininlärningsmetoder använder etablerade maskininlärningsalgoritmer tillsammans med språkliga särdrag för att konstruera klassificerare som kan avgöra om en text uttrycker någonting positivt eller negativt [ref=Sentiment analysis a survey]. Prestandan för maskininlärningsmetoder är beroende av mängden träningsdata, större mängder data ger vanligtvis bättre resultat [2]. Maskininlärningsmetoder är dessutom domänberoende, vilket gör att de inte presterar bra när de används på andra domäner än det de har tränats på.

Lexikonbaserade metoder använder sig av ordlistor för att analysera text [1]. Ordlistorna består av positiva och negativa termer som används för att beräkna vad en given text uttrycker för sentiment. Fördelen med lexikonbaserade metoder är att de inte kräver någon träningsdata [2]. Men faktumet att de är beroende av statiska ordlistor betyder att de inte tar hänsyn till termer som inte finns i ordlistorna. Det betyder att ordlistor måste uppdateras kontinuerligt för innehåll som är dynamiskt och ständigt under utveckling, vilket är särskilt problematiskt för texter som förekommer på sociala medier.

Hybrida metoder kombinerar lexikonbaserade metoder med maskininlärningsmetoder [2]. Genom att kombinera metoderna med varandra kan de väga upp för varandras svagheter. Nackdelen med hybrida metoder är dock att de kräver en hög beräkningskomplexitet.

Experimentet som presenteras i studien fokuserar enbart på maskininlärnings-

metoder.

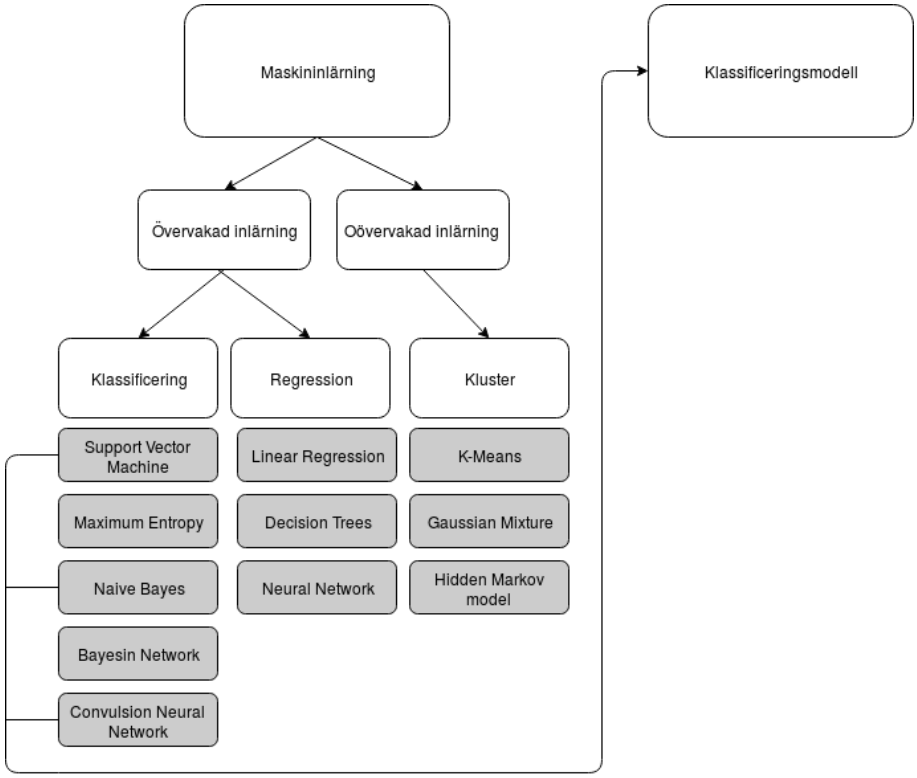
2.2 Maskininlärning

Maskininlärning är ett delområde inom artificiell intelligens (AI), där målet är att göra det möjligt för datorer att lära sig på egen hand. Maskininlärningsalgoritmer gör det möjligt att identifiera olika mönster från observerad data, bygga upp en generell modell som kan förutsäga saker utan att ha blivit förprogrammerade med explicita regler för hur den ska lösa ett problem. Under de senaste åren har stora framsteg inom maskininlärning gjorts. Exempelvis utvecklade DeepMind [3] 2015 en agent som mästrade 49 st Atari-spel [4], med en klassificeringsmodell med endast pixlar och spelpoäng som indata. Under 2016 utvecklade DeepMind sin AlphaGo [5] agent som besegrade en av världens bästa Go spelare, Lee Sedol [6] med 4-1 i matcher. Detta var ett framsteg för AI eftersom Go är ett komplext spel med $2 * 10^{170}$ möjliga drag [7].

Maskininlärning används också för att lösa vardagliga problem. I dagens mobiltelefoner finns en så kallad intelligent personlig assistent, även kallade Siri [8] och Google Assist [9], där användarna får hjälp med t.ex. “vad är det för väder idag?”, “Skicka ett meddelande till mamma att jag blir sen till middagen idag” och “Påminn mig imorgon att jag måste köpa mjölk”. Facebook använder även maskininlärning för ansiktsigenkänning [10] på bilder som laddas upp, och för att rekommendera nya vänner [11].

Figur 2.2 illustrerar att maskininlärning delas upp i två typer av inlärningsprocesser dvs. i övervakad eller oövervakad inlärning och deras underliggande algoritmer.

Oövervakad inlärning används när datamängden bara består av indata och inget förväntat resultat. Målet med oövervakad inlärning är att algoritmen själv lär sig att modellera den komplexa underliggande strukturen så att den kan lära sig mer om datan och själv komma fram till ett resultat [12]. Exempelvis kan en oövervakad klusteralgoritm användas för att hitta likheter i bilder och följaktligen gruppera dem. Övervakad inlärning används när datamängden består av både indata och dess förväntade utdata. Den övervakade algoritmen använder datamängden för att

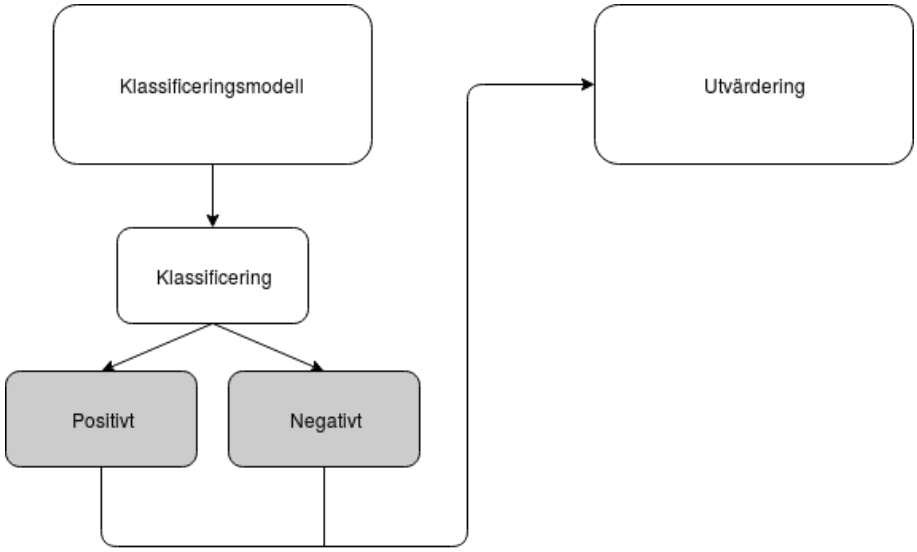


Figur 2.2: Inlärningsprocesser för maskininlärning.

lära sig hur utdata beror på indata genom att skapa en klassificeringsmodell som används för att förutse utdata från ny indata som illustreras i figur 2.3.

Övervakad inlärning kan man tänka sig som att en lärare övervakar programmets inlärningsprocess. Under inlärningsprocessen försöker algoritmen iterativt förutse utdata från datamängden och blir rättad av läraren vid fel förutsägelse [13].

Övervakad inlärning kan brytas ner till klassificerings- och regressionsproblem och eftersom TSA kan kallas ett typiskt klassificeringsproblem [1], kommer studien att använda algoritmer lämpade för klassificeringsproblem. I figur 2.2 kan vi se några av dessa klassificeringsalgoritmer och enligt [2] är speciellt Naive Bayes (NB) och Support Vector Machine (SVM) bäst lämpade för TSA.



Figur 2.3: Klassificeringsprocess av data.

