ARCADA

Effektiv extrahering av ordpar från kundrecensioner

En webbskrapnings- och NLP-baserad metod för sentimentanalys

Toivo Seppä

Lärdomsprov
Informationsteknik
2025

Lärdomsprov

Toivo Seppä

Effektiv extrahering av ordpar från kundrecensioner. En webbskrapnings- och NLP-baserad metod för sentimentanalys.

Yrkeshögskolan Arcada: Informationsteknik, 2025.

Sammandrag:

Detta lärdomsprov undersöker hur man effektivt kan extrahera ofta förekommande ordpar (bigrams) från kundrecensioner på nätet genom en kombination av webbskrapning och Natural Language Processing (NLP). Syftet var att utveckla ett program som skrapar recensioner från Trustpilot, lagrar dem i en strukturerad datatabell och bearbetar dem med NLP-tekniker för att identifiera de vanligaste ordparen i positiva och negativa recensioner. Arbetet avgränsades till att använda endast en webbplats och huvudsakligen engelskspråkiga recensioner, för att begränsa komplexiteten.

Materialet bestod av cirka 5000 recensioner från två företag, EasyJet och FlixBus. Webbskrapningen utfördes med Python-biblioteket BeautifulSoup och datan organiserades i en Pandas-tabell. NLP-bearbetningen genomfördes med hjälp av NLTK och inkluderade textnormalisering, tokenisering, borttagning av stop words, lemmatisering samt extrahering av bigrams.

Resultaten visade att programmet effektivt kunde skrapa och analysera recensioner och identifiera frekventa ordpar som "customer service" och "flight delayed". EasyJets recensioner var till största delen negativa, medan FlixBus hade en större andel positiva recensioner. Begränsningar i projektet inkluderade ett snävt urval av källor och språk, vilket påverkade bredden av extraherade data.

De extraherade ordparen kan användas för att förbättra kundservice genom att ge insikter i vanliga kundproblem och som underlag för träning av kundtjänst-chattbotar. Arbetet visar att kombinerad användning av webbskrapning och NLP är ett effektivt sätt för att analysera stora mängder kundfeedback på nätet.

Nyckelord:

webbskrapning, NLP, kundrecensioner, bigrams, sentimentanalys, Python

Degree Thesis

Toivo Seppä
Efficient Extraction of Word Pairs from Customer Reviews – A Web Scraping and NLP-Based Method for Sentiment Analysis.
Arcada University of Applied Sciences: Information Technology, 2025.

Abstract:

This thesis investigates how to effectively extract frequently occurring word pairs (bigrams) from online customer reviews through a combination of web scraping and Natural Language Processing (NLP). The aim was to develop a program that scrapes reviews from Trustpilot, organizes them into a structured data table, and processes them with NLP techniques to identify the most common word pairs in positive and negative reviews. The study was limited to a single review platform and primarily English-language reviews to minimize complexity.

The material consisted of approximately 5000 reviews from two companies, EasyJet and FlixBus. Web scraping was performed using the Python library BeautifulSoup, and the data was organized using Pandas. NLP processing was conducted with NLTK and included text normalization, tokenization, stop word removal, lemmatization, and bigram extraction.

The results showed that the software effectively scraped and analyzed reviews, identifying frequent word pairs such as "customer service" and "flight delayed." EasyJet's reviews were predominantly negative, whereas FlixBus had a higher proportion of positive reviews. Project limitations included a narrow selection of sources and languages, affecting the breadth of extracted data.

The extracted word pairs can be used to improve customer service by providing insights into common customer issues and can serve as training material for customer service chatbots. The work demonstrates that combining web scraping and NLP is an effective approach for analyzing large volumes of online customer feedback.

Keywords:

web scraping, NLP, customer reviews, bigrams, sentiment analysis, Python

Innehåll

Figurer

1 Inledning

Arbetet utreder hur man kan plocka ut nyckelord som förekommer ofta i kundrecensioner. Målet med arbetet är att skapa ett program som skrapar kundrecensioner från en recensionsplattform. Skrapade recensionerna bearbetas sedan med NLP. Programmet ska kunna visa vilka ordpar förekommer oftast i positiva och negativa recensioner. Dessa ordpar ska sedan presenteras som data som kan användas för att få en inblick i kundsentiment och för att träna en kundbetjänings-chatbot.

Förväntade resultatet för programvaran är att den kan skrapa data från olika webbsidor, spara dessa data i en tabell, bearbeta den med NLP tekniker och visa vilka ordpar förekommer oftast i positiva och negativa recensioner. Ordparen kommer då att vara data som ett företag potentiellt kunde använda för att få bättre insikt i deras kundsentiment eller för att träna en chatbot för att kunna bättre hantera relevanta problem. Arbetet ska ge en bättre förståelse om webbskrapning och NLP och dess användbarhet när det kommer till att hitta värdefulla data som kan användas vidare i kundbetjäning.

Detta examensarbete är av intresse för företag som strävar efter att använda chatbottar för att hantera kundförfrågningar effektivt samtidigt som de tar itu med problem som lyfts fram i kundrecensioner. Arbetet är också till nytta för personer som vill undersöka hur man kan effektivt hitta information med NLP-tekniker från webbskrapade data.

