

Assignment 1

Thijs Veldkamp

50058543



25 juni 2023

Inhoud

Inhoud

[1.Samenvatting 2](#_Toc138624137)

[2.Inleiding 4](#_Toc138624138)

[3. Methodes 5](#_Toc138624139)

[3.1 Data discovery 5](#_Toc138624140)

[3.1.1 Webscraping 5](#_Toc138624141)

[3.1.2 Zelfgeschreven dataset 6](#_Toc138624142)

[3.1.3 Kaggle dataset. 7](#_Toc138624143)

[3.1.4 Samengevoegde dataframe. 8](#_Toc138624144)

[3.2 Data preparation 8](#_Toc138624145)

[3.2.1 Dataset in database laden. 8](#_Toc138624146)

[3.2.2 data cleaning 10](#_Toc138624147)

[3.2.3 Wordclouds 10](#_Toc138624148)

[3.2.4 Lemmatisering en Stemming 12](#_Toc138624149)

[3.2.5 Vectorizers 14](#_Toc138624150)

[3.3 Model planning 16](#_Toc138624151)

[3.3.1 Wat zijn machine learning modellen? 16](#_Toc138624152)

[3.3.2 Random Forest. 17](#_Toc138624153)

[3.3.2 Multinominal Naïve Bayes 19](#_Toc138624154)

[3.3.3 Logistic Regression 20](#_Toc138624155)

[3.4 Model building 22](#_Toc138624156)

[3.4.1 Random Forest 22](#_Toc138624157)

[3.4.2 Multinominal Naïve Bayes 23](#_Toc138624158)

[3.4.3 Logisitic regression 24](#_Toc138624159)

[4.Resultaten 25](#_Toc138624160)

[4.1 wat waren de scores. 25](#_Toc138624161)

[4.1.1 Accuracy 25](#_Toc138624162)

[4.1.2 Precision 25](#_Toc138624163)

[4.1.3 Confusion matrix 26](#_Toc138624164)

[4.1.4 Roc\_auc\_score 27](#_Toc138624165)

[5.Conclusie 28](#_Toc138624166)

# 1.Samenvatting

Dit verslag beschrijft een big data-opdracht waarbij een dataset is verzameld, bestaande uit hotelreviews. De dataset is verkregen van Kaggle, terwijl een ander deel is verkregen door webscraping van boekingssites. Deze datasets zijn vervolgens samengevoegd tot één dataset.

De volgende stap was het labelen van de reviews als positief of negatief, gevolgd door het opslaan van de dataset in een database. Vervolgens is de dataset uit de database gehaald om datacleaning toe te passen, inclusief stemming en lemmatisering. Daarna zijn vectorizers ingesteld om de dataset geschikt te maken voor machine learning-modellen.

Om de resultaten te verkrijgen, zijn verschillende modellen toegepast, waaronder Random Forest, Multinominal Naive Bayes en Logistic Regression. Om de modellen te verbeteren, is ook een lijst met stopwoorden gemaakt.

De resultaten van de modellen worden gepresenteerd en vervolgens wordt een conclusie gegeven. Deze conclusie zal de bevindingen samenvatten en eventueel aanbevelingen doen voor verdere verbeteringen of toepassingen van de dataset.

# 2.Inleiding

In dit verslag wordt beschreven welke stappen ik heb genomen voor het bouwen van een model om positieve en negatieve hotel reviews te voorspellen.

Als eerst zal ik het data discovery onderdeel uitleggen. Hier leg ik uit hoe ik aan de verschillende datasets ben gekomenen en hoe ik ze heb samengevoegd en gebruikt.

In het data preparation onderdeel leg ik uit hoe ik de datasets heb opgeslagen in een database en wat ik heb gedaan zodat de modellen met de data kunnen werken.

In het derde onderdeel leg ik uit welke modellen ik heb gekozen, waarom ik voor deze modellen heb gekozen en de verschillende technieken om hun effectiviteit te verbeteren, zoals gegevenssplitsing en parameter afstem technieken.

Tot slot leg ik uit wat de resultaten zijn.

**Beschrijving van de opdracht**

De aanleiding van deze opdracht was om alle kennis van BDSE van blok 1 van het semester te combineren in 1 opdracht. Voor deze opdracht moeten de volgende doelstellingen worden behaald.

-Minstens 100 hotel reviews van verschillende websites moesten worden verzameld via webscraping en in een Dataframe worden gezet.

-De Student schrijft zelf minstens 10 reviews zelf en zet deze in een dataframe

-Zorg ervoor dat de dataset die je hebt ontvangen van school in een dataframe komt met de relevante kolommen.

-Combineer al deze datasets in 1 dataframe.

-De data moet worden gelabeld en in een Database worden gezet. Verder moet er ook data cleaning worden uitgevoerd.

-De student kan 3 verschillende modellen uitleggen en beargumenteren welke beter zijn.

De student gebruikt het model voor zijn script en voert analyses uit op de data.

Afbeelding met tekst, clipart, Graphics, Lettertype

Automatisch gegenereerde beschrijving

# 3. Methodes

## 3.1 Data discovery

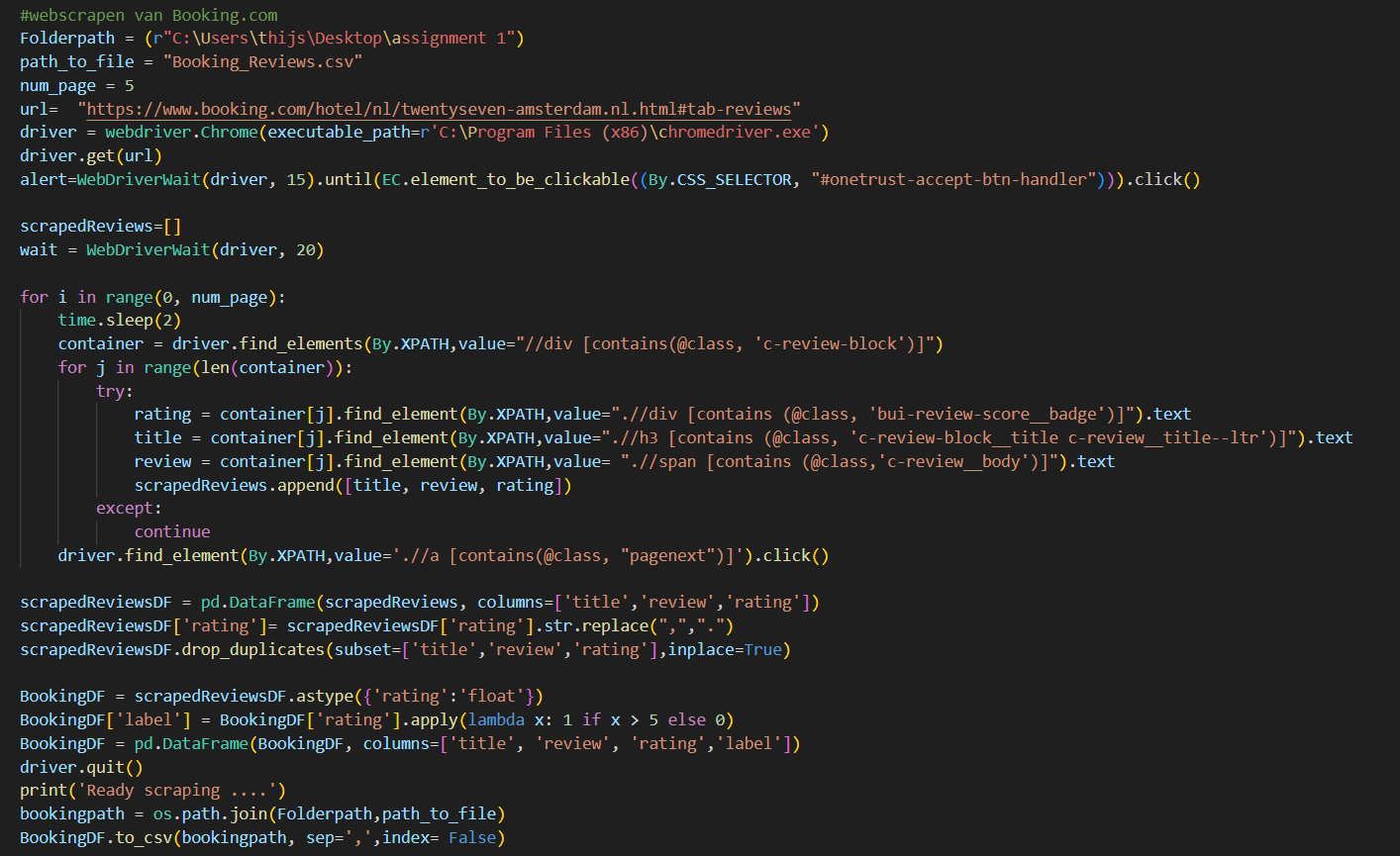
In het onderdeel data discovery verzamel ik data van verschillende databronnen. Het is de bedoeling om de verschillende databronnen in een dataframe te laden. De databronnen zijn reviews die ik van hotel booking websites heb gehaald door middel van webscraping, Zelfgeschreven reviews en een Kaggle dataset.

