

# 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, 5 mars 2018

Johan Selberg och Johannes Bandgren



# Innehåll

<b>1</b>	<b>Introduktion</b>	<b>1</b>
<b>2</b>	<b>Bakgrund</b>	<b>2</b>
2.1	Introduktion . . . . .	2
2.2	Sentimentanalys . . . . .	3
2.2.1	Klassificeringstekniker . . . . .	4
2.3	Maskininlärning . . . . .	5
2.3.1	Naive Bayes . . . . .	7
2.3.2	Support Vector Machine . . . . .	7
2.3.3	XXX . . . . .	8
2.4	Datamängd . . . . .	8
2.4.1	SemEval . . . . .	8
2.4.2	Stanford Twitter Sentiment . . . . .	9
2.5	Twittersentimentanalys . . . . .	9
2.5.1	Utmaningar . . . . .	11
2.5.2	Problem . . . . .	12
2.6	Utvärdering . . . . .	13
2.7	Sammanfattning . . . . .	14
<b>3</b>	<b>Experiment</b>	<b>15</b>
3.1	Intro . . . . .	15
3.2	Bearbetning av datamängd . . . . .	15
3.3	Särdragsurval . . . . .	18
3.4	Algoritmer . . . . .	18

3.4.1	Naive Bayes . . . . .	18
3.4.2	Support Vector Machine . . . . .	20
3.5	Implementation . . . . .	21
3.5.1	Naive Bayes . . . . .	21
3.5.2	Support Vector Machine . . . . .	21
3.5.3	Prediction program. . . . .	21
3.5.4	* ev GUI implementation om tid finns * . . . . .	21
3.6	Sammanfattning . . . . .	21
<b>4</b>	<b>Resultat</b>	<b>22</b>
4.1	Intro . . . . .	22
4.2	Resultatet mellan modellerna . . . . .	22
4.2.1	Dataset 1 -> jämför resultat mellan modellerna . . . . .	22
4.2.2	Dataset 2 -> jämför resultat mellan modellerna . . . . .	22
4.2.3	Dataset 3 -> jämför resultat mellan modellerna . . . . .	22
4.3	Implementations mässigt vilken modell är lättast? . . . . .	23
4.4	implementations jämförelse (resultat VS förväntat) . . . . .	23
4.5	Summering . . . . .	23
<b>5</b>	<b>Slutsats</b>	<b>24</b>
5.1	Sammanfattning . . . . .	24
5.2	Problem . . . . .	24
5.3	Begränsningar . . . . .	24
5.4	Vidare utveckling . . . . .	25
5.5	Slutord . . . . .	25



# Figurer

2.1	Utvärderingsprocessen av maskininlärningsmodeller . . . . .	3
2.2	Inlärningsprocesser för maskininläring. . . . .	6
2.3	Klassificeringsprocess av data. . . . .	7
2.4	Twitterinlägg innehållande "hashtag", mention", retweet". . . . .	11
2.5	Utvärderingsmetoder för TSA. . . . .	14
3.1	Bearbetningsprocessens olika steg. . . . .	16

# Tabeller

2.1	Förringsmatris . . . . .	13
-----	--------------------------	----

# Kapitel 1

## Introduktion

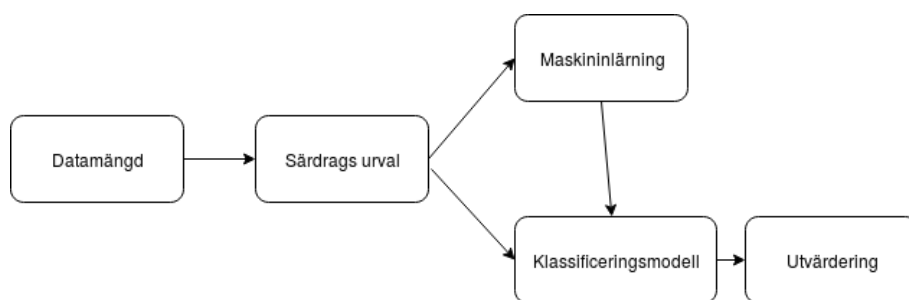
# Kapitel 2

## Bakgrund

### 2.1 Introduktion

Syftet med studien är att utvärdera (x) 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 klassificeringsmodellerna är Naive Bayes (NB), Support Vector Machine (SVM) och ZZZZZ. Figur 2.1 visar en förenklad överblick 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 3.2. På den bearbetade datan utförs ett särdragsurval, där dataspecifika särdrag plockas ut... MER .. som diskuteras vidare i 3.3. Träningsmängden används av maskininlärningsalgoritmen för att skapa klassificeringsmodellen. När klassificeringsmodellen är framtagen används valideringsmängden 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