1.1 Syfte

Syftet med arbetet är att utveckla ett program som med hjälp av webbskrapning och NLP-tekniker kan identifiera och extrahera ofta förekommande ordpar (bigram) från kundrecensioner. Genom att analysera frekventa ordpar i positiva respektive negativa recensioner kan företag få insikter om vilka specifika faktorer kunder uppskattar eller är missnöjda med. Dessa ordpar är särskilt värdefulla eftersom de tydligt visar vanligt förekommande kundproblem och teman som en kundtjänst-chatbot skulle kunna tränas att identifiera och hantera. På så sätt kan chatboten ge snabbare och mer relevanta svar, vilket förbättrar kundupplevelsen.

Min forskningsfråga är:

- Hur kan man effektivt hitta ofta förekommande ordpar från kundrecensioner på webben?

1.2 Avgränsning

Detta lärdomsprov kommer att fokusera på webbskrapning och NLP, men eftersom dessa processer har många olika verktyg och tekniker kommer endast vissa av dem användas. För webbskrapning var API:er ett alternativ. På grund av att API:er ofta kostar att använda och de begränsar vilken data man kan få, har jag valt att detta projekt utförs med webbskrapning.

Eftersom Python är ett programmeringsspråk med mycket användning i både webbskrapning och NLP-utveckling, kommer det att användas i detta projekt. För att minimera komplexiteten kommer endast ett verktyg användas för webbskrapning och NLP. På grund av detta kommer endast recensioner på ett språk användas i slutliga NLP-utvecklingen, eftersom användning av olika språk skulle betyda behovet av flera olika verktyg. Endast en recensionsplattform, Trustpilot, kommer att skrapas från. Detta är också för att minimera komplexiteten av projektet.

För webbskrapning kommer Python-biblioteket BeautifulSoup användas. BeatifulSoup lämpar sig väl för enkel webbskrapning och lämpar sig väl för detta projekt. NLP-verktyget måste väljas på basis av vilket språk som har mest recensioner, eftersom olika verktyg fungerar olika bra beroende på språk. Troligtvis kommer majoriteten av recensionerna att vara på engelska, i vilket fall NLTK-verktyget kommer att användas.

Även om ordparen som kommer att extraheras kan användas för chatbot-utveckling, kommer själva chatbot-utvecklingen inte att ske i detta arbete. Arbetet fokuserar på sökningen av ordparen.

1.3 Metodik

Programvaran som utvecklas kommer att bestå av två delar. Första delen är en webbskrapare och andra delen består av NLP-bearbetning. Webbskraparen kommer att skrapa ner webbrecensioner från Trustpilot webbsidan, varefter de lagras i en tabell för att lättare kunna bearbetas. Efter det används NLP-tekniker för att bearbeta recensionerna för att kunna extrahera ordparen från recensionerna.

1.4 Definitioner

- Webbskrapning = En process för att hämta stora mängder data från en eller flera webbsidor

- HTML (Hypertext Markup Language) = Standard märkspråk för webbutveckling.

- BeautifulSoup = Ett Python-bibliotek för webbskrapning.

- Pandas = Ett Python-bibliotek för databearbetning. Möjliggör skapandet av tabeller och förenklar organisering av data.

- NLP (Natural Language Processing) = Ett område inom artificiell intelligens som handlar om att få datorer att förstå människospråk.

- NLTK (Natural Language Toolkit) = Ett Python-bibliotek för NLP-utveckling.

- Bigrams = På varandra följande ord (ordpar).

2 Teori

I detta kapitel kommer webbskrapning och NLP att förklaras. De olika verktygen och tillämpningarna för båda processerna kommer att gås igenom.

2.1 Webbskrapning

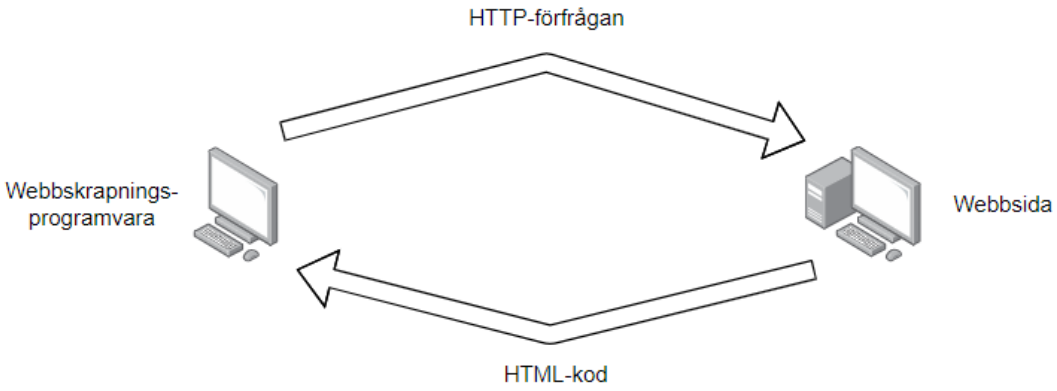
Webbskrapning är en teknik för att konvertera ostrukturerade webbdata till strukturerade data som kan sparas och analyseras i ett kalkylblad eller en databas. Webbskrapning möjliggör hämtning av stora mängder data i en kort tid, vilket är en stor fördel i dagens datadrivna värld. Webbskrapning uppnår samma resultat som manuell textkopiering från webbsidor, men det försnabbar och automatiserar processen. (Khder, 2021)