### 3.1.1 Webscraping

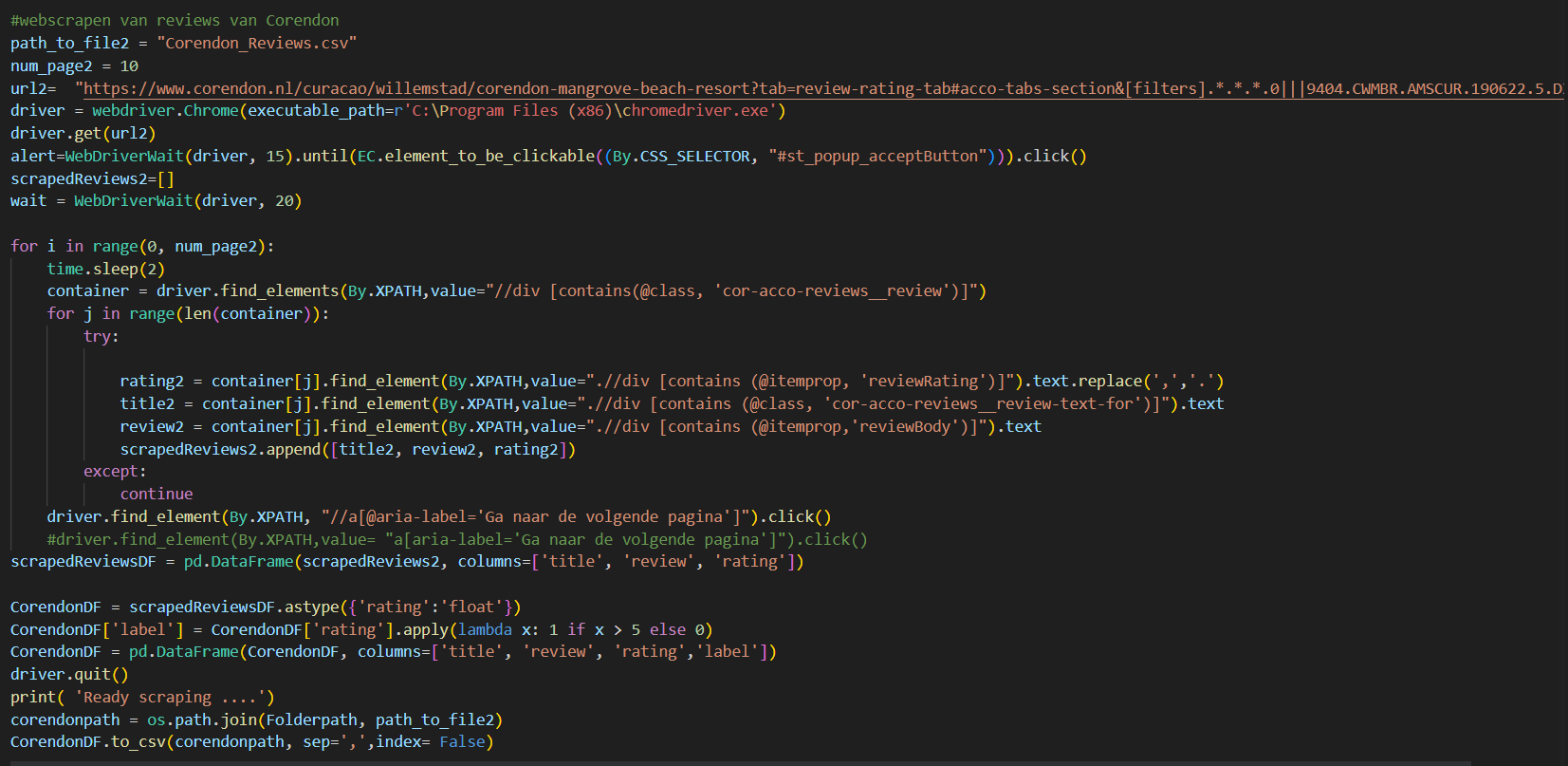
Voor het verzamelen van hotel reviews maakte ik gebruik van webscrapen via Selenium. Hiervoor moest ik eerst een aantal prerequisites installeren zoals selenium en de Chrome webdriver. Verder waren er ook nog een paar functies zoals Pandas die ik installeerde via pip install en importeerde in python. Ik maakte deels gebruik van de template die op DLO stond voor Tripadvisor. De template was verouderd waardoor ik de find\_element regels moest aanpassen. Verder heb ik de template aangepast om van Booking.com en de website van Corendon reviews te webscrapen. Het is me gelukt om meer dat 100 reviews te scrapen.

Voor het accepteren van de cookies van de websites die ik ging webscrapen heb ik de path gekopieerd van de css. selector in plaats van de Xpath. Soms lukte het niet om de Xpath te gebruiken. Voor het inlezen van de titel, rating en review. Verder heb ik ook de xpath voor het klikken naar de volgende pagina via de XPATH laten klikken.

Vervolgens heb ik ervoor gezorgd dat de coma’s in de rating worden veranderd naar een punt en converteerde de rating vervolgens van string naar float. Voor het labelen zorgde ik ervoor dat reviews met een rating hoger dan 5 worden gelabeld met 1 en zo niet met 0.



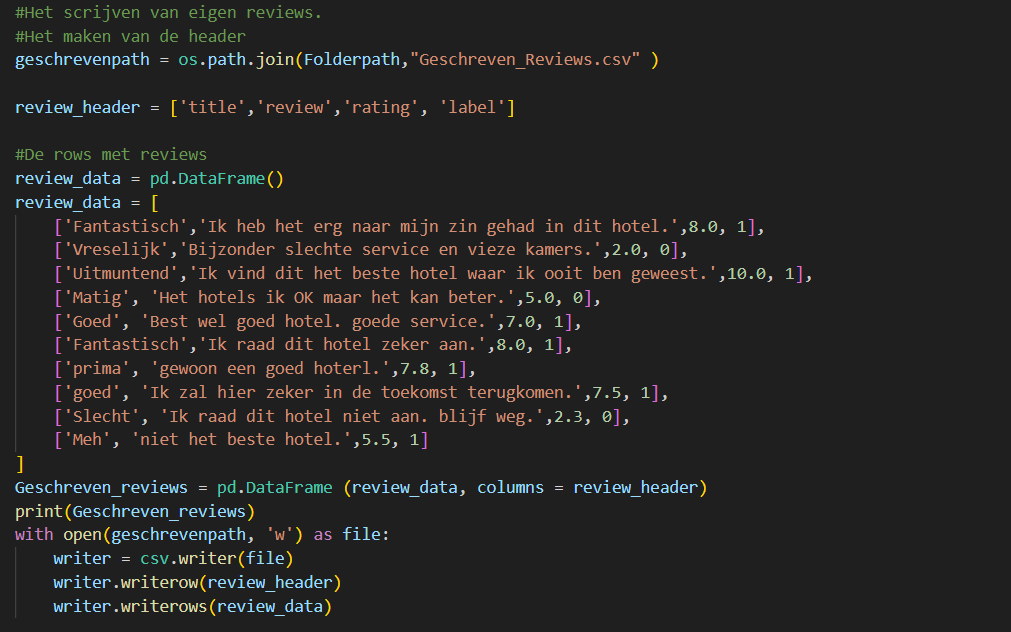
1 Booking.com webscraper



Corendon webscraper

### 3.1.2 Zelfgeschreven dataset

Ik heb zelf 10 reviews geschreven, waarvan 7 positief zijn en 3 negatief zijn. Ik labelde de positieve reviews met 1 en negatieve met 0. Ik zorgde ervoor dat in de review header de kolommen dezelfde namen hadden als de gewebscrapte datasets. Tot slot maakte ik er een CSV-file van.



Zelfgeschreven dataset

### 3.1.3 Kaggle dataset.

Dit is de grootste dataset. Het is een CSV-bestand en het bevat 515.739 reviews van 1493 hotels in Europa. De velden in het bestand zijn als volgt:

Review\_Date

Hotel\_Address:

Average\_Score:

Hotel\_Name:

Reviewer\_Nationality:

Negative\_Review:

ReviewTotalNegativeWordCounts:

Positive\_Review:

ReviewTotalPositiveWordCounts:

Reviewer\_Score:

TotalNumberofReviewsReviewerHasGiven:

TotalNumberof\_Reviews:

Tags:

Dayssincereview:

AdditionalNumberof\_Scoring:

Van deze data set heb ik alleen de velden Hotel\_name, Positive\_review, Negative\_review en Reviewer\_score. De andere kolommen heb ik daarom gedropt. Ik heb vervolgens de melt functie gebruikt om van positive en negative review 1 kolom te maken. Verder zorgde ik ervoor dat positieve reviews gelabeld worden met 1 en negatieve reviews met 0. Ook wanneer de rating hoger is dan een 5 wordt de review positief.



Kaggle dataset naar dataframe

### 3.1.4 Samengevoegde dataframe.

Tot slot heb ik alle CSV-files samengevoegd naar 1 CSV file door middel van de glob functie. De CSV-file heb ik vervolgens weer ingelezen in python en in een dataframe gezet genaamd DF1. Tot slot heb ik alle duplicaten uit de dataframe gegooid.