Figur 2.1: Utvärderingsprocessen av maskininlärningsmodeller

lösningar man vill erbjuda sina kunder. Lösningar skulle kunna vara chatbotar där 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 2.2 förklaras vad SA är och olika klassificeringstekniker inom SA beskrivs kortfattat. I avsnitt 2.3 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 2.4 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 2.5 ges en introduktion till TSA och vilka utmaning samt problem TSA har. Slutligen i avsnitt 2.6 förklaras vilka hjälpmedel som kommer att användas för att utvärdera algoritmernas prestanda.

## 2.2 Sentimentanalys

SA används för att studera människors åsikter, attityder och känslor mot andra 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 åsikter, identifiera känslan för de åsikterna och slutligen klassificera motsatsförhållandet dem emellan [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.

### 2.2.1 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 maskininlärningsmetoder.

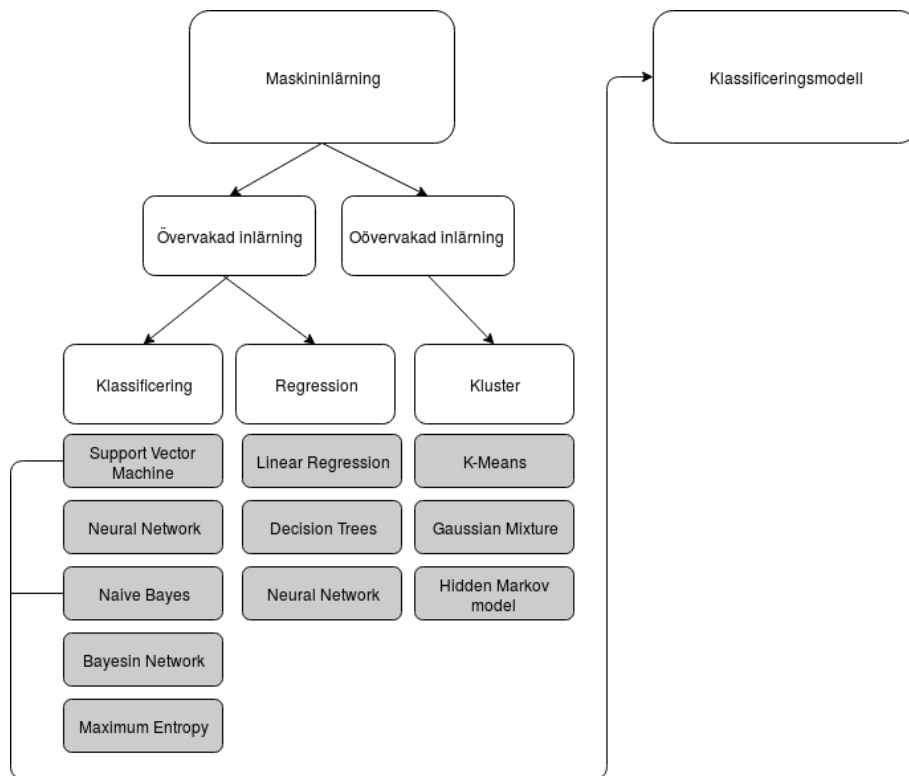
## 2.3 Maskininläring

Maskininläring 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 maskininläring 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].

Maskininläring 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 maskininläring för ansiktsgenkänning [10] på bilder som laddas upp, och för att rekommendera nya vänner [11].