2.1.1 Webbstruktur och HTTP-förfrågan

För att webbskrapa data krävs en förståelse för hur webbsidor är strukturerade. Webbsidor byggs upp med HTML (Hypertext Markup Language), ett standardiserat märkspråk som definierar sidans innehåll och layout med hjälp av olika element och attribut, såsom tags och klasser. Dessa element används för att strukturera och identifiera innehåll på sidan, vilket är avgörande när man skrapar data automatiskt. (W3Schools, u.å.-a)

För att hämta data från webbsidor skickas en HTTP-förfrågan (Hypertext Transfer Protocol). HTTP är internetets protokoll för kommunikation mellan klienter och servrar. I webbskrapning används främst HTTP-metoden GET, där webbskraparen begär HTML-innehållet från webbsidan. Webbsidan returnerar sedan sin HTML-kod, vilken webbskraparen analyserar för att extrahera önskad information med hjälp av olika programmeringsverktyg, till exempel Python-biblioteken Requests och BeautifulSoup. (Cloudflare, u.å.)

Denna process möjliggör snabb extrahering av strukturerade data som sedan kan sparas och analyseras vidare. Figur 1 visar hur informationen mellan ett webbskrapningsprogram och en webbsida fungerar. Webbskrapningsprogrammet skickar en HTTP GET-förfrågan till webbsidan och webbsidan skickar en fil som innehåller webbsidans HTML-kod tillbaka.



Figur 1 Överföring av information från webbsida till webbskrapare

2.1.2 API (Application programming interface)

Webbskrapning var det enda sättet för ett program att erhålla webbdata tills uppkomsten av API:er. API:er är specialiserade gränssnitt för att underlätta kommunikation mellan applikationer och servrar. API:er kan förse data i ett organiserat sätt i många olika datatyper. Även om en del webbsidor erbjuder API:er så är det ofta begränsade i den datamängd som de har och ibland kan de ha en användningsavgift. Därför kan man inte alltid vara beroende av API:er och i stället använda webbskrapning för att säkerställa de data som vill fås. Khder (2021)

2.1.3 Tillämpningar av webbskrapning

Webbskrapning har ett brett utbud av tillämpningar inom många branscher. Här är några populära användningsområden:

- **Prisjämförelse**: Extrahering av prisinformation från olika nätbutiker för att jämföra priser och på den basen välja bästa erbjudandet. (WebHarvy, u.å.)

- **Konkurrensövervakning**: Företag som erbjuder produkter eller tjänster kan få omfattande data om konkurrenter som erbjuder liknande produkter och tjänster. (WebHarvy, u.å.)

- **Aktiemarknadsanalys**: Analysering av data som skrapats från relevanta aktiewebbsidor för att göra informerade beslut på aktiemarknaden. (Heath, 2023)

- **Träningsdata för maskininlärningsmodeller**: Maskininlärningsalgoritmer kräver stora mängder data för att tränas. Detta kan förvärvas från webbsidor som är relevanta för modellen, till exempel recensionsplattformar eller fastighetsannonser. (WebHarvy, u.å.)

- **Kundsentiment**: Skrapning av en stor mängd kundrecensioner kan ge en bättre insikt i ett företags kundsentiment. (Heath, 2023)

2.1.4 Webbskrapningsverktyg

Det finns många verktyg och moduler tillgängliga för webbskrapning, från enkel dataextraktion till komplex automatisering. Här är några av de mest populära verktygen och ramverken:

- **BeautifulSoup:** BeautifulSoup är ett Python-programbibliotek som underlättar bearbetning av HTML-filer. BeautifulSoup skapar ett parsningsträd, vilket underlättar sökning av data från en webbsidas HTML-kod. Lämpar sig väl för mindre projekt med enkel webbskrapning. (Crummy, u.å.)

- **Scrapy:** Scrapy är ett kraftfullt Python-ramverk för storskaliga webbskrapningsprojekt. Den hanterar genomsökning, extrahering och export med lätthet. Scrapy är mycket effektivt för komplexa skrapningsuppgifter som involverar flera sidor och webbplatser. (Johansson, 2023)

- **Selenium:** Selenium används för att skrapa dynamiska webbplatser genom att interagera med webbsidor som en riktig användare skulle göra. Det fungerar bra med JavaScript-tunga webbplatser men är långsammare jämfört med andra skrapverktyg. (Johansson, 2023)

2.1.5 Etiska frågor kring webbskrapning

Etiken kring webbskrapning är ett komplext och brett omdebatterat ämne. Det handlar om att balansera fördelarna med datatillgänglighet med respekt för integritet, ägande och avsikten hos informationens ursprungliga värd.

En utmaning är att fastställa om webbskrapning bryter mot upphovsrätten. Rakovic (2020) beskriver att det kan vara svårt att avgöra vilken typ av data som skrapas och om det rör sig om skyddat material. Dessutom kan det vara problematiskt att bevisa att ett intrång skett, särskilt om informationen redan är publik. Om webbskrapning omfattar personuppgifter kan det leda till integritetsproblem, särskilt enligt GDPR. Många

webbplatser har vidtagit åtgärder för att minska risken för att personuppgifter skrapas, men det är fortfarande en riskfaktor.