Samengevoegde dataframe

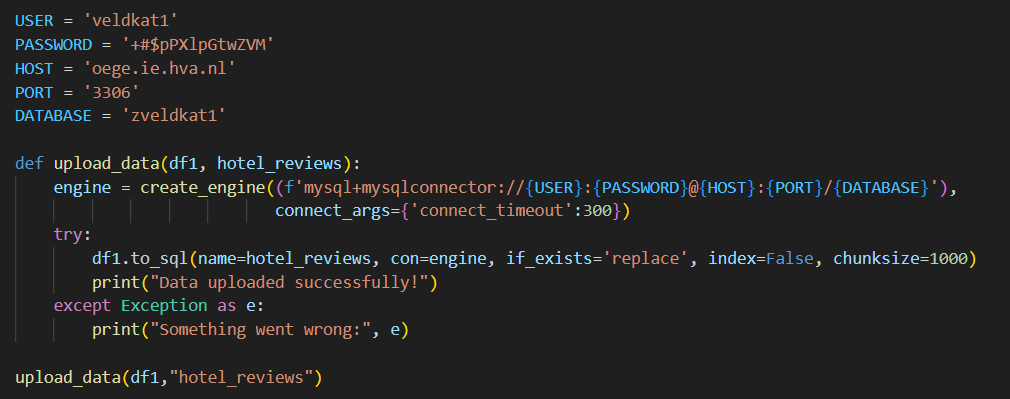
## 3.2 Data preparation

Het deel data preparation gaat over het cleanen van transformeren van ruwe data voorafgaand aan de verwerking en analyse. Het is een belangrijke stap voorafgaand aan de verwerking en omvat vaan data opnieuw formatteren, corrigeren en data verrijken. Hierdoor wordt de datakwaliteit beter zodat de modellen er betere inzichten uit kunnen halen.

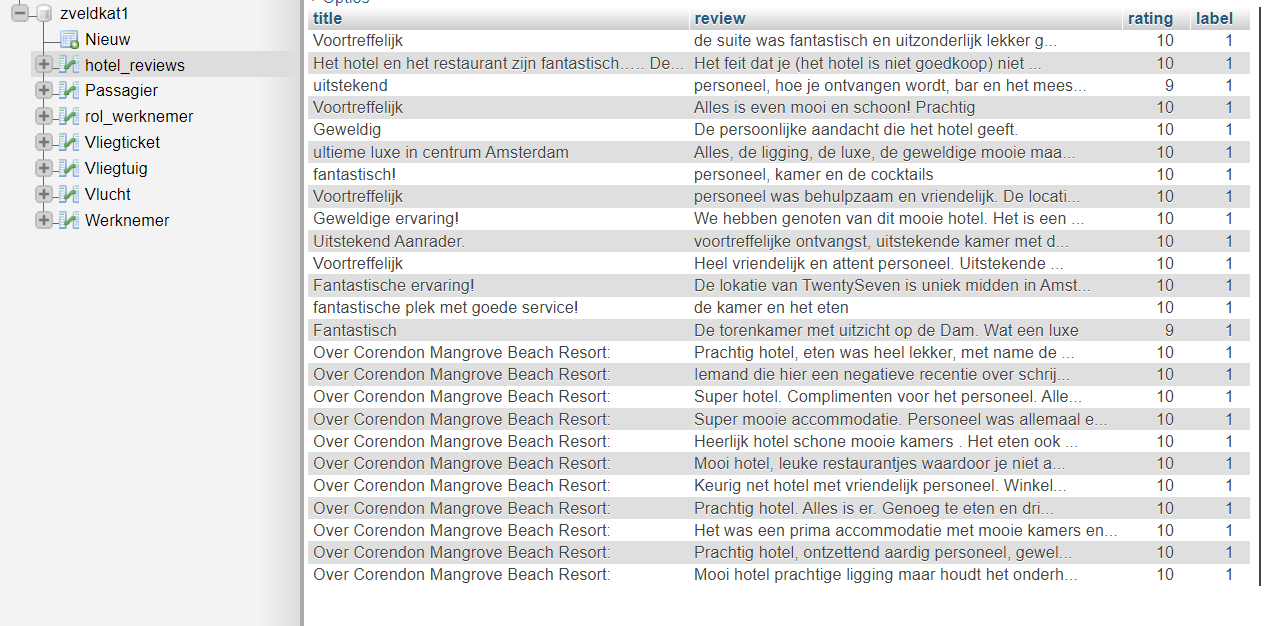
In dit gedeelte zal ik uitleggen hoe de gegevens zijn opgeslagen en verwerkt zodat de modellen ermee kunnen werken.

### 3.2.1 Dataset in database laden.

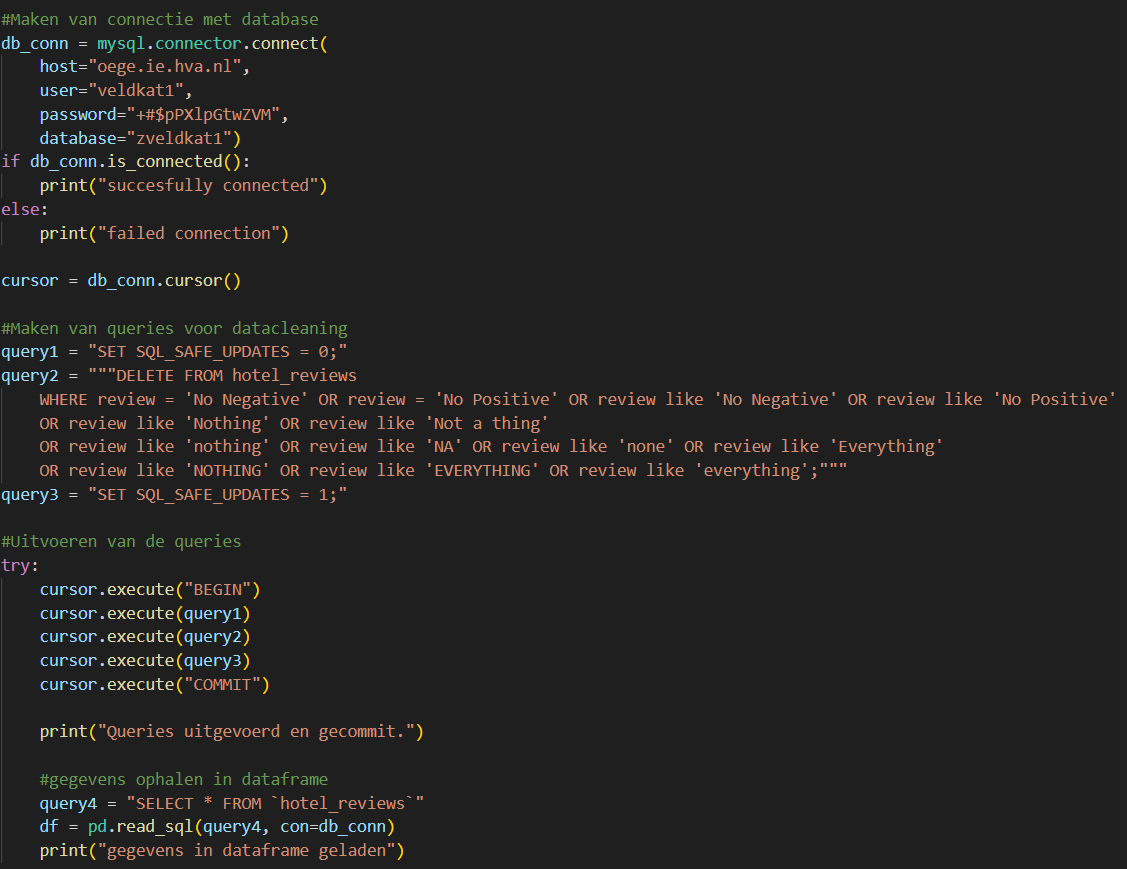
Om de dataframe in de database van de HVA te maken had ik een functie geschreven die een connectie maakt met de Oege server. De naam van de database is zveldkat1. Ik noemde de dataframe hotel\_reviews. In het geval dat de data succesvol was geüpload print de functie “Data uploaded successfully! In het geval dat het niet lukt print de functie “Something went wrong”



Functie voor het uploaden van dataframe in database



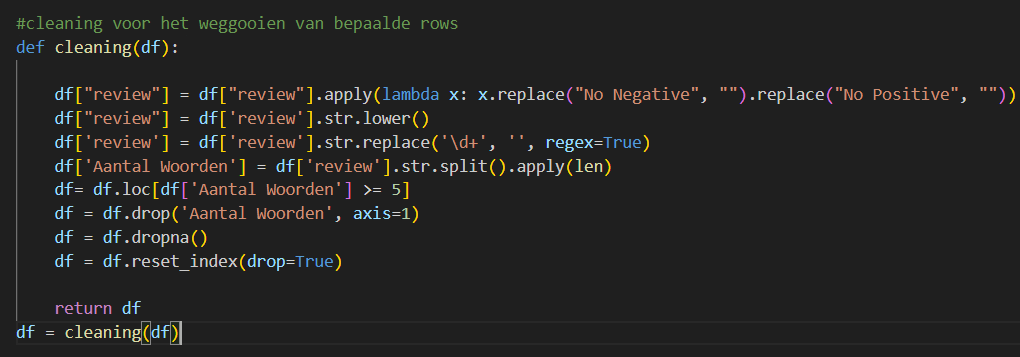
Database

Met de data succesvol geüpload kon ik over gaan naar het data cleanen. Om te beginnen maakte ik opnieuw verbinding met database en schreef ik in python een paar query’s om reviews die bepaalde woorden bevatten te verwijderen.  