2.2.1 Naive Bayes

NB är en klassificeringsalgoritm som är baserad på Bayes theorem [14] med starka ("naive") oberoende antaganden mellan särdragen d.v.s NB förutsätter att närvaron av ett visst särdrag i en klass inte relaterar till närvaron av ett annat särdrag. Exempelvis kan en frukt anses vara ett äpple om det är grönt, runt och är 10 cm i diameter. Även om särdragen grönt, runt och diameter kan bero på varandra så bidrar alla särdragen självständigt till sannolikheten att frukten är ett äpple, det är därför algoritmen kallas ("naive") [15].

2.2.2 Support Vector Machine

SVM är en övervakad maskininlärningsalgoritm som kan användas till både klassificerings- och regressionsproblem, för det mesta används SVM för klassifikationsproblem. SVM är baserat på idén att hitta ett hyperplan [16] som bäst delar upp datamängden i två klasser [17]. Givet en träningsmängd där förväntad utdata är markerad till en av två kategorier bygger SVM-algoritmen upp en klassificeringsmodell som kan användas för att förutse vilken kategori ny indata ger [18].

2.2.3 Convolutional neural network

Ett av de snabbast växande områden inom maskininlärning är djup inlärning [2]. Begreppet djup inlärning syftar till artificiella neuronnät (ANN) som är uppbyggda med flera lager [19]. ANN är system som försöker ta efter beteende hos biologiska neuronnät för att stegvis bli bättre på att utföra en angiven uppgift genom att studera exempel [20].

CNN är ett av de mest populära neuronnäten [19]. Det var ursprungligen framtaget för maskininlärningsproblem som rör datorseende och har tidigare påvisat imponerande resultat för bildigenkänning [21, 22]. Men på senare tid har det även visat sig att CNN är effektivt att använda för problem som rör sentimentanalys. Med relativt enkla CNN-modeller har imponerande resultat kunnat uppnås inom textklassificering. I [21] presenteras en enkel CNN-modell för textklassificering som påvisade goda resultat över flertalet datamängder. CNN-modellen som har använts i den här studien har utgått från den modell som Yoon Kim presenterar i [21].

2.3 Datamängd

En datamängd för övervakad inlärning består av en mängd in- och utdata som diskuteras i sektion 2.2. Processen för hur rätt utdata (positiv/negativ) märks kan utföras på två olika sätt, antingen genom så kallad mänsklig märkning där människor markerar indata som positiv/negativ eller genom "distant supervision" där en dator märker indata som positiv/negativ utefter någon parameter. Den stora skillnaden mellan mänsklig märkning och "distant supervision" är att "distant supervision" kan generera en mycket större datamängd än mänsklig märkning men risken är större att märkningen blir felaktig [2]. Som är beskrivit i sektion **??** är ett delsytfte med denna studie att utvärdera hur de olika märkningsmodellerna påverkar maskininlärningsalgoritmernas precision. Det finns ett urval med publika twitterdatamängder där både mänsklig märkning och "distant supervison" används [2]. Vi har valt att använda datamängderna Stanford Twitter Sentiment (STS) [23] och SemEval [24] eftersom de är de största publika datamängderna inom respektive märkningsmodell.

2.3.1 SemEval

SemEval-datamängden består av 20633 tweets och är framtagen till en årlig TSA-tävling som har gått sedan 2013 [25]. För datamängden har mänsklig märkning tillämpats, där fem personer manuellt märker varje tweet via Amazon Mechanical Turk [26]. I [27] beskrivs metoden för hur märkningen utförs, varje person markerar varje tweet antingen som mycket positivt, positivt, neutralt, negativt eller mycket negativt. Efter att alla tweets är märkta kartläggs alla tweets till kategorierna positivt, neutralt eller negativt utefter tre kriterier. Antingen att alla personer har märkt en tweet samma, en majoritet har märkt samma eller genom att ta ut ett medelvärde. Medelvärdet räknas ut genom att kartlägga de fem kategorierna till heltal mellan 2 och -2, medelvärdet räknas sedan ut och kartläggs till den närmsta kategorin. För SemEval-datamängden blev utfallet 2760 tweets där alla personer gjort samma märkning, 9944 tweets är där majoritet har märkt samma och för resterande 7928 tweets har medelvärdet räknats ut.

2.3.2 Stanford Twitter Sentiment

STS-datamängden är framtagen mellan April 2009 och Juni 2009 av Alec Go, Richa Bhayani och Lei Huang. STS-datamängden är märkt med hjälp av "distant supervision" där märkningen bestäms av vilken typ av emoji [28] ett twitterinlägg innehåller, exempelvis märks ett twitterinlägg positivt om det innehåller ":), :-),:), :D, =)" och negativt om det innehåller ":(, :-(, : (" [23].

2.4 Twittersentimentanalys

TSA är den del av SA som specifikt handlar om att analysera inlägg som användare gör på Twitter. Twitter är en av de populäraste mikrobloggarna där användare kan skriva och kommunicera med varandra genom twitterinlägg. 2013 var Twitter en av de tio mest besökta sidorna på internet och 2016 uppmättes antalet aktiva användare per månad till 319 miljoner. Twitter är definierat som en mikroblogg på grund av det låga antalet tecken som är tillåtet för ett inlägg. I November 2017

fördubblades antalet tillåtna tecken från de tidigare 140 till 280 [29].

Det finns en rad olika begrepp som kännetecknar Twitter och som är viktiga att känna till [2]. En "tweet" är vad som tidigare nämnts ett twitterinlägg. Det är ett inlägg från en användare som är begränsat till 280 tecken, där användaren exempelvis kan delge sina åsikter i olika ämnen eller dela med sig av personliga upplevelser. En "tweet" behöver inte enbart innehålla ren text utan kan även innehålla länkar, bilder och videor. I fortsättningen av rapporten kommer en "tweet" att benämnas twitterinlägg.

När ett twitterinlägg innehåller "mentions" betyder det att andra användare nämns i inlägget. Det kan vara användbart för att exempelvis delge åsikter om andra användare eller för att öppet starta en diskussion med en nämnd användare. För att nämna en användare i ett twitterinlägg skrivs symbolen @ före användarnamnet.

På Twitter har användare möjligheten att följa andra användare. Det betyder att användare kan följa andra användares aktivitet i deras egna twitterflöden och dela med sig av sin egen aktivitet till sina följares twitterflöden. En användare som följer en annan benämns på Twitter som en "follower". Att följa andra är det primära tillvägagångssättet för att skapa kontakter med andra användare på Twitter.

Användare har möjlighet att kategorisera twitterinlägg och det är vad "hashtags" används för. Genom att använda "hashtags" kan användare märka sina twitterinlägg med etiketter för att knyta inlägget till ett specifikt ämne. Användandet av hashtags gör det enkelt för användare att följa ett ämne. De behöver enbart söka på en specifik "hashtag" för att få fram alla twitterinlägg i ämnet. För att skapa en "hashtag" skrivs symbolen # före namnet på etiketten.

Det är även möjligt att dela andra användares twitterinlägg till ens egna följare. Den funktionen kallas för "retweet" och ett sådant twitterinlägg startar vanligtvis med förkortningen RT följt av en "mention" av den ursprungliga författaren av twitterinlägget. Det kan exempelvis vara användbart för att sprida information till följare eller för att skapa en diskussion om innehållet i twitterinlägget med sina egna följare.

När användare svarar på andras twitterinlägg benämns det som "replies" och det är till för att det ska gå att skapa konversationer, där det ska gå att urskilja vanliga

twitterinlägg från svar på twitterinlägg. En användare svarar på ett twitterinlägg genom att göra en referens till den ursprungliga författaren av inlägget följt av svaret på inlägget.

Användare behöver inte göra alla sina inlägg offentliga för alla användare, de kan begränsa synligheten för deras twitterinlägg att enbart synas för deras egna följare.

I figur 2.4 presenteras ett exempel på hur ett twitterinlägg kan se ut. Twitterinlägget är en "retweet som ursprungligen har skrivits av användaren johanselberg. Det innehåller en "hashtag" med etiketten exempel och en "mention" av användaren KAU. Twitterinlägget innehåller även en extern länk.

Figur 2.4: Twitterinlägg innehållande "hashtag", "mention", "retweet".

2.4.1 Utmaningar

På grund av restriktionen av antalet tillåtna tecken i ett twitterinlägg innehåller majoriteten av twitterinlägg enbart en mening. Därför är det skillnad på klassificeringsnivåerna i TSA och SA. I TSA är det ingen skillnad på dokument- och meningsnivå. Därför används det enbart två klassificeringsnivåer inom TSA: meningsnivå (meddelandenivå) och aspektnivå. [2]

Restriktionen av antalet tillåtna tecken utgör den stora skillnaden mellan TSA och SA. Att analysera sentiment på en text i ett twitterinlägg skiljer sig markant från att göra det på vanliga texter som återfinns i produktrecensioner och nyhetsartiklar. Det gör att TSA ställs inför en rad andra utmaningar än vad SA ställs inför.