2.2 NLP (Natural Language Processing)

NLP, eller naturlig språkbehandling, är ett område inom artificiell intelligens som handlar om att få datorer att förstå, analysera och bearbeta mänskligt språk, både tal och text. Enligt Malik et al. (2022) bär vår kommunikation, vare sig den sker genom prat, skrivande eller till exempel sociala medier som tweets, på massor av information. Denna information är ofta ostrukturerad och svår att analysera med vanliga databaser, men med hjälp av modern teknik som maskininlärning har stora framsteg gjorts.

NLP gör det möjligt för datorer att inte bara känna igen ord utan också förstå innebörden, tonen, avsikten och känslan bakom det som sägs. Det används bland annat i maskinöversättning, röststyrning, textsammanfattning, söksystem och inom medicin och affärsanalys. (Malik et al., 2022)

NLP fungerar genom att först förbehandla och analysera text med språkliga verktyg och sedan använda algoritmer för att få datorer att förstå textens innebörd och struktur. NLP är en kombination av lingvistik, AI och dataanalys, och gör att vi idag har många smarta tjänster och applikationer som effektivt kan kommunicera och interagera med människor på ett naturligt sätt. (Malik et al., 2022)

2.2.1 NLP tekniker

NLP omfattar flera tekniker som gör det möjligt för datorer att behandla och förstå människospråk. Det finns massvis med olika tekniker för språkbehandling och därför listas här endast tekniker som kommer vara kritiska för detta projekt:

- **Tokenisering**: Tokenisering är uppdelningen av text till mindre delar. Detta kan vara uppdelning av ett stycke till meningar eller uppdelning av meningar till ord. Att bryta upp ord gör det lättare för datorer att bearbeta och analysera texten. (GeeksforGeeks, 2025)

- **Lemmatisering**: Reducerar ord till sin basform. Lemmatisering av ord gör för en mera förenlig representation. (”hoppar”, ”hoppade” -> ”hoppa”) (GeeksforGeeks, 2025)

- **Stop word removal**: Tar bort vanliga ord som inte har större betydelse (och, är, att). Detta gör att datorn kan fokusera på det meningsfulla innehållet i texten. (GeeksforGeeks, 2025)

- **Text normalization**: Gör texten standardiserad och enhetlig. Detta innebär att alla bokstäver görs små, skiljetecken tas bort och skrivfel korrigeras. (GeeksforGeeks, 2025)

- **N-grams:** N-grams är på varandra följande ord i en text. N-grams tillåter användaren att se vilka ord som förekommer efter varandra. Detta är användbart i till exempel sökmotorer där man kan förutspå sökningen baserat på tidigare ord. N-grams ger också mera kontext än ett ord för sig själv ger. Bigrams, d.v.s. ordpar, är exempel på n-grams. (GeeksforGeeks, 2024)

- **Korpus:** En korpus är ett dataset av text som kan användas för att träna ett maskininlärningssystem. En korpus kan bestå av nyhetsartiklar, recept, noveller, eller andra texter som är relevanta för dess användningsområde. I denna undersökning kommer korpusen bestå av webbrecensioner. (Subex 2023)

2.2.2 Tillämpningar av NLP

NLP har flera tillämpningar som kan hjälpa med olika textbaserade uppgifter. Här är en lista på olika tillämpningar av NLP:

- **Informationssökning:** Informationssökning hjälper datorer att hitta rätt information från stora datamängder. Sökmotorer, chatbottar och rekommendationssystem använder sig av informationssökning för att ge användaren relevanta svar. (Rawat, 2023)

- **Kundsentiment:** Avgöra om en text uttrycker positivt, negativt eller neutralt innehåll. (Patwardhan et al., 2023)

- **Språkmodellering:** Förutsäga nästa ord i en mening. Detta kan användas i sökmotorer som Google, för att göra sökningar snabbare. (Patwardhan et al., 2023)

- **Maskinöversättning:** Översätta mellan språk. (Patwardhan et al., 2023)

- **Textgenerering:** Skapa text utifrån en prompt. Detta är användbart i till exempel chatbottar, som kundservice chattar eller ChatGPT. (Patwardhan et al., 2023)

- **Textklassificering:** Sortera texter i kategorier, till exempel nyhetsartiklar eller kundrecensioner. (Patwardhan et al., 2023)

2.2.3 Verktyg för NLP

Det finns flera NLP verktyg för olika användningsområden. Här är några verktyg för NLP:

- **NLTK (Natural Language Toolkit)**: Ett omfattande Python-bibliotek för NLP, särskilt bra för undervisning och prototypande. Stöder klassificering, tokenisering, parsing och mycket mer. (Malik et al., 2022)

- **spaCy**: Ett snabbt och produktion fokuserat Python-bibliotek för NLP. Har stöd för djupinlärning och används ofta i praktiska applikationer. Har stöd för många språk och moderna modeller. (Malik et al., 2022)

- **Apache OpenNLP**: Ett maskininlärningsbaserat bibliotek i Java för NLP-uppgifter som tokenisering, meningssegmentering, POS-taggning, namnigenkänning och parsing. Det används både i forskning och industri. (Malik et al., 2022)

- **ChatScript**: Ett ramverk för att skapa chatbots med reglerbaserad logik. Det håller koll på kontext mellan inmatningar och svar, och gör det möjligt att bygga interaktiva konversationer med hjälp av egna skript. (Malik et al., 2022)

2.2.4 Vikten av ordpar (bigram) för chatbot-utveckling

För att en chatbot effektivt ska kunna hantera vanliga kundproblem behöver den förstå vad kunden faktiskt menar, även om olika kunder uttrycker sig på många olika sätt. En viktig del i detta är att chatboten kan identifiera kundens avsikt eller syfte med en fråga eller ett påstående. Enligt Lacasa et al. (2024) består en typisk chatbot av tre huvudkomponenter: Natural Language Understanding (NLU), Dialog Management (DM) och Natural Language Generation (NLG). Det är just NLU som har uppgiften att förstå användarens meddelanden och omvandla dem till tydliga handlingar.