Uitvoeren van query's en inladen van dataframe

### 3.2.2 data cleaning

Verder maakte ik een functie die alle hoofdletters naar kleine letters converteert, alle nummers uit de reviews haalt en om alle reviews die leeg zijn of minder dan 5 woorden uit de dataframe te verwijderd. Tot slot reset de functie de index.



Data cleaning functie

### 3.2.3 Wordclouds

Een van de eisen voor de opdracht was om word clouds te maken. Na het cleanen van de data ben ik daarom begonnen met het maken van 3 verschillende wordclouds.

Een wordcloud, ook wel woordwolk, is een visuele representatie van een verzameling woorden waarbij de woorden die het meest voorkomen groter worden weergegeven. Het doel van een wordcloud is om de frequentie en het belang van verschillende woorden in een tekst of dataset op een beknopte en visuele manier weer te geven.

In een wordcloud worden vaak gebruikt om snel inzicht te krijgen in de belangrijkste trends in een tekstcorpus door de belangrijkste woorden te identificeren zonder de tekst volledig te hoeven lezen. Ze worden toegepast in verschillende domeinen zoals tekstanalyse, datavisualisatie en een marktonderzoek.

Om te beginnen maakte in een aantal stopwoorden aan door algemene Nederlandse en Engelse woorden te downloaden. Vervolgens maakte ik een algemene word cloud. Hieruit kwamen de meest voorkomende woorden die ik vervolgens toevoegde aan een lijst met mijn eigen stopwoorden om vervolgens een positieve en negatieve wordcloud te kunnen maken.



Stopwords

Afbeelding met tekst, Lettertype, Graphics, grafische vormgeving

Automatisch gegenereerde beschrijving

Algemene word cloud

De meest voorkomende woorden Zijn Hotel, Room, staff en breakfast. Deze woorden heb ik vervolgens aan de stopwoord lijst toegevoegd om de andere 2 word clouds te maken.

Afbeelding met tekst, Lettertype, Graphics, schermopname

Automatisch gegenereerde beschrijving

Positieve word cloud

De positieve woorden bevatten vooral woorden zoals: Helpful. Friendly, clean, nice en good.

Afbeelding met tekst, Lettertype, Graphics, grafische vormgeving

Automatisch gegenereerde beschrijving

Negatieve word cloud

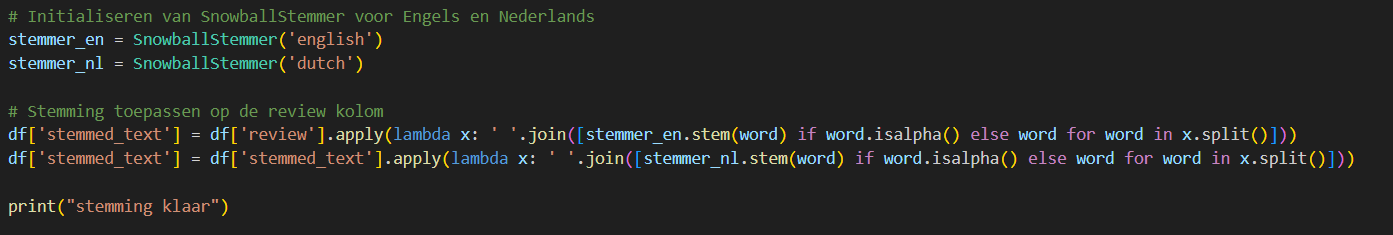
De negatieve woorden zijn iets minder duidelijk aangezien location, smal en one nogal algemene woorden zijn. Toch zijn er woorden zoals dirty, bad, poor en rude te vinden in deze word cloud.

### 3.2.4 Lemmatisering en Stemming

Het doel van lemmatisering en stemming is het verwerken en normaliseren van tekstuele gegevens, met name woorden, om ze beter bruikbaar te maken voor tekstverwerkingstaken van bijvoorbeeld machine learning modellen.

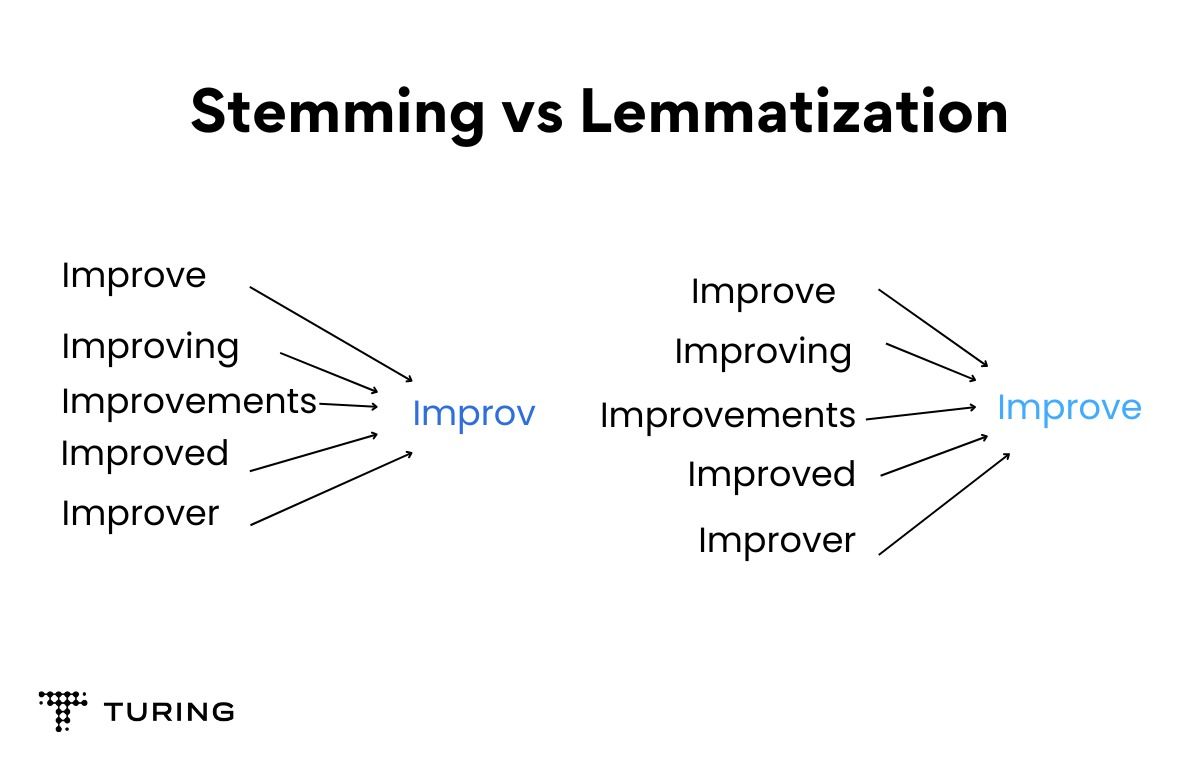
**Stemming**in machine learning kan stemming worden toegepast om woorden te reduceren tot hun gestemde vorm. Machine learning modellen kunnen hierdoor mogelijk beter presteren bij taken zoals tekstclassificatie waarbij de nadruk ligt op de betekenis van de woorden.

Nadelen van stemming zijn overstemming en onder stemming. Dit is wanneer het reduceren van woorden naar de stam te overdreven of te weinig wordt gedaan. Overstemming treedt op wanneer niet-verwante woorden naar dezelfde vorm worden afgeleid, waardoor het onderscheid tussen hen verloren gaat.



**Lemmatisering**Lemmatisering kan worden toegepast als pre-processing stap in machine learning. Door woorden terug te brengen naar hun lemma’s kunnen machine learning modellen een beter begrip hebben van de semantiek van de tekst. Hierdoor presteren de modellen beter bij taken zoals informatieverwerking, waarbij het identificeren van de onderliggende betekenis van woorden belangrijk is.