I [2] tar författarna upp de viktigaste utmaningar med TSA. För att bra resultat ska uppnås med TSA måste dessa utmaningar hanteras. Det som ligger till grund för utmaningarna med TSA är huvudsakligen restriktionen av antalet tillåtna tecken, att det är en informell typ av medium samt att innehållet på Twitter är dynamiskt och ständigt utvecklas.

Det låga antalet tillåtna tecken och att det är en informell typ av medium, gör att språket som används på Twitter skiljer sig från språket som används i vanlig text. Twitterinlägg innehåller ofta felaktigt språkbruk. Det är vanligt förekommande att Twitterinlägg innehåller förkortningar, slang, nybildade ord och att ord betonas genom att de förlängs eller att de skrivs med versaler.

På grund av att användandet av felaktigt språkbruk är så pass vanligt på Twitter, innehåller twitterinlägg en hel del brus. Felstavade termer gör att antalet gånger en specifik term förekommer i en mängd av text blir mindre. Det resulterar i datagleshet (data sparsity) och har en negativ påverkan på resultatet vid SA. För att minska dataglesheten omvandlas vanligtvis felstavade termer till den korrekta stavningen eller en mer korrekt stavning.

En annan utmaning med TSA är att hantera negationer, vilket även gäller SA. Om negationer förekommer i ett twitterinlägg kan det vända på inläggets sentiment. Därför är det viktigt att kunna tolka och identifiera negationer för att sentimentanalysen ska bli korrekt.

I många fall av SA analyseras texter som är skrivna på ett specifikt språk, exempelvis när nyhetsartiklar utgivna av en viss tidning analyseras. Vid TSA är det inte lika enkelt eftersom twitterinlägg kan vara skrivna på flera olika språk och det är inte ovanligt att språk blandas i inlägg. Den här studien kommer enbart analysera twitterinlägg skrivna på engelska och inhämtade inlägg skrivna på annat språk kommer filtreras bort.

Vid TSA filtreras vanligtvis stoppord bort för att öka prestandan. Stoppord är ord som är vanligt förekommande i texter men som saknar någon större betydelse för texten ifråga. I engelskan är “the”, “is” och “who” exempel på stoppord.

Twitterinlägg behöver inte enbart innehålla text utan de kan även innehålla bilder och videor. Bilder och videor kan ge värdefull information om vad för sentiment som uttrycks i ett twitterinlägg. Det kan exempelvis ge information om vem som uttrycker en åsikt eller om vem en åsikt riktas mot. Den här studien kommer inte att ta hänsyn till mediaobjekt utan kommer enbart att analysera text. Främst på grund av att det i dagsläget är ett outforskat område.

Tabell 2.1: Förvirringsmatris

	Förustspådd positiv	**Förutspådd negativ**
Positiv	Sann positiv	Falsk negativ
Negativ	Falsk positv	Sann negativ

2.4.2 Problem

I [2] listar författarna de problem med TSA som de anser bör utforskas ytterligare. Ett av de viktigaste problemen med TSA anser de vara bristen på datamängder som kan användas som riktmärken vid utvärdering av olika klassificeringsmodeller. Forskning som bedrivs i ämnet använder sig av olika datamängder. Dessutom är det vanligt att forskare själva samlar in och skapar egna datamängder som inte publiceras. Olika datamängder kan generera olika resultat. Därför är det svårt att jämföra olika klassificeringstekniker när det används flertalet olika datamängder. I experimentet, som presenteras i den här studien, har problemet adresserats genom att utvärderingen görs mot två kända och publika datamängder. Delvis för att kunna jämföra hur de olika klassificeringsmodellerna presterar mot olika typer av datamängder, men även för att kunna jämföra resultatet mot andra liknande utvärderingar.

2.5 Utvärdering

I tabell 2.1 ser vi en så kallad förvirringsmatris (FM) som utvärderar en klassificeringsmodell från datamängden där “positivt” eller “negativt” är förbestämt. Matrisen visar antalet sann positiv (SP), sann negativ (SN), falsk positiv (FP) och falsk negativ(FN) [30]. Med dessa värden kan vi jämföra och analysera modellerna m.h.a följande utvärderingsmetoder: noggrannhet (n) [31], precision (p), återkallelse (å) [32] och F-Score [33].

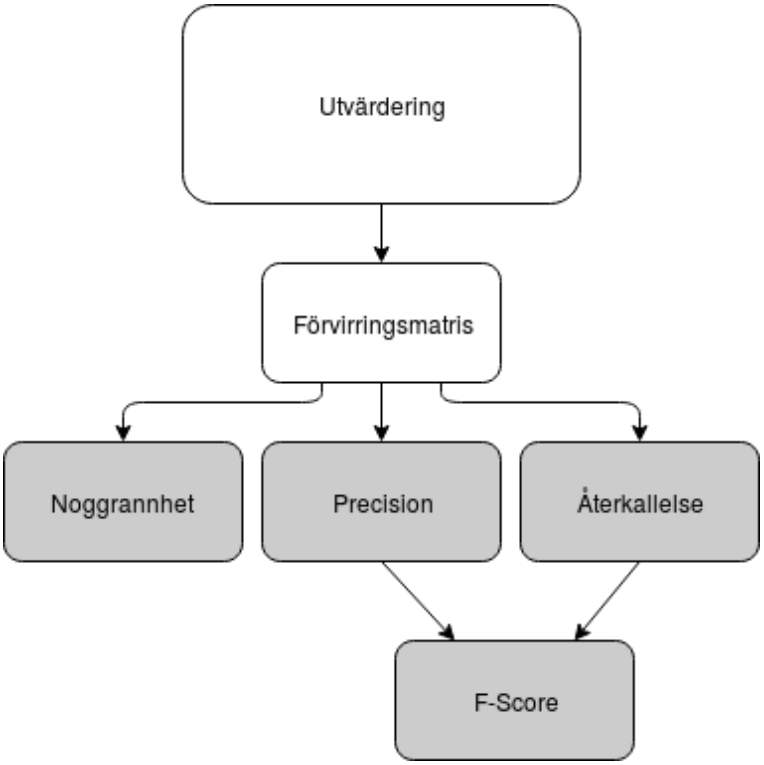
Noggrannhet Är modellens förmåga att kunna märka ett twitterinlägg korrekt som antingen positivt eller negativt. Detta görs genom att ta summan av sannmärkta tweets delat på summan av alla märkningar.

Precision Är förmågan att modellen inte märker ett tweet som positivt när det är negativt. Detta görs genom att ta antalet sann positiva delat på totalt antal positivt märkta tweets.

Återkallelse Är förmågan att modellen märker positiva tweets korrekt. Detta görs genom att ta antalet sann positiva delat på summan av antalet sann positiva och falsk negativa.

F-Score Även kallat det harmoniska medelvärdet mellan precision och återkallelse används då inte alltid precision och återkallelse räcker till för att göra en helhetsbedömning.

I figur 2.5 illustreras hur utvärderingmetoderna hänger ihop.



Figur 2.5: Utvärderingsmetoder för TSA.

2.6 Sammanfattning

I detta kapitel har bakgrunden till projektet diskuterats och där nyttan av SA och TSA har tagits upp. Intressant för uppdragsgivaren är hur de kan integrera SA och TSA i sina produkter t.ex. chatbotar och trendanalys.

Detta kapitel ger även en överblick av vad maskininlärning är och hur maskininlärningens viktigaste komponenter hänger ihop samt hur maskininlärning kan appliceras på SA och TSA. Dessutom beskrivs de problem och utmaningar som existerar inom TSA.

Kapitel 3

Experiment

I kapitlet kommer experimentet för studien att presenteras, hur det har utförts och hur de olika delarna har implementeras. Delarna utgörs av bearbetning av datamängderna, vilka särdragsurval som kommer utföras och en fördjupning i hur maskininlärningsalgoritmerna fungerar. Implementationen kommer ske i tre steg där först en lexikonbaserad modell kommer tas fram, för att ge ett basfall för respektive datamängd, för att sedan implementera algoritmerna med standardparametrar och slutligen testa och justera algoritmernas parametrar. Detta för att eventuellt uppnå en högre precision för respektive klassificeringsmodell. De slutgiltliga klassificeringsmodellerna kommer att diskuteras och jämföras i avsnitt 4.

3.1 Bearbetning av datamängd

En viktig del inom SA är bearbetningen av den data som ska analyseras. Bearbetning av data, i fallet SA, handlar om att tvätta och förbereda texter som ska klassificeras [34].