När en chatbot analyserar kunders meddelanden måste den identifiera särskilt viktiga ord eller fraser som tydligt indikerar kundens problem eller avsikter. Bigram, ordpar av två ord som följer varandra är särskilt viktiga eftersom de ger mer specifik information än enskilda ord. Till exempel kan ordparet ”dålig service” tydligare identifiera kundens missnöje än enbart ordet ”dålig” eller ”service” separat. (Sapardic, 2025)

Genom att extrahera och analysera frekventa ordpar från kundrecensioner kan chatboten effektivare förstå vad kunderna är missnöjda med eller har frågor om, och därmed snabbare ge relevanta svar. På detta sätt hjälper bigram chatboten att ge bättre kundservice genom att förstå och hantera kundens behov direkt.

3 Metod

I detta kapitel kommer utvecklandet av programvaran förklaras. Programvaran är utvecklad för att analysera kundsentiment från webbskrapade recensioner. För webbskrapningen används BeautifulSoup-biblioteket för Python. För varje webbskrapade recension sparas dess titel, brödtext, gradering, plats och datum. Alla recensionerna sparas sedan i en Pandas-tabell. Recensionerna klassificeras som positiva eller negativa baserat på deras förutbestämda gradering. För att simplifiera arbetet kommer bearbetningen endast göras på ett språk. För detta kommer Python-modulen langdetect användas för att hitta vilket språk förekommer oftast i recensionerna. NLP teknikerna som sedan implementeras inkluderar tokenisering, lemmatisering, stop word removal, text normalization och n-gram-analys. Huvudmålet är att extrahera de mest förekommande ordparen (bigrams) från både positiva och negativa recensioner, vilka tyder på återkommande teman eller känslor i recensionerna. Även om omfattningen av detta lärdomsprov inte inkluderar träning av en chatbot, är de extraherade bigrammen avsedda för hypotetisk användning i utvecklingen av en chatbot för kundtjänst. Specifikt kan dessa bigram hjälpa chatboten att hantera vanliga kundproblem effektivt.

3.1 Webbskrapning

Programvarans första del fokuserar på webbskrapning. I detta skede hämtas recensioner från en recensionsplattform och organiseras i en tabell för vidare bearbetning. Processen omfattar flera steg: först skickas en HTTP-förfrågan till webbplatsen för att hämta HTML-koden. Därefter skapas ett soup-objekt med hjälp av BeautifulSoup, vilket gör HTML-koden mer lättåtkomlig för dataextrahering. I nästa steg extraheras relevanta data från soup-objektet, inklusive recensionernas titlar, brödtexter, gradering, plats och datum. Slutligen organiseras dessa data i en Pandas-tabell, vilket underlättar den efterföljande NLP-bearbetningen.

3.1.1 Steg 1: HTTP förfrågan

För att webbskrapa data måste först en HTTP-förfrågan skickas till den webbsida som innehåller datan. I detta arbete används Python-biblioteket requests, vilket gör det enkelt att skicka och hantera HTTP-förfrågningar (PyPI, 2024). Genom att använda *get()*-funktionen skickas en så kallad GET-förfrågan till sidans webbadress (W3Schools, u.å.-b). Svaret på förfrågan är ett *response*-objekt som innehåller bland annat sidans HTML-kod (W3Schools, u.å.-c). HTML-koden innehåller all information som visas på sidan och är grunden för att kunna extrahera specifika data.

För att effektivt hämta en större mängd recensioner körs HTTP-förfrågningen i en loop, där varje sidnummer automatiskt justeras i webbadressen. På så sätt samlas HTML-innehåll från flera webbsidor snabbt och strukturerat, redo för nästa steg i processen.

I figur 2 visas Python-koden som skickar en HTTP-förfrågan och sparar det returnerade *response*-objektet.

```
urlres = []
for x in range(500):
    res = requests.get("https://fi.trustpilot.com/review/flixbus.com?languages=all&page=" + str(x+1))
    urlres.append(res)
```

Figur 2 requests-biblioteket använder get()-funktionen för att hämta data från webbsidan

3.1.2 Steg 2: Soup-objekt

När HTML-koden har hämtats från webbsidan med hjälp av requests, är nästa steg att göra denna kod hanterbar. Det görs med hjälp av biblioteket BeautifulSoup, som är ett populärt verktyg i Python för att strukturera och tolka HTML-dokument. För att använda det importeras det med kommandot *from bs4 import BeautifulSoup*.