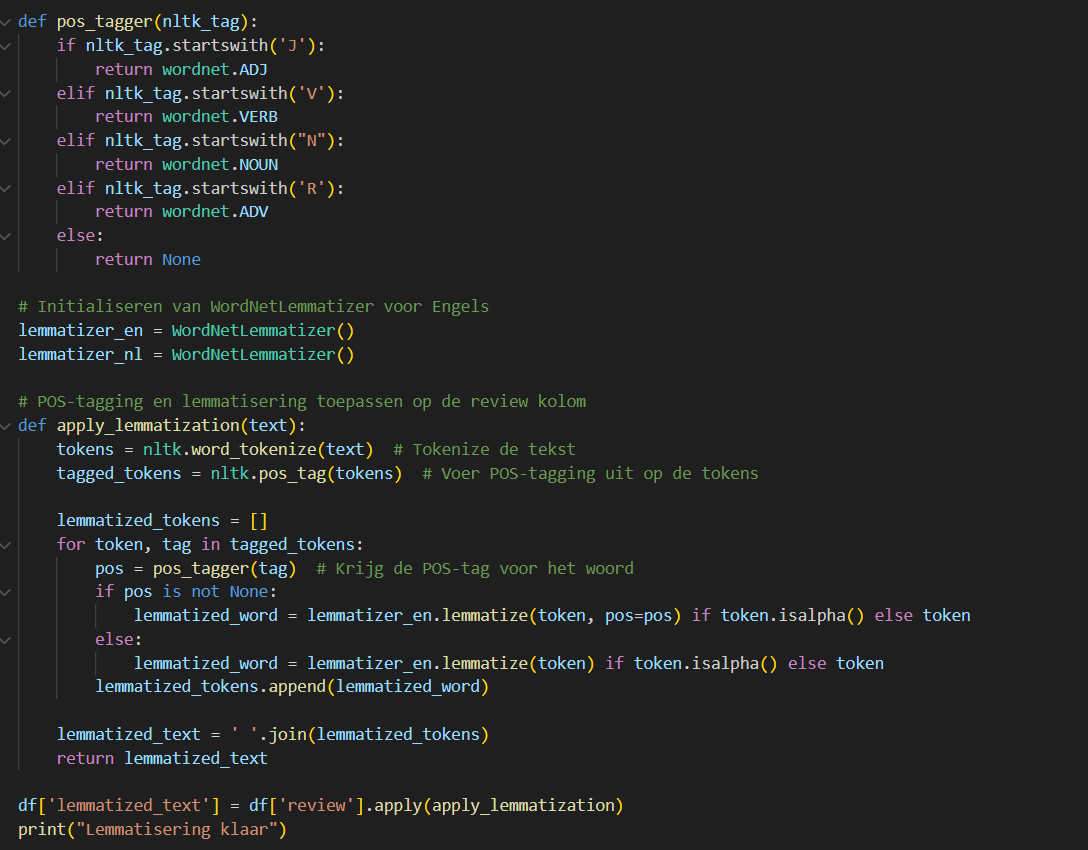
Een nadeel van lemmatisering is dat het vaak toegang vereist tot woordenboeken of taalbronnen om het juiste lemma van een woord te identificeren. Deze extra verwerking kan invloed hebben op de prestaties en verwerking van taken.



Voorbeeld stemming en lemmatization van Turing.com

Ik heb zelf stemming toegepast door de SnowballStemmer te gebruiken. De reden dat ik hiervoor heb gekozen is omdat Nederlandse en Engelse woorden in mijn dataset heb zitten. De Snowballstemmer is een uitbreiding van de Porterstemmer en bevat meerdere talen.

Voor Lemmatisering heb ik de WordNetLemmatizer gebruikt.



De pos\_tagger functie is verantwoordelijk voor het toewijzen van een WordNet POS-stag (Part of Speech) op basis van de gegeven tag van de NLTK-bibliotheek. De tag definieert het type van het woord. Hierdoor weet de Lemmantizer of hij het woord moet omzetten in een werkwoord, zelfstandig naamwoord of bijvoeglijk. Tot slot voeg ik een kolom toe aan de dataframe waar de tekst van de reviews is gelemmantiseerd.

Omdat het stemmen en lemmantiseren enorm lang duurde heb ik de nieuwe dataframe met de kolommen stemmed\_text en lemmatized\_text geüpload in de database om die later te kunnen gebruiken. Op deze manier hoef ik niet lang te wachten voordat het stemmen en lemmantiseren klaar is. Ik pas hierna pas vectoriseren toe omdat ik verschillende vectorisatiemethoden uit wil proberen.

### 3.2.5 Vectorizers

De volgende stap van data preparation is om de review kolom om te zetten in meerdere kolommen, zodat de classificatiemodellen met de gegevens kunnen werken. Hiervoor heb ik de Count Vectorize methoden gebruikt.

CountVectorizer maakt het makkelijk om tekstgegevens direct te gebruiken in machine learning en deep-learning modellen zoals tekstclassificatie. CountVectorizer van de sklearn bibliotheek dat tekstdocumenten converteert naar een matrix tokens

Afbeelding met tekst, lijn, nummer, maatstok

Automatisch gegenereerde beschrijving

CountVectorizer van researchgate.net



Vectorizers

Deze functie werkt krijgt als input DF en de combined\_stopwords. Dit was een combinatie van Nederlandse, Engelse stopwoorden en eigen stopwoorden.

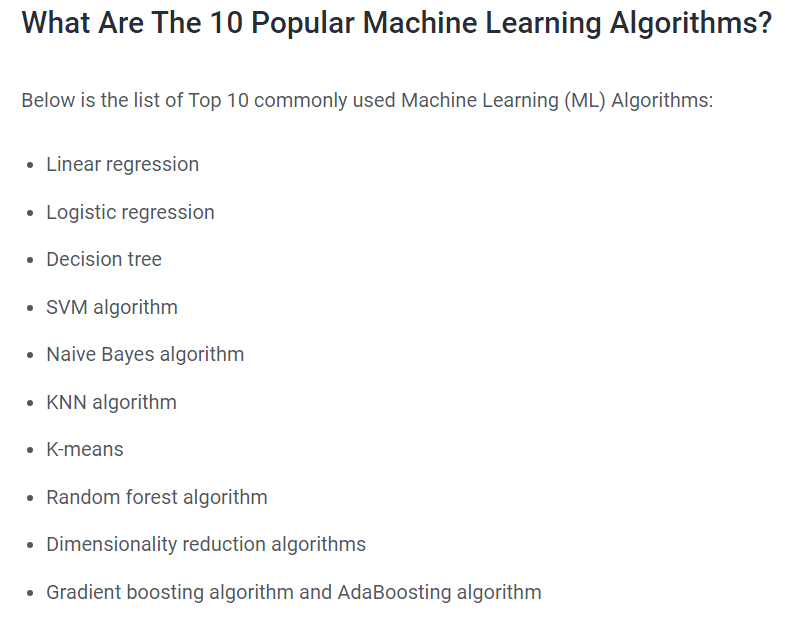
**Ngram\_range** is een parameter om het bereik van n-grams te specificeren. Een n-gram is een opeenvolging van woorden. 1 wordt gevormd door enkele woorden, 2 gram door opeenvolgende paren van woorden en 3 gram door opeenvolgende 3-tallen. Range (1, 2) betekent dat zowel 1-grams als 2-grams zullen worden gegenereerd. Dit kan helpen om contextuele informatie uit de tekst te halen.

**Max\_features** is een parameter die aangeeft hoeveel features er maximaal moeten worden gemaakt bij het converteren van tekst. In dit geval worden de 1000 meest voorkomende woorden opgenomen in de vectoren.

**Token\_Patern** is een parameter om het patroon te specificeren dat wordt gebruikt om woorden of tokens in de tekst te identificeren. Het bepaalt welke sequenties van woorden worden gebruikt en welke woorden genegeerd. Token\_pattern=r'\b[^\d\W][^\d\W]+\b' zoekt naar opeenvolgende tekstreeksen die geen cijfers en speciale tekens bevatten, waardoor alleen alfabetische tekens worden opgenomen en andere tekens genegeerd.

## 3.3 Model planning

Nu dat het data cleanen klaar is ga ik verder naar het model building onderdeel. In het model building onderdeel leg ik uit welke modellen ik heb gekozen, waarom ik deze modellen heb gekozen, wat de modellen inhouden en hoe ik ze heb toegepast.



simplilearn.com

3.3.1 Wat zijn machine learning modellen?  
Machine learning modellen zijn algoritmen die worden gebruikt om computers te trainen om te leren van gegevens en voorspellingen te doen zonder expliciet geprogrammeerd te zijn. Ze maken gebruik van wiskundige en statistische technieken om patronen en relaties in de gegevens te ontdekken, zodat ze instaat zijn om nieuwe ongeziene gegevens te analyseren en voorspellingen te doen.

Het doel van machine learning modellen is om inzichten en voorspellingen te genereren op basis van complexe gegevens, die anders moeilijk handmatig te modelleren zouden zijn. Ze worden veel gebruikt in verschillende sectoren.

Bij het bouwen van machine learning modellen worden gegevens verdeeld in een training set om het model te trainen en een testset om de prestaties ervan te evalueren.

Ik heb online gezocht naar de meest gebruikte machine learning modellen en welke het meest nuttig zijn voor hotel reviews. Ik heb uiteindelijk gekozen voor logistic regression, naïve bayes en Random forest omdat ik daar veel voorbeelden van kon vinden. Verder leg ik ook de voor- en nadelen uit van deze modellen.