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

Oövervakad inläring används när datamängden bara består av indata och inget förväntat resultat. Målet med oövervakad inläring 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 gruppera dem följaktligen. Övervakad inläring används när datamängden består av både indata



Figur 2.2: Inlärningsprocesser för maskininläring.

och dess förväntade utdata. Den övervakade algoritmen använder datamängden för att 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 inläring 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 inläring 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.

### 2.3.1 Naive Bayes

NB är en klassificeringsalgorithm 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 beror på varandra så bidrar alla särdragen självständigt till sannolikheten att frukten är ett äpple, det är därför algoritmen kallas (“naive”) [15].

### 2.3.2 Support Vector Machine

SVM är en övervakad maskininlärningsalgorithm som kan användas till både klassificerings- och regressionsproblem, för det mesta används SVM för klassifikationsproblem. SVM är baserat på iden 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].

### 2.3.3 XXX

## 2.4 Datamängd

En datamängd för övervakad inlärning består av en mängd in- och utdata som diskuteras i sektion 2.3. Processen för hur rätt udata (positivt/negativ) markeras till varje indata 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 positivt/negativt 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 felaktigt [2]. Beskrivet i sektion 2.1 är delsyftet att utvärdera hur de olika märkningsmodellerna påverkar maskininlärningsalgoritmernas precision. Det finns ett urval med publika twitter-datamängder där både mänsklig märkning och “distant supervision” används [2]. Vi har valt att använda Stanford Twitter Sentiment (STS) [19] och SemEval [20] datamängderna eftersom de är de största publika datamängderna inom respektive markeringsmodell.

### 2.4.1 SemEval

SemEval-datamängden består av 20633 tweets och är framtagen till en årlig TSA tävling som har gått sedan 2013 [21]. Datamängden är märkt med hjälp av mänsklig märkning där fem personer manuellt märker varje tweet via Amazon Mechanical Turk [22].

I [23] beskrivs metoden för hur märkningen utförs, varje person markerar varje tweet antingen som mycket positivt, positivt, neutralt, negativ eller mycket negativt. Efter att alla tweets är märkta kartläggs alla tweets till kategorierna positivt, neutral eller negativt utefter tre kriterier. Antingen att alla personer har märkt ett 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.

SemEval datamängden är fördelad utefter de ovanstående kriterierna där 2760 tweets är där alla personer märkt ett tweet samma, 9944 tweets är där majoritet har märkt samma och resterande 7928 tweets har medelvärdet räknats ut.

### 2.4.2 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 [24] ett twitterinlägg innehåller, exempelvis märks ett twitterinlägg positivt om det innehåller “:), :-), :D, =)” och negativt om det innehåller “:(, :-(, :(“ [19].

## 2.5 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 tillåtna tecken till 280 [25].

Det finns en rad olika begrepp som kännetecknar Twitter och som är viktiga att känna till [2]. En “tweet” är vad som tidigare benämnts som 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 som ett 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 twitterflöde 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.



RT @johanselberg Exempeltweet med en hashtag #exempel, mention av användaren @KAU och länken <https://www.kau.se>

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

### 2.5.1 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 utforskat område.

### 2.5.2 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.

Tabell 2.1: Förvirringsmatris

	Förutspådd positiv	Förutspådd negativ
Positiv	Sann positiv	Falsk negativ
Negativ	Falsk positiv	Sann negativ

## 2.6 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 positiva (SP), sann negativ (SN), falsk positiv (FP) och falsk negativ (FN) [26]. Med dessa värden kan vi jämföra och analysera modellerna m.h.a följande utvärderingsmetoder: noggrannhet (n) [27], precision (p), återkallelse (å) [28] och F-Score [29].