HTML-koden som hämtats är egentligen en lång textsträng, vilket gör det svårt att manuellt navigera bland dess olika delar. Därför används BeautifulSoup för att omvandla denna kod till ett så kallat *soup*-objekt. Ett *soup*-objekt är en strukturerad version av HTML-dokumentet, där alla taggar, klasser och element blir lättillgängliga i ett så kallat parse-träd. Detta träd gör det möjligt att söka igenom koden efter specifika delar, såsom rubriker, textstycken eller datum. (Sulcas, 2025)

Varje *response*-objekt som hämtats från en webbsida innehåller HTML-kod i sin *text*-variabel (W3Schools, u.å.-c). I figur 3 ser man hur *soup*-objektet skapas genom att BeautifulSoup tolkar *text*-variabeln av *respons*-objekten. Figur 4 är en skärmdump av början på *soup*-objektet, d.v.s. parse-trädet. Variabeln ser stökig ut, men den innehåller hela HTML-koden för webbsidan. Alla dessa *soup*-objekt sparas för varje sida, och de kommer att användas i nästa steg för att hämta ut den data vi är intresserade av, nämligen recensionerna.

```
soup = BeautifulSoup(urlres[0].text, 'html.parser')
```

Figur 3 BeautifulSoup parsar igenom HTML-koden

```
<!DOCTYPE html>
<html lang="fi-FI"><head><meta charset="utf-8"/><meta content="width=device-width, initial-scale=1" name="viewport"/><link hr
ef="https://cdn.trustpilot.net/brand-assets/4.3.0/favicons/favicon.ico" rel="shortcut icon" type="image/x-icon"/><link href
="/manifest.json" rel="manifest"/><meta content="Trustpilot" name="application-name"/><meta content="#1c1c1c" name="theme-col
or"/><link href="https://cdn.trustpilot.net/brand-assets/4.3.0/favicons/apple-touch-icon.png" rel="apple-touch-icon" sizes="1
80x180"/><link href="https://cdn.trustpilot.net/brand-assets/4.3.0/favicons/favicon-32x32.png" rel="icon" sizes="32x32" type
="image/png"/><link href="https://cdn.trustpilot.net/brand-assets/4.3.0/favicons/favicon-16x16.png" rel="icon" sizes="16x16"
type="image/png"/><link color="#00b67a" href="https://cdn.trustpilot.net/brand-assets/4.3.0/favicons/safari-pinned-tab.svg" r
el="mask-icon"/><meta content="Trustpilot" name="apple-mobile-web-app-title"/><meta content="#1c1c1c" name="msapplication-Til
eColor"/><script>
```

Figur 4 Skärmdump av soup-objektet

3.1.3 Steg 3: Data extrahering

När HTML-koden har strukturerats i *soup*-objekt är det dags att hämta ut själva data från recensionerna. De datatyper som ska extraheras från varje recension är: titel, brödtext, gradering, plats och datum. Alla dessa syns för användaren på webbsidan och finns nu tillgängliga i *soup*-objektet som HTML-element.

För att hitta exakt var denna data finns, används webbläsarens utvecklarverktyg. Genom att inspektera webbsidans uppbyggnad kan man se vilket HTML-element som används, till exempel <h2> för titel, och vilka klasser dessa element har. Ett exempel på ett titel-element kan vara:

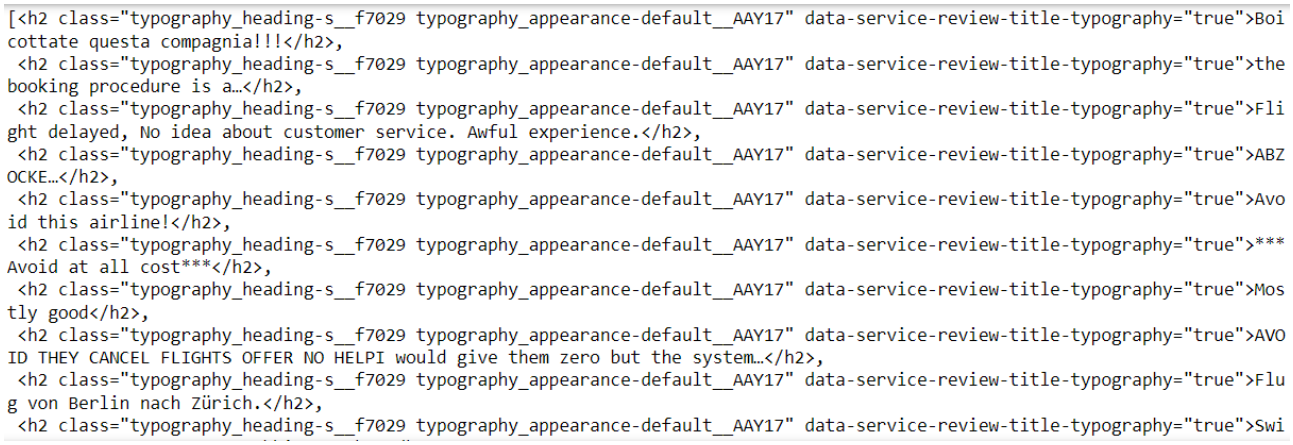
<h2 class="typography_heading-s__f7029 typography_appearance-default__AAY17">Titeln</h2>

Med hjälp av BeautifulSoup kan man söka efter alla element av typen <h2> som har en viss klass, med funktionen *findAll()*. Resultatet blir en lista med alla matchande HTML-element. Denna lista rensas sedan genom att extrahera endast textinnehållet från varje element med *.text*. På så vis får man en ren lista med bara titlar. Figur 5 visar Python-koden för sökning av titlar med hjälp av element- och klass-namn. (Vasilis, 2024)