Kvaliteten på data som ska analyseras är avgörande för vilka resultat som kan uppnås vid maskininlärningen [34]. I [35] rapporteras att bearbetning av text innan maskininlärning kan ha en tydlig positiv påverkan på en klassificeringsmodells prestanda vid sentimentanalays. Därför behöver data som ska analyseras bearbe-

tas innan maskininlärningen, vilket betyder att data normaliseras och reduceras på brus. Teorin med att bearbeta en datamängd innan maskininlärningen är att det delvis kan förbättra prestandan för klassificeringsmodellen men även att klassificering kan utföras snabbare, vilket kan vara av betydelse för sentimentanalys som utförs i realtid [36].

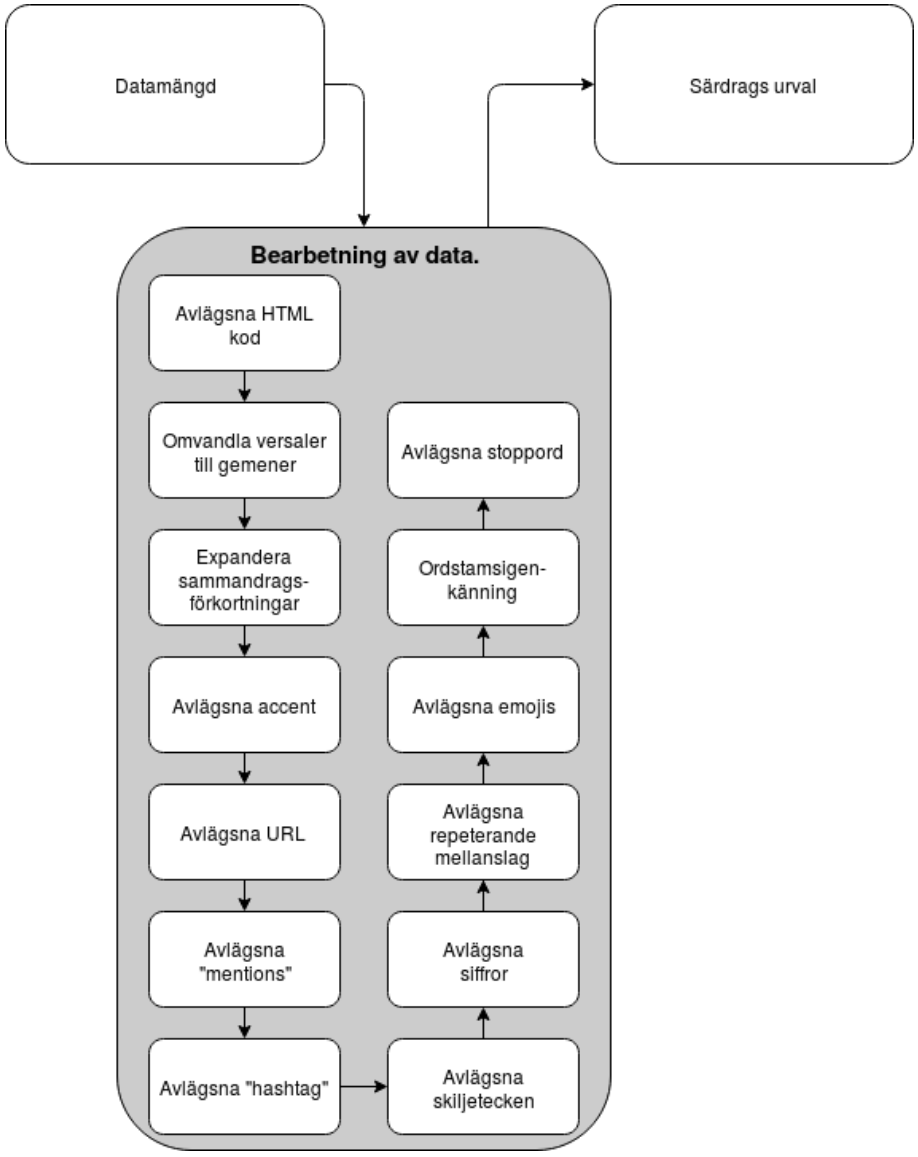
Som nämndes i avsnitt 2.4.1, innehåller twitterinlägg vanligtvis stora mängder brus på grund av det informella språket som används på Twitter, dvs data som inte är användbar för analysprocessen. Det finns exempelvis många ord som inte har någon påverkan på en texts sentiment och därför är det vanligtvis inte nödvändigt att ta med alla ord från den ursprungliga texten vid maskininlärning [34, 35]. Ord som dessa kan tas bort, ersättas eller slås samman med andra. Det är det som kallas för att bearbeta data. Genom att göra det minskas dimensionen på problemet och textklassificeringen blir enklare att utföra , eftersom varje ord behandlas som en dimension [35].

Det existerar flertalet olika tekniker som kan användas vid bearbetningen av data, vissa mer vanliga än andra. I arbetet med studien har två artiklar använts som undersöker effekten av olika bearbetningstekniker vid TSA, [34] och [36]. Båda artiklarna utvärderar bearbetningsteknikerna utefter hur de påverkar prestandan vid sentimentklassificering.

I [34] utvärderas 15 olika bearbetningstekniker. Teknikerna utvärderas var för sig och testas mot två olika datamängder, där de för varje datamängd testas med tre olika klassificeringsmodeller. Några bearbetningstekniker ger bättre noggrannhet för båda datamängderna, andra sämre och resultaten varierar för några. Resultaten varierar inte enbart mellan datamängderna utan de varierar även mellan klassificeringsmodellerna. De tekniker som rekommenderas av författarna och resulterar i hög noggrannhet för alla klassificeringsmodeller och de båda datamängderna är: borttagning av nummer, omvandling av ord till deras ursprungliga form samt ersättning av upprepade skiljetecken. Hantering av negationer, lemmatisering samt ersättning av URL:er och “mentions” av användare påvisar också bra resultat för båda datamängderna men dock inte för alla klassificeringsmodeller. Borttagning av skiljetecken var den bearbetningsteknik som gav sämst resultat. Det som saknas i [34] är hur de olika förbehandlingsteknikerna presterar i kombination med varandra, vilket enligt författarna själva kan ge andra resultat.

Detta är dock något som görs i [36]. Istället för att utvärdera teknikerna isolerade från varandra som i [34], utvärderas de istället utifrån hur de presterar i kombination med andra. Dock har de enbart valt att fokusera på sex olika bearbetningstekniker: expandering av sammandragsförkortningar, expandering av förkortningar, avlägsning av nummer, avlägsning av stoppord, ersättning av förlängda ord och avlägsning av URL:er. Bearbetningsteknikerna har utvärderats utefter hur de presterar på fem olika datamängder på fyra klassificeringsmodeller. Vid bedömningen av prestandan vid sentimentklassificeringen har ett basfall använts. Basfallet har använt samtliga sex bearbetningstekniker för bearbetningen av datan vid testningen. När en specifik teknik har utvärderats, har tekniken i fråga exkluderats från mängden av bearbetningstekniker. Därefter har testerna repeterats, med fem bearbetningstekniker istället för sex. Sedan har förändringen av resultatet gentemot basfallet använts för att bedöma hur tekniken påverkar prestandan vid sentimentklassificeringen. Resultaten visar att expandering av sammandragsförkortningar och expandering av förkortningar kan ha positiv påverkan på noggrannheten vid klassificeringen. Avlägsning av URL:er, nummer och stoppord påverkar resultaten för klassificeringen minimalt, men är effektiva för att minska brus.

Utifrån resultaten som har presenterats i [34] och [36] har bearbetningsteknikerna för experimentet i den här studien valts. Det svåra med att välja bearbetningstekniker, utifrån litteraturen, är att det kan existera flera olika varianter av en viss teknik och att olika namn kan användas för att beskriva en viss teknik. I [34] hanterar de exempelvis URL:er genom att ersätta varje URL med etiketten “URL”, medan de avlägsnas helt i [36]. Eftersom bearbetningsteknikerna uteslutande kommer att användas i kombination med varandra vid det här experimentet har störst vikt lagts på de tekniker som presenteras i [36].



Figur 3.1: Bearbetningstekniker som används för att bearbeta data vid experimentet.