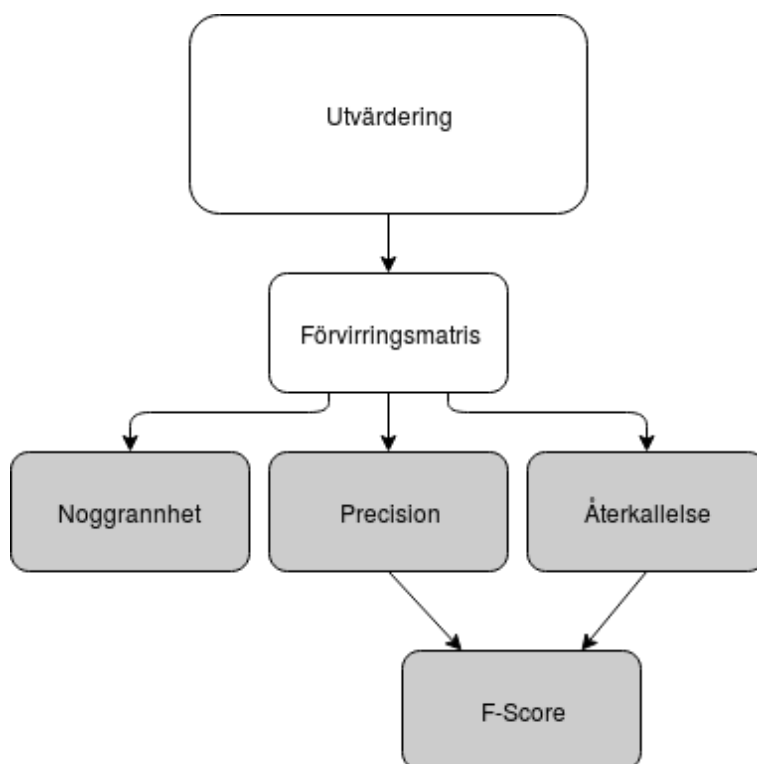
**Noggrannhet** Är modellens förmåga att kunna märka ett tweet korrekt som antingen positivt eller negativt. Detta görs genom att ta summan av sann märkta tweets delat på summa av alla märkningar.

**Precision** Är förmågan att modellen inte märker ett tweet som positivt när det 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.

## 2.7 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 maskininlärnings 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