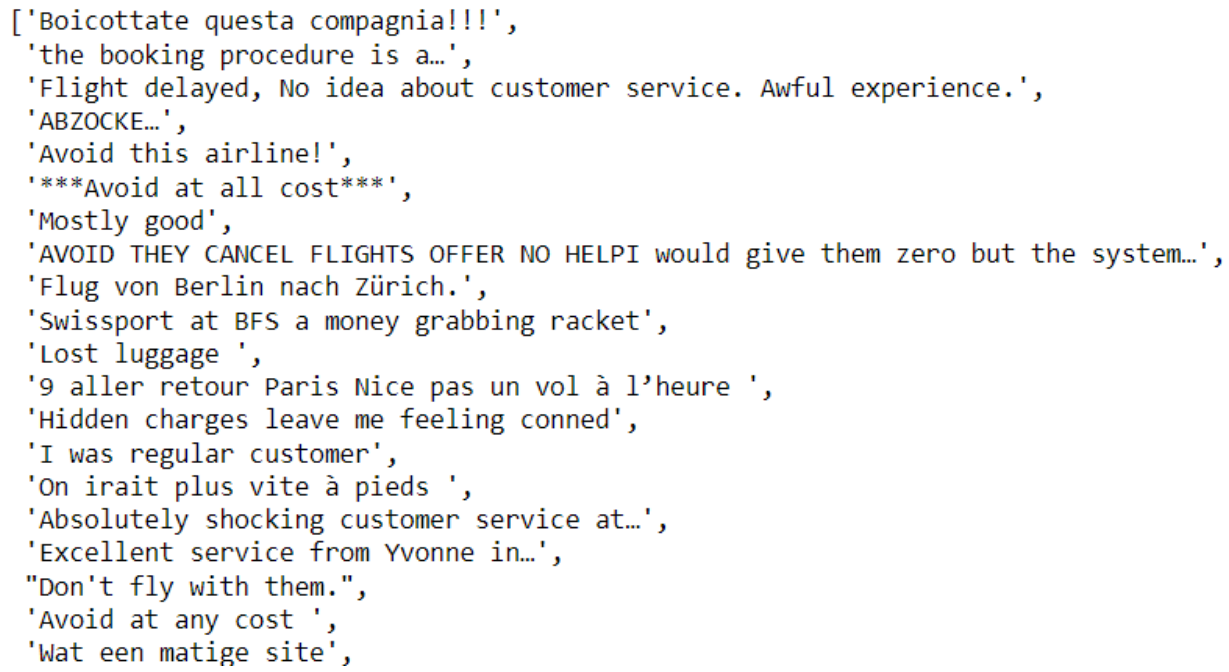
```python
titles = soup.findAll("h2", {"class": "typography_heading-s__f7029 typography_appearance-default__AAY17"})
titleClean = []
for row in titles:
    titleClean.append(row.text)
```

Figur 5 FindAll()-funktionen används för att hitta element med titlar

Nedanför är två skärmdumpar (Figur 6 och 7) av listan med titlarna före bearbetning och efter. I figur 7 finns ännu elementet kvar, medan i figur 8 finns endast titlarna kvar. Elementet är onödigt i det här fallet och endast titeln behövs. Genom att ta *text*-värdet av elementen och spara det i en ny lista får man en lista med endast titlarna.



Figur 6 Lista av element med titlar före bearbetning



Figur 7 Lista av titlar efter bearbetning

Samma metod upprepas för att hämta brödtext, gradering, plats och datum. För varje av dessa datatyper används rätt HTML-tag och klass, och informationen sparas i separata listor. Resultatet blir fem listor som var och en innehåller ett specifikt attribut från recensionerna. Denna uppdelning gör det enkelt att skapa en strukturerad tabell i nästa steg. Figur 8 visar sökningen av de resterande data. Data skrapas i ordningen brödtext -> gradering -> plats -> datum. Efter varje data skrapats, tas HTML-elementen bort så att endast texten blir kvar.

```
bodyText = soup.findAll("p", {"class": "typography_body-l__KUYFJ typography_appearance-default__AAY17 typography_color-black__5LY
bodyTextClean = []
for row in bodyText:
    bodyTextClean.append(row.text)


ratings = soup.findAll("div", {"class":"styles_reviewHeader__iU9Px"})
ratingClean = []
for row in ratings:
    ratingClean.append(str(row)[68])


location = soup.findAll("div", {"class":"typography_body-m__xgxZ_ typography_appearance-subtle__8_H2l styles_detailsIcon__Fo_ua"}
locationClean = []
locationClean1 = []
for row in location:
    locationClean.append(str(row.find_all("span")))

for row in locationClean:
    locationClean1.append(row[7:9])


date = soup.findAll("div", {"class":"typography_body-m__xgxZ_ typography_appearance-subtle__8_H2l styles_datesWrapper__RCEKH"})
dateClean = []
dateClean1 = []
for row in date:
    dateClean.append(str(row.find_all("time")))

for row in dateClean:
    dateClean1.append(row[67:77])
```