**Wat is underfitting?**Underfitting is een scenario waarbij een model niet in staat is om de onderliggende patronen en relaties in de trainingsdata vast te leggen. Het treedt op wanneer een model te eenvoudig is of niet de benodigde complexiteit heeft om de data goed weer te gegeven.  
Het gebeurt meestal wanneer het model niet genoeg kan leren van de trainingsdata. De voorspellingen van het model kunnen te algemeen zijn en de complexiteit van de data niet goed vastleggen. Om dit te voorkomen moet de data gesplit worden in een training set en een test set.

**Wat is overfitting?**Overfitting is een scenario waarbij het model te specifiek is afgestemd op de trainingsdata en daarom slecht presteert op nieuwe data. Het model heeft te veel geleerd van de trainingsvoorbeelden en heeft moeite om algemene patronen en trends te herkennen. Overfitting treedt meestal op wanneer het model te complex is of te veel parameters heeft in vergelijking met de trainingsdata.

### 3.3.2 Random Forest.

**Wat is Random forest?**Het eerste model waar ik voor heb gekozen is Random Forest. Dit is een algoritme dat wordt gebruikt voor zowel classificatie als regressie taken. Het is een ensemblemethode die voorspellingen van meerdere beslissingsbomen combineert om nauwkeurige voorspellingen te doen. Bij Random forest wordt een verzameling beslissingsbomen gecreëerd, elk getraind op een subset van de trainingsdata en met behulp van een willekeurige subset van functies. De willekeurigheid helpt overpassing te verminderen en de generatie van het model te verbeteren.

Afbeelding met tekst, lijn, diagram, schermopname

Automatisch gegenereerde beschrijving

19 Wat is Random forest van amo-nl.com

**Waarom koos ik voor random Forest?**Het heeft een hoge nauwkeurigheid voor het maken van voorspellingen. Random Forest maakt ook gebruik van meerdere beslissingsbomen die onafhankelijk van elkaar worden getraind. Door de voorspellingen van meerdere bomen te combineren, kan Random Forest overfitting verminderen.

**Wat zijn de voordelen van Random Forest?**1. Het is robuust tegen overfitting vanwege de willekeurigheid die tijdens de training wordt geïntroduceerd.

2. Ze kunnen een groot aantal input features verwerken. Zowel categorische als numerieke variabelen.

3. Het kan een goede voorspellingsprestaties behouden bij ontbrekende gegevens.

**Wat zijn de nadelen van Random Forest?**1. Het trainen van Random Forest kan lang duren. Vooral bij grote datasets en als er veel bomen worden gebruikt.

2.Wanneer de dataset ongebalanceerd is kan Random Forest vooringenomenheid vertonen en de neiging hebben om de meerderheidsklasse beter te voorspellen dan de minderheidsklasse.

**Wat zijn de gebruikte parameters?**De belangrijkste parameters die ik heb gebruikt zijn:   
n\_estimators: Dit zijn het aantal bomen.  
n\_jobs: Het aantal parallelle taken dat wordt uitgevoerd bij het bouwen van de bomen.  
Max\_depth: Dit verwijst naar de maximale diepte van de bomen. Dus tot hoever de bomen kunnen groeien.

Dit zijn de andere parameters die mogelijk gebruikt kunnen worden: bootstrap=True, class\_weight=None, criterion='gini', min\_impurity\_decrease=0.0, min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, n\_estimators=10, n\_jobs=3, oob\_score=False, random\_state=None, verbose=0, warm\_start=False

### 3.3.2 Multinominal Naïve Bayes

**Wat is Multinominal Naïve bayes?**  
Multinominal Naïve bayes is een algoritme dat veel wordt gebruikt om de classificatie van tekst te voorspellen. Het algoritme berekent de waarschijnlijkheid van elke mogelijke classificatie voor een gegeven tekst en kiest classificatie met de hoogste kans. Wat naïve bayes uniek maakt is dat het ervan uitgaat dat elk kenmerk in de classificatie onafhankelijk is van andere kenmerken. Met andere woorden, het algoritme behandelt elk kenmerk als een op zichzelf staand gegeven en negeert eventuele onderlinge afhankelijkheden tussen de kenmerken.

Het algoritme werkt door eerst het aantal keren dat elk woord voorkomt in het dataframe te tellen en de kenmerken te converteren naar directe tellingen. Vervolgens berekent de conditionele waarschijnlijkheden voor elke klasse op basis van de trainingsgegevens. Bij het voorspellen van de klasse van een nieuw datapunt wordt de bayesiaanse formule toegepast om de waarschijnlijkheid van elk datapunt voor elke klasse te berekenen en wordt de klasse met hoge waarschijnlijkheid toegewezen.

**Waarom koos ik voor Multinominal naïve bayes?**Omdat het een erg snel model is. Naïve bayes kan ook makkelijk worden geschaald naar grote datasets

**Wat zijn de voordelen?**

1. Het is eenvoudig en snel, vooral bij grote datasets vanwege de aanname van onafhankelijkheid.
2. Het algoritme werkt goed met een groot aantal kenmerken en kan goed omgaan met hoge dimensionale gegevens.

**Wat zijn de nadelen?**

1. De aanname van onafhankelijkheid tussen kenmerken kan beperkend zijn wanneer er wel afhankelijkheden tussen de kenmerken bestaan. Dit kan leiden tot minder nauwkeurigheid.
2. Als een bepaald kenmerk in de trainingsgegevens nooit voorkomt in een bepaalde klasse, zal de kansberekening resulteren in een nul waarde. Dit kan problematisch zijn, vooral bij kleine datasets.

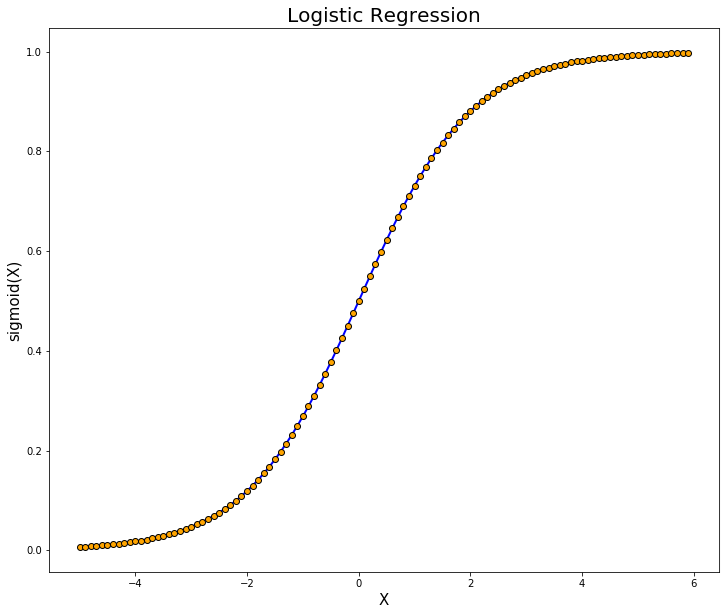
**Wat zijn de parameters?**

Geen, omdat de nauwkeurigheid zonder parameters mee te geven op 87% zit wat al redelijk goed is. Wanneer ik parameters zoals Alpha en fit\_prior meegaf, verlaagde de nauwkeurigheid naar 68%.

### 3.3.3 Logistic Regression

**Wat is Logistic Regression?**Logistic regression is een statistische analysemethode die wordt gebruikt om de relatie tussen afhankelijke variabelen of onafhankelijken variabelen te onderzoeken. Het wordt veel gebruikt voor classificatietaken, waarbij het doel is om gegeven observatie te delen in een of twee of meerdere categorieën.

Bij logistic Regression analyses kan je een voorspellend model maken om de kans op een positieve uitkomst van een categorische afhankelijke variabele te voorspellen. Dit kan met één of meerdere onafhankelijke variabelen. Hierbij is de categorische variabele vaak dichotoom maar het kan ook met een categorische variabele met meer dan twee categorieën. Hierbij voorspel je dus de kans op bijvoorbeeld een positieve uitkomst.



wat is logistic regresssion van towardsdatascience.com

**Waarom koos ik voor logistic regression?**Omdat het erg snel is vergeleken met andere modellen en omdat de nauwkeurigheid erg hoog is. Vooral bij grote datasets.

**Wat zijn de voordelen?**

1. Het trainen en toepassen van Logistic regression modellen is vaak snel, vooral bij grote datasets.
2. Het is minder gevoelig voor extreme waarden in vergelijking met andere classificatiemethoden zoals decision trees.

**Wat zijn de nadelen?**

1. Het kan alleen lineaire beslissingsgrenzen modelleren, wat betekent dat het mogelijk niet goed presteert bij complexe classificatie problemen waarbij de grens niet lineair is .
2. Als de klassen in de dataset erg ongebalanceerd zijn, kan logistic regression vooringenomen voorspellingen produceren en moeite hebben om minderheidsklassen correct te classificeren.

**Wat zijn de gebruikte parameters?**

Max\_iter: Het maximumaantal iteraties dat het mag uitvoeren om op de optimale coëfficiënt te komen.