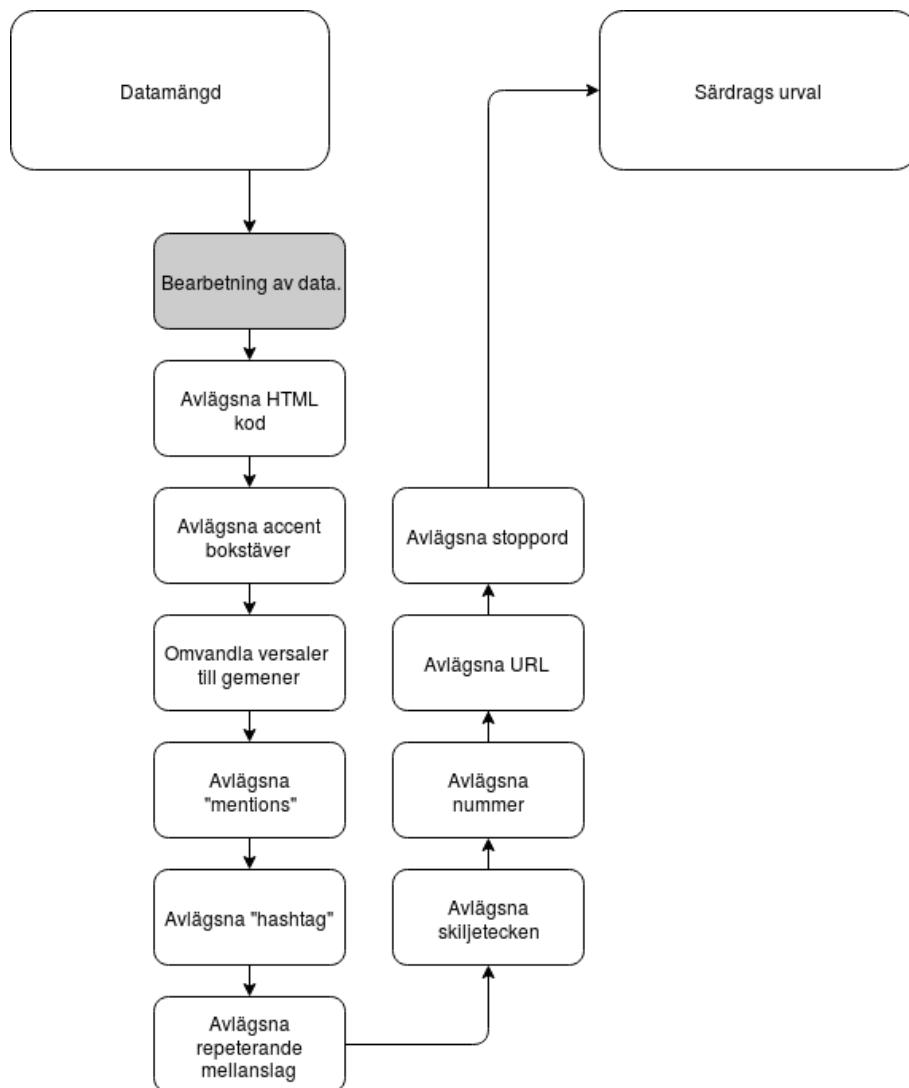
### 3.1 Intro

I kapitlet kommer experimentet för studien att presenteras, hur det har utförts och hur de olika delarna har implementeras för att utföra experimentet. Såsom bearbetning av datamängderna, vilka särdragsurval som kommer utföras och en fördjupning i hur maskininlärningsalgoritmerna fungerar. Implementationen kommer ske i tre steg där först en lexikonbaseradmodell 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 för att eventuellt uppnå en förbättrad klassificeringsmodell för respektive datamängd. De slutgiltiga klassificeringsmodellerna kommer att diskuteras och jämföras i avsnitt 4

### 3.2 Bearbetning av datamängd

En viktig del inom SA är bearbetningen av den data som ska analyseras. Som nämndes i 2.5.1, innehåller twitterinlägg stora mängder brus på grund av det

informella språket som används på Twitter, dvs data som inte är användbar för analysprocessen. Kvaliteten på datan som ska analyseras är avgörande för vilka resultat som kan uppnås vid maskininlärningen [30]. Därför behöver data som ska analyseras bearbetas innan maskininlärningen, vilket betyder att data normaliseras och rensas på brus. Exempelvis kan det bestå av att termer som inte är nödvändiga tas bort, ersätts eller slås samman med andra.



Figur 3.1: Bearbetningsprocessens olika steg.

Det existerar flertalet olika tekniker som kan användas vid bearbetningen av da-

ta. I [30] utvärderas 15 av dessa bearbetningstekniker. Teknikerna utvärderas var för sig och testas mot två olika datamängder, där de för varje datamängd testas med tre olika maskininlärningsalgoritmer. Vid utvärderingen av bearbetningsteknikerna studeras noggrannheten vid sentimentklassificeringen samt antalet särdrag som genereras för varje bearbetningsteknik. Ur resultaten konstaterar författarna att det inte existerar någon direkt koppling mellan noggrannheten för klassificeringsmodellerna och antalet genererade särdrag. Författarna sammanställer resultaten från utvärderingen och kategoriserar de olika bearbetningsteknikerna i fem olika kategorier efter noggrannhet. 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 bearbetningstekniker som 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. Det som saknas i [30] är hur de olika förbehandlingsteknikerna presterar i kombination med varandra.((( Nämna att de pysslar med three point classification?? ))

[31] undersöker också effekten av bearbetningstekniker vid TSA. Men istället för att utvärdera varje teknik isolerat från varandra, utvärderas de istället utifrån hur de presterar i kombination med andra. Författarna har valt att fokusera på 6 olika bearbetningstekniker: ersättning av negationer, 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 2 olika klassificeringsuppgifter: binär och 3-vägs klassificering (kanske behövs förklaras vad 3-vägs klassificering?). Där teknikerna testas med 2 särdragsmodeller (hur ska vi hålla oss till begrepp som nämns i sektionen efter?) och 4 klassificeringsmodeller på 5 olika datamängder.

Vid bedömningen av prestandan vid sentimentklassificeringen har ett basfall använts. Basfallet har använt samtliga 6 bearbetningstekniker för bearbetningen av datan vid testningen. När en specifik teknik har utvärderats, har tekniken i frå-

ga exkluderats från mängden av bearbetningstekniker. Därefter har testerna körts igen, med 5 bearbetningstekniker istället för 6. Sedan har förändringen av resultatet gentemot basfallet använts för att bedöma hur tekniken påverkar prestandan vid sentimentklassificeringen. Vid utvärderingen av prestandan har noggrannhet och F-score använts som måttvärden.