Nedan presenteras och beskrivs de bearbetningstekniker som har valts att användas för experimentet. De valda teknikerna visualiseras i figur 3.1. För några av bearbetningsteknikerna har ordningen betydelse. Exempelvis kommer inte expanderingen av sammandragsförkortningar kunna genomföras ifall avlägsnandet av accenter sker innan. Totalt har 13 bearbetningstekniker använts. Majoriteten

av dessa handlar om att avlägsna text från twitterinläggen. Det som avlägsnas vid bearbetningen av twitterinläggen är: HTML-kod, accenter, URL:er, “mentions”, “hashtags”, skiljetecken, siffror, repeterande mellanslag, emojis och stoppord. Avlägsning av emojis diskuteras i avsnitt 3.2. Stoppord, som beskrevs i 2.4.1, tas bort genom att twitterinläggen söks igenom på stoppord från en fördefinierad lista. Trots att man i [34] rapporterar sämre prestanda vid avlägsning av skiljetecken har det valts att användas som en bearbetningsteknik för det här experimentet. Det valet gjordes eftersom att man i [34] samtidigt rapporterar att det bidrar till att reducera storleken på klassificeringsproblem.

De övriga teknikerna som används omvandlar text under bearbetningen av ett twitterinlägg. Ett av de första stegen i den bearbetningsprocess, som har använts för detta experiment, är att omvandla versaler i ett twitterinlägg till gemener. Det gör att ord som är skrivna på flera olika sätt slås samman till ett [34]. Vokabulär blir då mindre, vilket gör att storleken på problemet minskas.

En annan teknik som används är att expandera sammandragsförkortningar. I 2.4.1 diskuterades vikten av att kunna tolka och identifiera negationer för att SA ska bli korrekt. Eftersom sammandragsförkortningar är vanligt förekommande i engelskan fokuserar tekniken på att expandera sammandragsförkortningar med motsägelser. Exempelvis är termen “don’t” exempel på en sammandragsförkortning för motsägelsen “do not”. För experimentet har sammandragsförkortningarna “n’t”, “can’t” och “won’t” omvandlats till “not”, “can not” och “will not”, utefter hur metoden beskrevs i [36].

Slutligen har också ordstamsigenkänning använts vid bearbetningen av ett twitterinlägg. Ordstamsigenkänning går ut på att ta bort ändelser från ord för att omvandla dem till deras ursprungliga form [34]. Det leder också till att ord slås samman och att storleken på problemet minskar. Tabell 3.1 visar ett exempel på vad som sker då ordstamsigenkänning används på olika böjningar av det engelska ordet ”arrive”.

Tabell 3.1: Exempel på vad som sker när ordstamsigenkänning används på olika böjningar av det engelska ordet ”arrive”.

Före	**Efter**
Arriving	Arriv
Arrive	Arriv
Arrives	Arriv

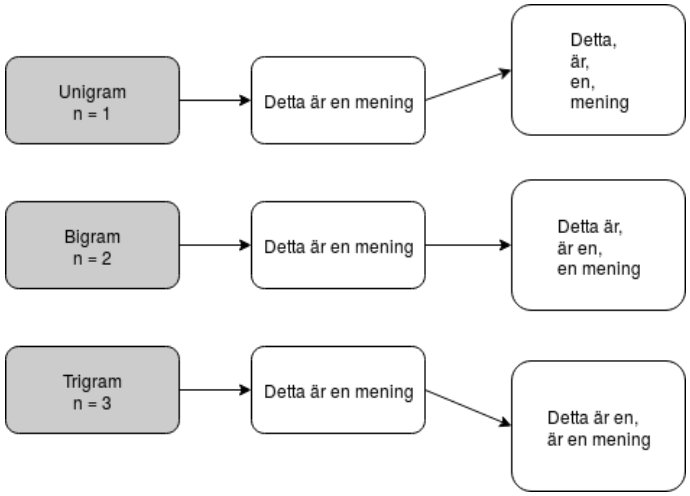
Tabell 3.2 visar ett exempel på vad som sker när ett twitterinlägg bearbetas med bearbetningsteknikerna som är valda för experimentet.

Tabell 3.2: Exempel på vad som sker med ett twitterinlägg efter det har bearbetas med alla valda bearbetningstekniker.

Före	@SportsCenter Hands down the Irish amateur, Paul Dunne, thrilled the world. More to come tomorrow!!! #NICE https://t.co/kTYNkltfl6
Efter	hand irish amateur paul dunn thrill world come tomorrow nice

3.2 Särdragsurval

Inom maskininlärning definieras ett särdrag som en individuell mätbar egenskap eller kännetecken på ett fenomen som har observerats, begreppet särdrag kommer från variabler som används inom statistik. Att välja korrekta, informativa och oberoende särdrag är ett kritiskt moment för att maskininlärningsalgoritmen ska kunna uppnå sin fulla potential och undvika fel klassificeringar [37]. I [2] presenteras det fyra olika särdragsklasser inom TSA: semantiska, syntaktiska, stilistiska och twitter-specifika särdrag.



Figur 3.2: Illustration av n-grams på en mening.

Semantiska särdrag används mestadels för att ta ut åsiktsord, sentimentord och negationer. Åsiktsord är ord eller meningar som kan innehålla någon typ av åsikt medan sentimentord är ord som innehåller något positivt eller negativt. Negationer är ett viktigt koncept eftersom en mening som innehåller någon typ av negation kan skifta om twitterinlägget är positivt eller negativt. Studien kommer använda sentimentord i lexikonimplementationen som diskuteras i avsnitt 3.4.2 och negationer som även diskuterats i avsnitt 3.1 kommer implementeras i avsnitt 3.4.1.

Syntaktiska särdrag används för att utforska påverkan av olika termer i SA och TSA. Syntaktiska särdrag bryts ner till unigram, bigram, trigram, n-gram, termfrekvens och "Part of Speech" (POS), där n-gram är ett samlingsord för uni-, bi- och trigram. I figur 3.2 illustreras hur n-grammodellen fungerar på en enkel mening och hur meningens ord grupperas på olika sätta till ett ord eller fras. **ta upp något om varför det används?**

Termfrekvens kan används i kombination med n-gram. Längre twitterinlägg kan ha ett högre antal förekomster av ett ord än ett kortare twitterinlägg, vilket kan leda till att ord kan få större betydelse för klassificeringen. Genom användandet av termfrekvens delas antalet förekomster av ett ord i ett twitterinlägg med alla ord som finns i twitterinlägget undviks sådana problem [38].

POS utförs genom att man räknar hur många substantiv, verb och adjektiv som

existerar i ett twitterinlägg. POS kan t.ex användas för att ta reda på åsikter i ett twitterinlägg där antalet adjektiv kan relateras till vilken åsikt twitterinlägget har.

Studien kommer implementera n-gram och termfrekvens för varje klassificeringsalgoritm i avsnitt 3.4. POS kommer inte implementeras i studien då [39, 40] rapporterar försämrad precision. Däremot rapporteras i [41] små förbättringar vid användning av POS.

Stilistiska särdrag är särdrag som kommer från det informella skrivsättet som används på Twitter där t.ex emojis, förkortningar, slang och skiljetecken används. Enligt [41] kan emojis ha en stor betydelse för TSA. Men eftersom STS-datamängden har avlägsnat emojis, då "distant supervision" [23] har använts, kommer inte studiens modeller ta hänsyn till emojis. Därför har även emojis från SemEval-datamängden filtrerats bort i avsnitt 3.1. Studien kommer inte hantera slang eftersom inget större slanglexikon var öppet för användning. Förkortningar och skiljetecken kommer hanteras i denna studie och har diskuterats i avsnitt 3.1 och implementeras i avsnitt 3.4.1.

Twitter-specifika särdrag är särdrag som är specifika för Twitters domän som t.ex "hashtags" och "mentions" vilka diskuterats i 2.4 och 3.1.

3.3 Algoritmer

För att ge en bättre förståelse för hur NB, SVM och CNN kan appliceras på sentimentanalys, kommer detta avsnitt ge en överblick av hur algoritmerna går till väga för att klassificera text och hur de kan anpassas för att uppnå bra resultat vid sentimentanalys.

3.3.1 Naive Bayes

I 2.2.1 beskrivs att Naive Bayes-algoritmen är baserad på Bayes theorem [14] vilket för TSA betyder att räkna ut sannolikheten att ett twitterinlägg (B) är positivt eller negativt (A) för en mängd C, bestående av ett antal B med ett förbestämt A. Där $P(A|B)$ är en villkorlig sannolikhet för att A inträffar givet att B är

Tabell 3.3: Exempeldatamängd med twitterinlägg som är märkta som positiva eller negativa.