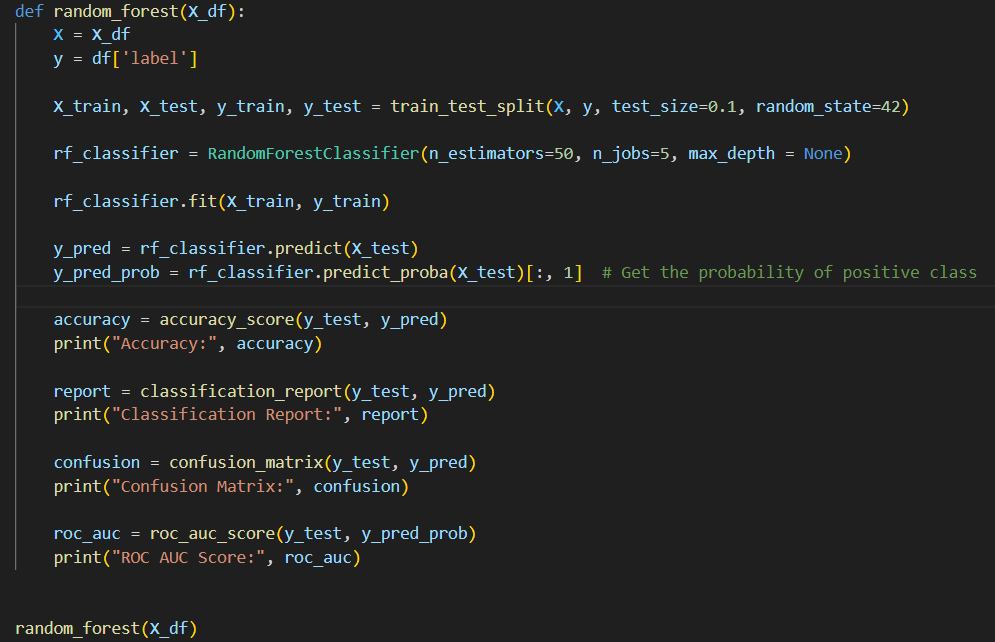
Random\_state: Dit wordt gebruikt om de random number generator te initialiseren, wat helpt bij het reproduceren van de resultaten.

Solver: Deze parameter specificeert het optimale-algoritme dat wordt gebruikt om de coëfficiënt van het logistic regression model te schatten.

## 3.4 Model building

In dit onderdeel laat ik zien hoe ik de modellen heb toegepast in mijn eigen code en leg ik uit waarom ik bepaalde parameters heb ingesteld.

### 3.4.1 Random Forest

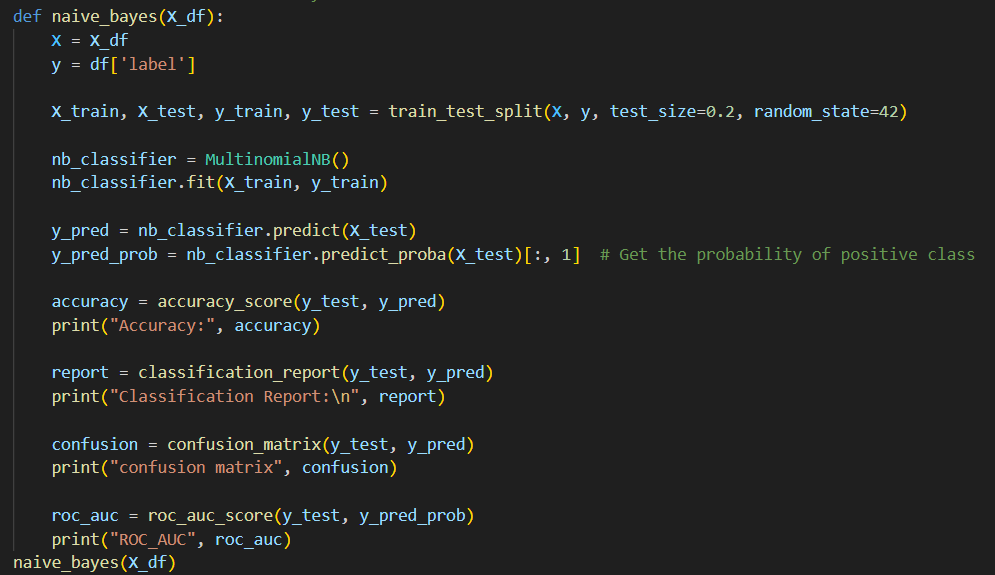


Ik heb de test data op 10% gezet omdat het om een erg grote dataset gaat en het anders enorm lang zou duren. De parameter n\_estimators staat op 50, Hiermee worden 50 bomen gemaakt. De reden dat ik het op 50 heb gezet laat ik verder in de resultaten zien met een plot. Het is hiermee accuraat genoeg en zorgt ervoor dat ik niet heel lang hoef te wachten.

N\_jobs staat op 5 om 5 parallelle processen uit te voeren. Max depth staat op None. Dit zorgt ervoor dat de bomen zo diep worden als nodig is om de gegevens volledig te modelleren.

Vervolgens zorg ik ervoor dat de accuraatheid, het classification report, het confusion matrix en Roc\_auc\_score worden geprint.

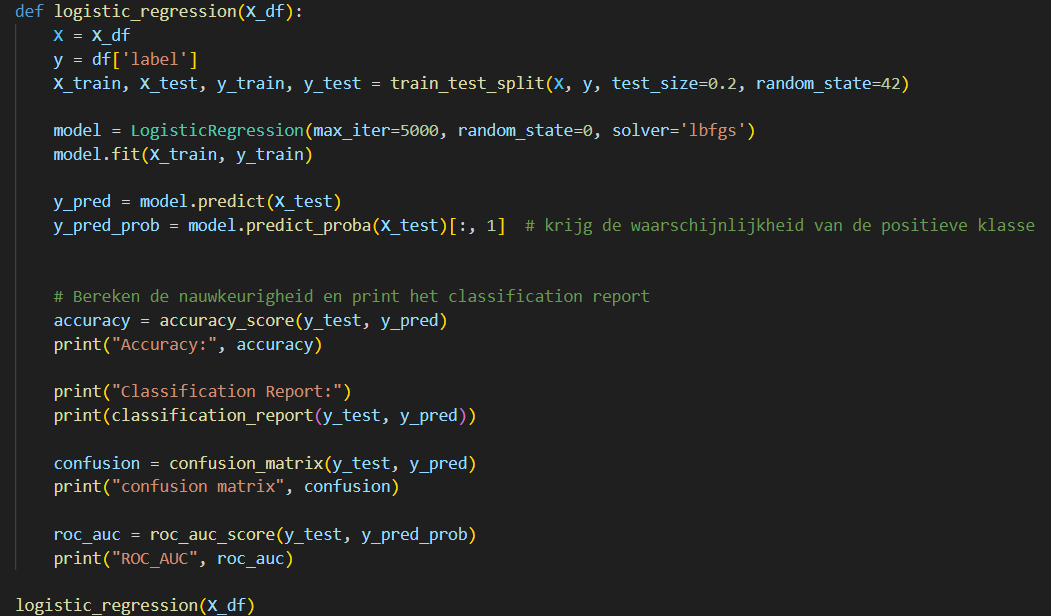
### 3.4.2 Multinominal Naïve Bayes



Ik heb bij naïve bayes de test grootte op 20% gelaten aangezien dit model een stuk sneller is dat Random Forest. Ik heb verder geen parameters toegevoegd. Dit is omdat de nauwkeurigheid al op 87% stond en wanneer ik parameters zoals: Alpha en Fit\_prior ging toevoegen de nauwkeurigheid omlaagging.

Ook hier print ik de accuracy\_score, Classification\_report, Confusion\_matrix en roc\_auc\_score

### 3.4.3 Logisitic regression



Ik heb net als bij Multinominal Naïve bayes de testgrootte op 20% gezet. Het maximumaantal iteraties staat op 5000.

# 4.Resultaten

In dit onderdeel laat ik de resultaten zien wat de scores zijn en wat ze nou precies betekenen.

## 4.1 wat waren de scores.

### 4.1.1 Accuracy

De accuracy(nauwkeurigheid) Geeft aan hoe goed het model presteert in termen van correct classificeren van de voorbeelden. In andere woorden de verhouding van het aantal correct voorspelde reviews ten opzichte van het totaal. Bijvoorbeeld een accuracy van 95% betekent dat het model 95% van de reviews correct heeft voorspeld. Het is belangrijk dat er niet alleen naar de accuracy wordt gekeken, vooral wanneer de dataset ongebalanceerd is. Als er bijvoorbeeld een meerderheid aan positieve reviews is kan het model een hoge nauwkeurigheid krijgen door de meerderheid te voorspellen

Nauwkeurigheid van Random Forest:  


Nauwkeurigheid van Multinominal Naive\_bayes:  

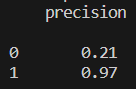

Nauwkeurigheid van Logistic Regression:  

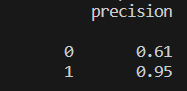

### 4.1.2 Precision

Bij Precision geldt dat TP het aantal terecht positieven is en FP het aantal fout positieven. 0 staat hier voor de negatieve reviews en 1 staat voor de positieve reviews. Als het percentage van negatief op 21% staat betekent het dat de reviews die door het model als negatief zijn voorspeld, slechts 21% negatieve reviews waren.