Resultaten visar att ersättning av negationer och expanderings av förkortningar kan ha positiv påverkan på noggrannheten vid klassificeringen. Avlägsnande av URL:er, nummer och stoppord påverkar resultaten för klassificeringen minimalt, men är effektiva för att minska brus.

(Figur 3.1 kommer att nämnas och beskrivas i nästa stycke som inte är klart ännu)

## 3.3 Särdragsurval

## 3.4 Algoritmer

\*\* Skall det skrivas något kort intro här?\*\*\*

### 3.4.1 Naive Bayes

Det finns tre varianter utav NB nämligen Gaussian, Multinomial och Bernoulli. Gaussian används när attributen är kontinuerligt fördelade och då antas det att attributen förknippade med varje klass fördelas enligt Gaussian d.v.s normalfördelning [32]. Bernoulli användas när attributen är fördelade enligt multivariat Bernoulli [33] och Multinomial NB (MNB) används när attributen är multinomial fördelat. I [34] diskuteras det att MNB är väl lämpad för text klassificering vilket gör att studien kommer använda MNB.

I [35] diskuteras MNB där text behandlas som en sekvens av ord där det antas att varje ords position är oberoende från varandra. För klassificering antas det finnas ett fast antal klasser,  $c \in \{1, 2, \dots, m\}$ , där var och en har en fast uppsättning av multinomiala parametrar. För varje klass finns en parametervektor,

$\vec{\theta}_c = \{\theta_{c1}, \theta_{c2}, \dots, \theta_{cn}\}$ , där  $n$  är storleken på ordlistan,  $\sum_i \theta_{ci} = 1$  och  $\theta_{ci}$  är sannolikheten att ordet  $i$  förekommer i klassen. Sannolikheten för en text är produkten av ordens parametrar som återfinns i texten, där  $f_i$  är antalet förekomster av ordet  $i$  inuti texten  $t$ .

$$p(t|\vec{\theta}_c) = \frac{(\sum_i f_i)}{\prod_i f_i!} \prod_i (\theta_{ci})^{f_i} \quad (3.1)$$

Genom tilldelning av en priorfördelning [36],  $p(\vec{\theta}_c)$ , kan vi härleda en miniklassificeringsregeln [37] som väljer klassen med störst sannolikhet, där  $b_c$  är tröskelvärde och  $w_{ci}$  är klassens vikt för ord  $i$ .

$$l(t) = \operatorname{argmax}_c [\log(\vec{\theta}_c) + \sum_i f_i \log \theta_{ci}] \quad (3.2)$$

$$= \operatorname{argmax}_c [(b_c + \sum_i f_i w_{ci})] \quad (3.3)$$

För binär klassificering är detta enklare eftersom gränsen sätts genom att sätta skillnaderna mellan de positiva och negativa klass parametrarna till noll.

$$(b_+ - b_-) + \sum_i f_i (w_{+i} - w_{-i}) = 0$$

Eftersom klasser och märkningen av klasserna är bestämda av datamängden måste klassernas parametrar estimeras på något sätt. Detta utförs genom en Dirichlet prior [38][39] som tar de förväntade parametrarna med avseende på den bakre fördelningen. Dirichlet prior ger oss en enkel form för att uppskatta våra multinomiala parametrar, vilket består av antal gånger ordet [matte i] förekommer i texterna för den förväntade klassen ( $N_{ci}$ ), delat med det totala antalet ord i klassen ( $N_c$ ). För ord  $i$  adderas en så kallad additiv utjämning [40],  $\alpha_i$ , som även kallas Laplace utjämning och  $\alpha$  betecknar summan av  $\alpha_i$ .

$$\hat{\theta}_{ci} = \frac{N_{ci} + \alpha_i}{N_c + \alpha} \quad (3.4)$$

Genom att substituera  $\theta_{ci}$  från 3.2 med det framräknade  $\hat{\theta}_{ci}$  får vi den slutgiltiga MNB algoritmen.