Twitterinlägg	**Märkning**
ser fram emot en dag i solen	Positiv
jävla skitväder	Negativ
vilken jävla kung du är linkan	Positv
fyfan vilken dålig match	Negativ

sant, $P(B|A)$ är också en villkorlig sannolikhet för att B inträffar givet att A är sant. $P(A)$ och $P(B)$ är sannolikheten att observera $P(A)$ och $P(B)$ oberoende av varandra,

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}. \tag{3.1}$$

För att ekvation 3.1 skall vara "Naive" antas att varje ord b_w i B är oberoende av varandra vilket betyder att vi inte längre klassificerar B utan b_w, $P(B|A) = P(b_{w1}|A) * P(b_{w2}|A) * \ldots * P(b_{wn}|A)$, där n är antalet ord i B,

För att beräkna $P(b_w|A)$ används Dirichlet prior [42, 43],

$$P(b_w|A) = \frac{N_{b_w} + \alpha}{N_A + \alpha_n}. \tag{3.2}$$

Där N_{b_w} är hur många gånger b_w förekommer i delmängden C_A, N_A är summan av alla ord i C_A, α och α_n används för att utföra additiv utjämning [44] där $\alpha = 1$ och α_n är summan av antal olika ord i C.

I tabell 3.3 illustreras en datamängd C med fyra twitterinlägg och deras märkning. Denna datamängd kan användas för att beräkna om twitterinlägget "solen skiner alltid i karlstad" (B) är positivt eller negativt (A) . Genom att räkna ut sannolikheten $P(Positiv|B)$ och $P(Negativ|B)$ kan märkning bestämmas, detta görs genom att jämföra vilken sannolikhet som blir störst och vi får följande ekvation,

$$P(A|B) = \prod_{i \epsilon B} \frac{P(i|A)}{P(B)}, \tag{3.3}$$

Och eftersom nämnaren är lika i bägge uträkningarna kommer enbart täljaren vara relevant vid beräkning.

Tabell 3.4: Beräkningar för om ett ord är positivt eller negativt.

Ord	**P(ord|Positiv)**	**P(ord|Negativ)**
solen	$\frac{1+1}{13+17}$	$\frac{0+1}{6+17}$
skiner	$\frac{0+1}{13+17}$	$\frac{0+1}{6+17}$
alltid	$\frac{0+1}{13+17}$	$\frac{0+1}{6+17}$
i	$\frac{1+1}{13+17}$	$\frac{0+1}{6+17}$
karlstad	$\frac{0+1}{13+17}$	$\frac{0+1}{6+17}$

För att räkna ut exempelvis $P(solen|Positiv)$ används ekvation 3.2 där $N_{b_w} = 1$, $N_A = 13$, $\alpha_n = 17$ och $\alpha = 1$ vilket ger oss $P(solen|Positiv) = \frac{1+1}{13+17} = 0,0667$. I tabell 3.4 illustreras alla ekvationer för att beräkna om "solen skiner alltid i karlstad" är positiv eller negativ. Ovanstående utförs på varje ord för både positiv och negativ vilket illustreras i tabell 3.4. Med hjälp av tabell 3.4 och ekvation 3.3 beräknas,

$$P(Positiv|\text{solen skiner alltid i karlstad}) = \frac{1.6428 * 10^{-}7}{P(B)}, \tag{3.4}$$

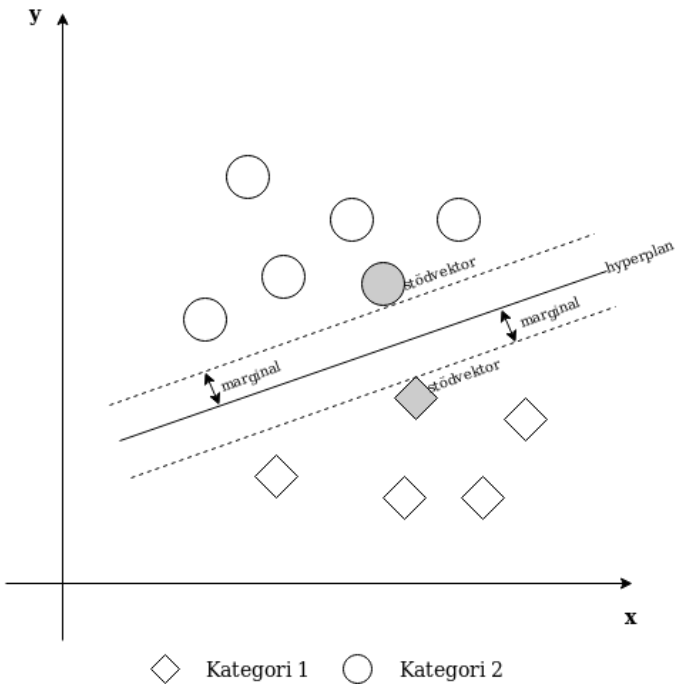
och

$$P(Negativ|\text{solen skiner alltid i karlstad}) = \frac{1.5576 * 10^{-}7}{P(B)}. \tag{3.5}$$

Jämförs dessa resultat kommer twitterinlägget klassificeras som positivt.

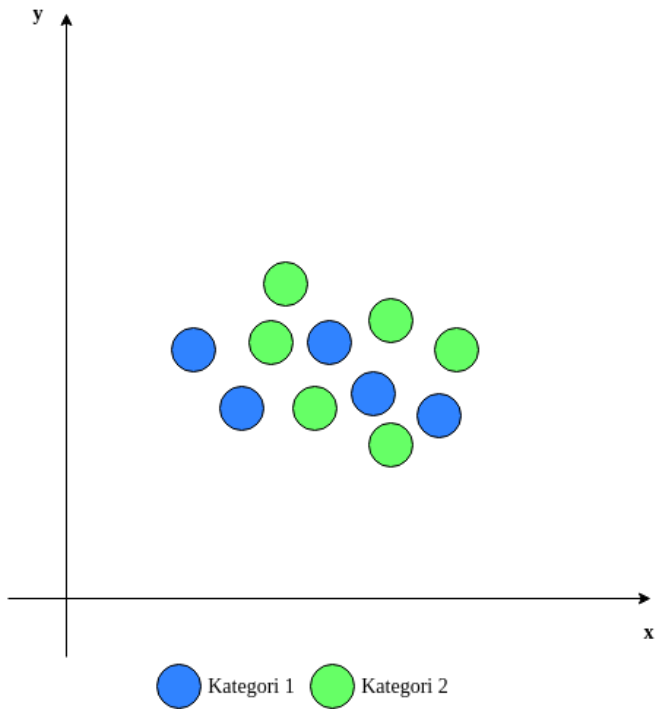
3.3.2 Support Vector Machine

I avsnitt 2.2.2 beskrevs SVM kortfattat, i detta avsnitt kommer SVM att beskrivas mer ingående. SVM-algoritmen skapar en modell utifrån en given mängd märkt träningsdata [18]. Modellen som skapas har sedan förmågan att kategorisera ny indata i en av de två kategorierna. Den skapade SVM-modellen representerar träningsdata som punkter i ett koordinatsystem. I koordinatsystemet kartläggs data så att de två kategorierna är tydligt separerade med ett så stort mellanrum som möjligt. När modellen kategoriserar ny indata kartläggs den datan till samma koordinatsystem, för att sedan kategoriseras utefter vilken sida av mellanrummet den hamnar på.



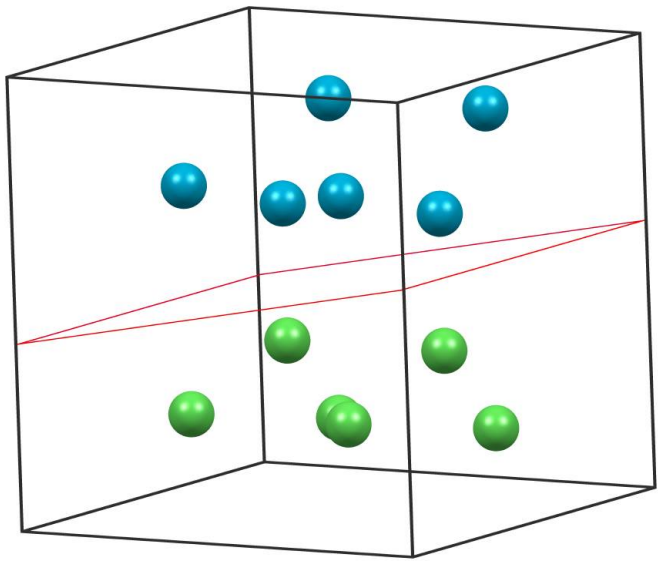
Figur 3.3: Exempel på en linjär klassificerare där de två kateogrierna är linjärt separabla.