Precision Random forest:



Precision Multinominal Naïve Bayes:  


Precision Logistic regression:  


### 4.1.3 Confusion matrix

Een confusion matrix, ook wel bekend als een error matrix, is een tabel die wordt gebruikt om de prestaties van een classificatiemodel te evalueren. Het geeft inzicht in de classificatieresultaten door de werkelijke klassen te vergelijken met de voorspelde klassen.

Werkelijke Klasse 0 True Negative (TN) False Positive (FP) Werkelijke Klasse 1 False Negative (FN) True Positive (TP)

Elke cel in de matrix vertegenwoordigt een combinatie van de werkelijke klasse en de voorspelde klasse.

True Positive (TP): Het aantal reviews dat correct is voorspeld als positief (label 1) door het model.

True Negative (TN): Het aantal reviews dat correct is voorspeld als negatief (label 0) door het model.

False Positive (FP): Het aantal reviews dat fout is voorspeld als positief (label 1) door het model, terwijl het werkelijk negatief was.

False Negative (FN): Het aantal reviews dat fout is voorspeld als negatief (label 0) door het model, terwijl het werkelijk positief was.

Confusion matrix: Random forest:  


Confusion matrix: Multinominal naïve bayes:  
  
Confusion matrix: logistic regression:   


Ik neem als voorbeeld random forest model

417 waren negatief en correct als negatief voorspeld.

3306 waren negatief en zijn als positief voorspeld.

520 waren positief maar zijn als negatief voorspeld.

63877 waren positief en zijn als positief voorspeld.

### 4.1.4 Roc\_auc\_score

De ROC AUC-score staat voor de Receiver Operating Characteristic Area Under the Curve-score. Het is een evaluatiemaatstaf die wordt gebruikt om de prestaties van een classificatiemodel te meten, met name bij het werken met ongebalanceerde datasets.

De ROC-curve is een grafische weergave van de relatie tussen de True Positive Rate en de False Positive Rate bij verschillende classificatiedrempelwaarden. De AUC (Area Under the Curve) van de ROC-curve geeft de mate van onderscheidend vermogen van het model aan. Hoe groter de AUC, hoe beter het model in staat is om onderscheid te maken tussen positieve en negatieve voorbeelden.

Random fores:  

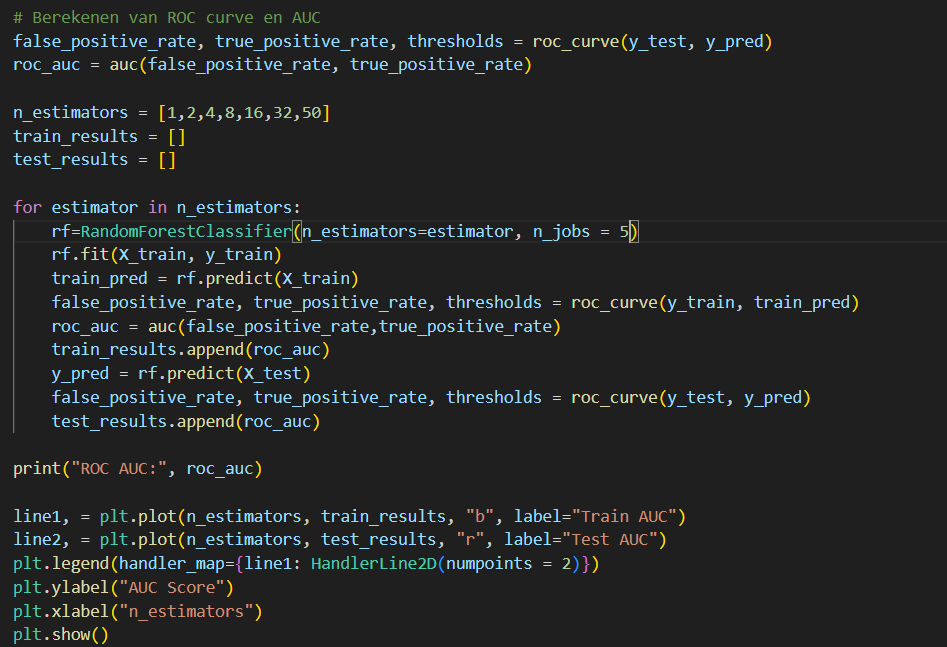

Multinominal naïve bayes:



Logistic regression:



Ik heb een plot gemaakt om te zien hoe de Roc\_auc score eruitziet bij het aantal gebruikte n\_estimators.



Afbeelding met schermopname, tekst, lijn, Perceel

Automatisch gegenereerde beschrijving

Hierboven de plot die weergeeft wat de score is bij het aantal estimators voor Random Forest.

# 5.Conclusie

Ik ben door alle stappen van de Big data opdracht heen gegaan. In het data discovery deel heb ik de dataset verzameld door de kaggle set te combineren met de webscraped reviews en de zelfgeschreven reviews en hoe ik de dataset heb gelabeld.

In het data cleaning onderdeel liet ik zien hoe ik de dataset uit de database haalde en de rows eruit heb gehaald die niet nuttig zijn voor de modellen. Verder heb ik ook stemming, lemmatisering en vectorisering toegepast.

Tot slot heb ik de machine learning modellen ingesteld en een aantal voorspellingen gedaan op de hotel reviews. Het is te zien dat de accuraatheid van alle drie gebruikte modellen hoog is.

RandomForest heeft een accuraatheid van:94%

Naïve Bayes heeft een accuraatheid van: 87%

Logistic regression heeft een accuraatheid van 94,7%

Het is ook te zien dat Logistic regression het beste werkt van de 3 modellen.

Referentie lijst

*User Guide — pandas 2.0.2 documentation*. (z.d.). <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>

*4. Locating Elements — Selenium Python Bindings 2 documentation*. (z.d.). <https://selenium-python.readthedocs.io/locating-elements.html>

Stemming vs Lemmatization in Python. (2022). *www.turing.com*. <https://www.turing.com/kb/stemming-vs-lemmatization-in-python>

GeeksforGeeks. (2019). Vectorization in Python. *GeeksforGeeks*. <https://www.geeksforgeeks.org/vectorization-in-python/>

Chugh, V. (z.d.). *Converting Text Documents to Token Counts with CountVectorizer - KDnuggets*. KDnuggets. <https://www.kdnuggets.com/2022/10/converting-text-documents-token-counts-countvectorizer.html>

*Figure 9. Count Vectorization.* (z.d.). ResearchGate. <https://www.researchgate.net/figure/Count-Vectorization_fig6_339997426>

*sklearn.feature\_extraction.text.TfidfVectorizer*. (z.d.). scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>

V, N. (2023). What Are N-Grams and How to Implement Them in Python? *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/09/what-are-n-grams-and-how-to-implement-them-in-python/>

Great Learning Team. (2023, 4 mei). *Top 5 Most Used Machine Learning Algorithms in Python | Great Learning*. Great Learning Blog: Free Resources what Matters to shape your Career! <https://www.mygreatlearning.com/blog/most-used-machine-learning-algorithms-in-python/>

Tavasoli, S. (2023). Top 10 Machine Learning Algorithms For Beginners: Supervised, and More. *Simplilearn.com*. <https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article>

AMO. (2022, 4 februari). *Het principe van de werking van Random Forest | AMO-PB Icastat*. AMO-PB Icastat | advies- en ingenieursbureau modellering, optimalisatie en statistiek. <https://www.amo-nl.com/het-principe-van-de-werking-van-random-forest/>

Great Learning Team. (2023b, juni 13). *Random forest Algorithm in Machine learning | Great Learning*. Great Learning Blog: Free Resources what Matters to shape your Career! <https://www.mygreatlearning.com/blog/random-forest-algorithm/>

SPSS Handboek. (2021, 22 juli). *Wat is de logistische regressie (of logistic regression)? - SPSShandboek*. SPSShandboek. <https://spsshandboek.nl/logistic-regression/#:~:text=Met%20Logistische%20Regressie%20(Logistic%20regression,%C3%A9%C3%A9n%20of%20meerdere%20onafhankelijke%20variabelen>.

Thorn, J. (2022, 30 augustus). Logistic Regression Explained - Towards Data Science. *Medium*. <https://towardsdatascience.com/logistic-regression-explained-9ee73cede081>

*Minerva*. (z.d.). Minerva Website. <http://www.minerva-ebm.be/Results/Glossary/1685>

Narkhede, S. (2021, 15 juni). Understanding Confusion Matrix - Towards Data Science. *Medium*. <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

*sklearn.metrics.precision\_score*. (z.d.). scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html#:~:text=The%20precision%20is%20the%20ratio,more%20in%20the%20User%20Guide>.

*sklearn.ensemble.RandomForestClassifier*. (z.d.). scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

*1.9. Naive Bayes*. (z.d.). scikit-learn. <https://scikit-learn.org/stable/modules/naive_bayes.html>

*sklearn.linear\_model.LogisticRegression*. (z.d.). scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

*sklearn.feature\_extraction.text.CountVectorizer*. (z.d.). scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.CountVectorizer.html