$$l_{MNB}(t) = \operatorname{argmax}_c [\log(\vec{\theta}_c) + \sum_i f_i \log \hat{\theta}_{ci} = \frac{N_{ci} + \alpha_i}{N_c + \alpha}]$$

### 3.4.2 Support Vector Machine

## 3.5 Implementation

### 3.5.1 Naive Bayes

### 3.5.2 Support Vector Machine

### 3.5.3 Prediction program.

### 3.5.4 \* ev GUI implementation om tid finns \*

## 3.6 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](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](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](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](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](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](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](https://en.wikipedia.org/wiki/Support_vector_machine). [Online; accessed 12-February 2018].
- [19] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.
- [20] International workshop on semantic evaluation 2017, 2016. URL <http://alt.qcri.org/semeval2017/>. [Online; accessed 12-February 2018].
- [21] Sentiment analysis in twitter, 2016. URL <http://alt.qcri.org/semeval2017/task4/>. [Online; accessed 12-February 2018].
- [22] Amazon. Amazon mechanical turk, n.d. URL <https://www.mturk.com/product-details>. [Online; accessed 12-February 2018].
- [23] 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.
- [24] Wikipedia contributors. Emoji — Wikipedia, the free encyclopedia, 2014. URL <https://sv.wikipedia.org/wiki/Emoji>. [Online; accessed 5-Februari 2018].
- [25] Wikipedia contributors. Twitter — Wikipedia, the free encyclopedia, 2007. URL <https://en.wikipedia.org/wiki/Twitter>. [Online; accessed 5-Februari 2018].

- [26] Wikipedia contributors. Confusion matrix — Wikipedia, the free encyclopedia, 2014. URL [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix). [Online; accessed 29-January 2018].
- [27] Wikipedia contributors. Accuracy and precision — Wikipedia, the free encyclopedia, 2002. URL [https://en.wikipedia.org/wiki/Accuracy\\_and\\_precision](https://en.wikipedia.org/wiki/Accuracy_and_precision). [Online; accessed 5-Februari 2018].
- [28] Wikipedia contributors. Precision and recall — Wikipedia, the free encyclopedia, 2016. URL [https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall). [Online; accessed 29-January 2018].
- [29] Wikipedia contributors. F1 score — Wikipedia, the free encyclopedia, 2014. URL [https://en.wikipedia.org/wiki/F1\\_score](https://en.wikipedia.org/wiki/F1_score). [Online; accessed 29-January 2018].
- [30] 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.
- [31] 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.
- [32] Rahul Saxena. How the naive bayes classifier works in machine learning, 2017. URL <http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/>. [Online; accessed 12-February 2018].
- [33] Stephanie. Bernoulli distribution: Definition and examples, 2017. URL <http://www.statisticshowto.com/bernoulli-distribution/>. [Online; accessed 12-February 2018].
- [34] scikitlearn documentation. 1.9 naive bayes – scikitlearn documentation, 2017. URL [http://scikit-learn.org/stable/modules/naive\\_bayes.html](http://scikit-learn.org/stable/modules/naive_bayes.html). [Online; accessed 12-February 2018].
- [35] Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the*

- 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.
- [36] Andrew Gelman. Prior distribution. *Encyclopedia of environmetrics*, 2002.
- [37] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification and scene analysis 2nd ed. *ed: Wiley Interscience*, 1995.
- [38] Wikipedia contributors. Dirichlet distribution — Wikipedia, the free encyclopedia, 2004. URL [https://en.wikipedia.org/wiki/Dirichlet\\_distribution](https://en.wikipedia.org/wiki/Dirichlet_distribution). [Online; accessed 26-Februari 2018].
- [39] David Heckerman. A tutorial on learning with bayesian networks. In *Learning in graphical models*, pages 301–354. Springer, 1998.
- [40] Wikipedia contributors. Additive smoothing — Wikipedia, the free encyclopedia, 2008. URL [https://en.wikipedia.org/wiki/Additive\\_smoothing#Pseudocount](https://en.wikipedia.org/wiki/Additive_smoothing#Pseudocount). [Online; accessed 26-Februari 2018].