För att separera kategorierna konstruerar SVM ett hyperplan [17]. I figur 3.3 representeras hyperplanet av linjen mellan kategorierna. Datapunkterna som ligger närmast hyperplanet för respektive kategori kallas för stödvektorer och det är utefter de punkterna som hyperplanet positioneras. Det kan finnas flera möjliga hyperplan som kan kategorisera datan. Idén bakom SVM är att hitta hyperplanet som bäst delar upp datamängden i två kategorier. Ju större marginal mellan kategorierna desto mindre känslig blir modellen för generaliseringsfel [18]. Därför anses hyperplanet som har störst marginal till de närmaste stödvektorerna från respektive kategori ge en bra separation av data.



Figur 3.4: Exempel på en linjär klassificerare där de två kategorierna inte är linjärt separabla.

Exemplet i figur 3.3 representerar en linjär klassificerare, då de två kategorierna är linjärt separabla. Dock finns det klassificeringsproblem som inte går att separera linjärt, så kallade icke-linjära klassificeringsproblem. Figur 3.4 visar ett exempel på ett sådant problem. SVM kan även hantera sådana klassificeringsproblem. I fall som dessa ritar SVM om datapunkterna till en högre dimension för att en separation ska bli möjlig [17], vilket illustreras i figur 3.5. Processen upprepas tills dess att ett hyperplan kan separera datan. Metoden SVM använder för att utföra detta effektivt kallas “kernel trick” [45].



Figur 3.5: Exempel på icke-linjär klassificerare, där hyperplanet har dragits i en högre dimension.

Om ett problem kräver en linjär eller en icke-linjär klassificerare beror på hur den data som ska klassificeras ser ut. Beteendet för SVM-algoritmen går att anpassa utefter den data som ska klassificeras genom att välja olika "kernel"-funktioner [46]. För textklassificering är det rekommenderat att använda en linjär "kernel", vilket ger en linjär klassificerare. Anledningen till det är dels att de flesta textklassificeringsproblem är linjärt separerbara. Men det är även bra att använda en linjär "kernel" när antalet särdrag för klassificeringsproblemet är högt, vilket är fallet vid textklassificering. När antalet särdrag är högt är det ingen märkbar skillnad i prestanda mellan en linjär kontra icke-linjär "kernel". Dessutom går en SVM-modell med en linjär "kernel" snabbare att träna än andra och det är färre parametrar som behöver justeras vid optimering av en modell. På grund av ovanstående anledningar har en linjär klassificerare valts att användas vid den här studien.

3.3.3 Convolutional Neural Network

3.4 Implementation

Detta kapitel beskriver detaljer angående implementationen av vår studie och dess flöde som illustrerats i figur 2.1. Kapitlet beskriver även teknikerna som används för att göra implementationen möjlig. Gemensamma tekniker som används i alla delar av implementation är programmeringsspråket Python3 [47] och Pandas [48]. Pandas är ett Python-paket som erbjuder snabba, flexibla datastrukturer som är utformade för att göra arbetet med relationell och märkt data smidigare.

3.4.1 Databearbetning

Som beskrivs i avsnitt 3.1 är databearbetning ett viktigt steg inom SA och TSA. I detta avsnittet beskrivs implementationen av databearbetningen m.h.a delarna som illustreras i figur 3.1.

För att avlägsna HTML-kod från twitterinlägg använder vi oss av BeautifulSoup [49], som är ett Python-paket för att extrahera data från HTML-kod,

Kodavsnitt 3.1: Test

```
def remove_html_encode(self):
        self.df["tweet"] = self.df["tweet"]./
        apply(lambda x: BeautifulSoup(x, 'lxml').get_text())
```

Om HTML-kod finns i twitterinlägg tas det således bort.

Omvandling från versaler till gemener utförs på varje twitterinlägg m.h.a den inbyggda Python-funktionen `str.lower()` som konverterar versaler till gemener.

Kodavsnitt 3.2: Test

```
def to_lower(self):
      self.df["tweet"] = self.df["tweet"].str.lower()
```

*** Skriv om expand cont *****

Kodavsnitt 3.3: Test

```
cont_dict = {'can\'t': 'cannot', 'won\'t': 'will not', 'n\'t': ' not'}
```

```
cont_re = re.compile('(%s)' % '|'.join(cont_dict.keys()))

def expand_cont(self, s, cont_dict=cont_dict):
       def replace(match):
              return cont_dict[match.group(0)]
        return cont_re.sub(replace, s)

def expand_contractions(self):
       self.df["tweet"] = self.df["tweet"].apply(lambda x: self.expand_cont(x))
```

Avlägsna accenter utförs genom att använda `unicodedata.normalize()` [50] som använder "Normalization Form Compatibility Decomposition" (NFKD) parametern, vilket betyder att `unicodedata.normalize()` byter ut en bokstav med accent mot sin ASCII-ekvivalenta bokstav. Resterande parametrar säger att om bokstaven redan är ascii eller utf-8 ignoreras dessa bokstäver.

Kodavsnitt 3.4: Test

```
def remove_accented_chars(self):
  self.df["tweet"] = self.df["tweet"].apply(lambda x:unicodedata./
  normalize('NFKD', x).encode('ascii', 'ignore')./
  decode('utf-8', 'ignore'))
```

För att avlägsna URL, "mentions", "hashtags" , skiljetecken, siffror, emojis och repeterade mellanslag används fyra reguljära uttryck (RU) [51]. För att avlägsna URL används `'https?://[A-Za-z0-9./]+'` vilket betyder att om en URL är http eller https kommer denna avlägsnas, samt om den efterföljs av ännu en URL kommer även den tas bort.

"Mentions" avlägsnas med det RU `'@[A-Za-z0-9]+'`, d.v.s om en alfanumerisk sträng börjar med '@' avlägsnas den alfanumeriska strängen.

Avlägsning av "hashtags", skiljetecken, siffror och emojis görs med det RU `'[^a-zA-Z]'`, d.v.s en sträng måste börja på bokstäver mellan a-z, om inte tas dessa bort.

Och slutligen så avlägsnas repeterande mellanslag, vilket utförs med det RU `' +'`. Ifall det är två eller fler repeterande mellanslag tas dessa bort.

Kodavsnitt 3.5: Test

```
def remove_links(self):
       self.df.loc[:, "tweet"] = self.df.loc[:, "tweet"].\
```

```
        replace(r'https?://[A-Za-z0-9./]+', '', regex=True)

def remove_twitter_mention(self):
        self.df.loc[:, "tweet"] = self.df.loc[:, "tweet"].\
        replace(r'@[A-Za-z0-9]+', '', regex=True)

def remove_hashtag(self):
        self.df.loc[:, "tweet"] = self.df.loc[:, "tweet"].\
        replace('[^a-zA-Z]', ' ', regex=True)

def remove_extra_whitepsace(self):
       self.df.loc[:, "tweet"] = self.df.loc[:, "tweet"].\
        replace(' +', ' ', regex=True)
```

För att utföra ordstamsigenkänning och avlägsning av stoppord måste varje twitterinlägg först tokeniseras, vilket betyder att varje ord i ett twitterinlägg blir ett eget element i en lista. Tokeniseringen utförs m.h.a funktionen `word_tokenize()` [52] från Natural Language Toolkit (NLTK).

Kodavsnitt 3.6: Test

```
def tokenize(self):
       self.df["tweet"] = self.df["tweet"].apply(word_tokenize)
```

Ordstamsigenkänning utförs m.h.a `SnowballStemmer('english').stem()` [53] från NLTK som försöker hitta varje ords ordstam vilket diskuteras i avsnitt 3.1.

Kodavsnitt 3.7: Test

```
def word_stemming(self):
        self.df["tweet"] = self.df["tweet"]./
        apply(lambda x: [SnowballStemmer('english').stem(y) for y in x])
```

Avlägsning av stoppord utförs m.h.a `stopwords.word('english')` [54] från NLTK som är en lista av engelska stoppord. `remove_stopwords(self)` kontrollerar om något ord i twitterinlägget finns med i listan, om det gör det tas det bort.

Kodavsnitt 3.8: Test

```
stop = stopwords.words('english')
def remove_stopwords(self):
        self.df["tweet"] = self.df["tweet"]./
```

```
    apply(lambda x: [word for word in x if word not in stop])
```

3.4.2 Lexikon

3.4.3 Naive Bayes

3.4.4 Support Vector Machine

3.4.5 Prediction program.

3.4.6 * ev GUI implementation om tid finns *

3.5 Sammanfattning

Kapitel 4

Resultat

4.1 Intro

4.2 Resultatet mellan modellerna

4.2.1 Dataset 1 -> jämför resultat mellan modellerna

4.2.2 Dataset 2 -> jämför resultat mellan modellerna

4.2.3 Dataset 3 -> jämför resultat mellan modellerna

4.3 Implementations mässigt vilken modell är lättast?

4.4 implementations jämförelse (resultat VS förväntat)

4.5 Summering

Kapitel 5

Slutsats

5.1 Sammanfattning

5.2 Problem

5.3 Begränsningar

5.4 Vidare utveckling

5.5 Slutord

Litteraturförteckning

[1] Hoda Korashy Walaa Medhat, Ahmed Hassan. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, Volume 5(Issue 4):1093–1113, December 2014. URL `https://ac.els-cdn.com/S2090447914000550/1-s2.0-S2090447914000550-main.pdf?_tid=77d36d1a-ff80-11e7-956d-00000aacb362&acdnat=1516631469_c14e5bc49162e2b9a4232c2931592298`.

[2] Anastasia Giachanou and Fabio Crestani. Like it or not: A survey of twitter sentiment analysis methods. *ACM Comput. Surv.*, 49(2):28:1–28:41, June 2016. ISSN 0360-0300. doi: 10.1145/2938640. URL `http://doi.acm.org/10.1145/2938640`.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[4] Wikipedia contributors. Atari games — Wikipedia, the free encyclopedia, 2016. URL `https://en.wikipedia.org/wiki/Atari_Games`. [Online; accessed 5-Februari 2018].

[5] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[6] Wikipedia contributors. Lee sedol — Wikipedia, the free encyclopedia,

2016. URL https://en.wikipedia.org/wiki/Lee_Sedol. [Online; accessed 5-Februari 2018].

[7] Wikipedia contributors. Go(game) — Wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/wiki/Go_(game). [Online; accessed 5-Februari 2018].

[8] Apple. Ios - siri - apple, n.d. URL https://www.apple.com/ios/siri/. [Online; accessed 5-March 2018].

[9] Google. Google assistant - your own personal google, n.d. URL https://assistant.google.com/. [Online; accessed 5-March 2018].

[10] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[11] Johan Ugander and Lars Backstrom. Balanced label propagation for partitioning massive graphs. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 507–516. ACM, 2013.

[12] Wikipedia contributors. Unsupervised learning — Wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/wiki/Unsupervised_learning. [Online; accessed 5-Februari 2018].

[13] Jason Brownlee. Supervised and unsupervised machine learning algorithms, 2016. URL https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/.

[14] Wikipedia contributors. Bayes' theorem — Wikipedia, the free encyclopedia, 2002. URL https://en.wikipedia.org/wiki/Bayes%27_theorem. [Online; accessed 12-February 2018].

[15] Sunil Ray. 6 easy steps to learn naive bayes algorithm (with codes in python and r), 2015. URL https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/. [Online; accessed 12-February 2018].

[16] Eric W. Weisstein. "hyperplane.from mathworld–a wolfram web resource, n.d. URL http://mathworld.wolfram.com/Hyperplane.html. [Online; accessed 12-February 2018].

[17] Noel Bambrick. Support vector machines: A simple explanation, n.d. URL https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html. [Online; accessed 12-February 2018].

[18] Wikipedia contributors. Support vector machine — Wikipedia, the free encyclopedia, 2002. URL https://en.wikipedia.org/wiki/Support_vector_machine. [Online; accessed 12-February 2018].

[19] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, Aug 2017. doi: 10.1109/ICEngTechnol.2017.8308186.

[20] Wikipedia contributors. Artificial neural network — wikipedia, the free encyclopedia, 2018. URL https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=830851868. [Online; accessed 28-March-2018].

[21] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[22] Shiyang Liao, Junbo Wang, Ruiyun Yu, Koichi Sato, and Zixue Cheng. Cnn for situations understanding based on sentiment analysis of twitter data. *Procedia Computer Science*, 111:376 – 381, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.06.037. URL http://www.sciencedirect.com/science/article/pii/S1877050917312103. The 8th International Conference on Advances in Information Technology.

[23] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.

[24] International workshop on semantic evaluation 2017, 2016. URL http://alt.qcri.org/semeval2017/. [Online; accessed 12-February 2018].

[25] Sentiment analysis in twitter, 2016. URL http://alt.qcri.org/semeval2017/task4/. [Online; accessed 12-February 2018].

[26] Amazon. Amazon mechanical turk, n.d. URL https://www.mturk.com/product-details. [Online; accessed 12-February 2018].

[27] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18, 2016.

[28] Wikipedia contributors. Emoji — Wikipedia, the free encyclopedia, 2014. URL https://sv.wikipedia.org/wiki/Emoji. [Online; accessed 5-Februari 2018].

[29] Wikipedia contributors. Twitter — Wikipedia, the free encyclopedia, 2007. URL https://en.wikipedia.org/wiki/Twitter. [Online; accessed 5-Februari 2018].

[30] Wikipedia contributors. Confusion matrix — Wikipedia, the free encyclopedia, 2014. URL https://en.wikipedia.org/wiki/Confusion_matrix. [Online; accessed 29-January 2018].

[31] Wikipedia contributors. Accuracy and precision — Wikipedia, the free encyclopedia, 2002. URL https://en.wikipedia.org/wiki/Accuracy_and_precision. [Online; accessed 5-Februari 2018].

[32] Wikipedia contributors. Precision and recall — Wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/wiki/Precision_and_recall. [Online; accessed 29-January 2018].

[33] Wikipedia contributors. F1 score — Wikipedia, the free encyclopedia, 2014. URL https://en.wikipedia.org/wiki/F1_score. [Online; accessed 29-January 2018].

[34] Dimitrios Effrosynidis, Symeon Symeonidis, and Avi Arampatzis. A comparison of pre-processing techniques for twitter sentiment analysis. In *International Conference on Theory and Practice of Digital Libraries*, pages 394–406. Springer, 2017.

[35] Emma Haddi, Xiaohui Liu, and Yong Shi. The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17:26 – 32, 2013. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2013.05.005. URL `http://www.sciencedirect.com/science/article/pii/S1877050913001385`. First International Conference on Information Technology and Quantitative Management.

[36] Z. Jianqiang and G. Xiaolin. Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, 5:2870–2879, 2017. ISSN 2169-3536. doi: 10.1109/ACCESS.2017.2672677.

[37] Wikipedia contributors. Feature (machine learning) — Wikipedia, the free encyclopedia, 2004. URL `https://en.wikipedia.org/wiki/Feature_(machine_learning)`. [Online; accessed 12-Mars 2018].

[38] scikitlearn documentation. Working with text data, n.d. URL `http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html`. [Online; accessed 12-Mars 2018].

[39] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.

[40] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11(538-541):164, 2011.

[41] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics, 2011.

[42] Wikipedia contributors. Dirichlet distribution — Wikipedia, the free encyclopedia, 2004. URL `https://en.wikipedia.org/wiki/Dirichlet_distribution`. [Online; accessed 26-Februari 2018].

[43] David Heckerman. A tutorial on learning with bayesian networks. In *Learning in graphical models*, pages 301–354. Springer, 1998.

[44] Wikipedia contributors. Additive smoothing — Wikipedia, the free encyclopedia, 2008. URL https://en.wikipedia.org/wiki/Additive_smoothing#Pseudocount. [Online; accessed 26-Februari 2018].

[45] Wikipedia contributors. Kernel method — Wikipedia, the free encyclopedia, 2005. URL https://en.wikipedia.org/wiki/Kernel_method. [Online; accessed 19-Mars 2018].

[46] Alexandre Kowalczyk. Linear kernel: Why is it recommended for text classification ?, 2014. URL https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/. [Online; accessed 27-Mars 2018].

[47] Python 3.0 release, n.d. URL https://www.python.org/download/releases/3.0/. [Online; accessed 4-April 2018].

[48] pandas: powerful python data analysis toolkit¶, 2017. URL https://pandas.pydata.org/pandas-docs/stable/. [Online; accessed 4-April 2018].

[49] Leonard Richardson. Beautiful soup documentation, 2004. URL https://www.crummy.com/software/BeautifulSoup/bs4/doc/#. [Online; accessed 4-April 2018].

[50] 6.5. unicodedata — unicode database, n.d. URL https://www.crummy.com/software/BeautifulSoup/bs4/doc/#. [Online; accessed 4-April 2018].

[51] Jeffrey Friedl. 6.2. re — regular expression operations, 2009. URL https://docs.python.org/3/library/re.html. [Online; accessed 4-April 2018].

[52] nltk.tokenize package - nltk 3.2.5 documentation, n.d. URL http://www.nltk.org/api/nltk.tokenize.html. [Online; accessed 4-April 2018].

[53] nltk.stem package - nltk 3.2.5 documentation, n.d. URL http://www.nltk.org/api/nltk.tokenize.html. [Online; accessed 4-April 2018].

[54] Removing stop words with nltk in python - geeksforgeels, n.d. URL http://www.nltk.org/api/nltk.tokenize.html. [Online; accessed 4-April 2018].