Figur 8 Skrapning av brödtext, gradering, plats och datum

3.1.4 Steg 4: Pandas-tabell

När all önskade data har extraherats från webbsidans HTML-kod och lagrats i separata listor, är det dags att organisera informationen. För detta används Pandas, ett kraftfullt Python-bibliotek för databehandling och analys. Med hjälp av Pandas kan man skapa en DataFrame, som är en tabelliknande datastruktur där varje kolumn innehåller ett datavärde, som till exempel titel eller datum. (Pandas, 2024a)

För att börja, används kommandot *import pandas as pd*, vilket gör att Pandas-funktionerna blir tillgängliga. En DataFrame skapas sedan med de fem listorna, titel, brödtext, gradering, plats och datum som kolumner. Varje rad i tabellen representerar en enskild recension. Utöver detta läggs en extra kolumn till som visar vilken webbsida recensionen hämtats från, vilket kan vara användbart vid analys av flera källor. Figur 9 visar Python-koden för att initialisera en Pandas-DataFrame.

```
reviewsDF = pd.DataFrame( [titleClean,bodyTextClean,ratingClean,locationClean1,dateClean1,website],
                          index=["Titel","Brödtext","Gradering","Plats","Datum","Webbsida"])
```

Figur 9 Skapandet av DataFramet reviewsDF

Eftersom Pandas normalt behandlar data med rader i fokus, och våra listor är kolumner, används funktionen *transpose()* för att vända på tabellen så att varje recension ligger som en egen rad. Detta gör tabellen mer logisk och lättläst. Med funktionen *head()* kan man sedan skriva ut de första raderna och kontrollera att tabellen ser korrekt ut. Figur

10 visar *transpose()*-funktionen samt *head()*-funktionen. Figur 11 visar de 20 första raderna av tabellen. Tabellen ger en bättre bild över hur data är strukturerat och är till nytta när man ska bearbeta det vidare. (Pandas, 2024d)

```
reviewsDF = reviewsDF.transpose()
reviewsDF.head(20)
```

Figur 10 Tabellen flyttas om från rader till kolumner

	Titel	Brödtext	Gradering	Plats	Datum	Webbsida
0	Boicottate questa compagnia!!!	Aereo fatto scendere a Cagliari anziché a Olbi...	1	IT	2024-09-29	Trustpilot
1	the booking procedure is a...	the booking procedure is a nightmare,retrievin...	2	GB	2024-09-29	Trustpilot
2	Flight delayed, No idea about customer service...	Flight delayed. No information. Finally arrive...	1	GB	2024-09-29	Trustpilot
3	ABZOCKE...	Wir wollten am 17.8.von Berlin nach Teneriffa ...	1	DE	2024-09-29	Trustpilot
4	Avoid this airline!	We had a terrible experience with Easyjet rece...	1	GB	2024-09-30	Trustpilot
5	***Avoid at all cost***	Rude customer service reps both at Gatwick and...	1	GB	2024-09-30	Trustpilot
6	Mostly good	Mostly good, but my wife cold not obtain her b...	3	DE	2024-09-30	Trustpilot
7	AVOID THEY CANCEL FLIGHTS OFFER NO HELP! would...	I would give them zero but the system won't le...	1	GB	2024-09-30	Trustpilot
8	Flug von Berlin nach Zürich.	Flug von Berlin nach Zürich.Ich durfte ohne An...	1	AT	2024-09-30	Trustpilot
9	Swissport at BFS a money grabbing racket	Easyjet ground handling (Swissport) at Belfast...	1	GB	2024-09-30	Trustpilot
10	Lost luggage	I had booked a flight from Amsterdam to Liverp...	1	US	2024-09-30	Trustpilot
11	9 aller retour Paris Nice pas un vol à l'heure	J'ai réaliser plus de 9 aller retour Paris Nic...	1	FR	2024-09-29	Trustpilot
12	Hidden charges leave me feeling conned	A company who never fail to leave a bad taste ...	1	GB	2024-09-27	Trustpilot
13	I was regular customer	I was regular customer, very deceiving when th...	1	GB	2024-09-30	Trustpilot
14	On irait plus vite à pieds	Encore et encore en retard de deux heures. Com...	1	FR	2024-09-30	Trustpilot
15	Absolutely shocking customer service at...	Absolutely shocking customer service at the ai...	1	GB	2024-09-30	Trustpilot
16	Excellent service from Yvonne in...	Excellent service from Yvonne in customer serv...	5	GB	2024-09-29	Trustpilot
17	Don't fly with them.	Flight there and back were both delayed, had t...	1	CH	2024-09-30	Trustpilot
18	Avoid at any cost	First time using the company. We had a flight ...	1	GR	2024-09-30	Trustpilot
19	Wat een matige site	Wat een matige site. Ik maak een account aan, ...	2	NL	2024-09-30	Trustpilot

Figur 11 Första 20 raderna av tabellen

För att säkerställa att data är komplett inför vidare analys används funktionen *dropna()*. Denna funktion tar bort alla rader som innehåller tomma värden – till exempel recensioner som saknar brödtext eller gradering. Genom att rensa data i detta skede minimeras risken för fel i nästa steg, då NLP-tekniker ska tillämpas. (Pandas, 2024b)

Efter tabellen satts ihop och städats upp kan undersökningen ta nästa steg mot att bearbeta data för att hitta de mest förekommande ordparen i negativa och positiva recensioner.

3.2 NLP

Efter att recensionerna skrapats och lagts in i en Pandas-tabell inleds NLP-bearbetningen. Syftet är att extrahera de vanligaste ordparen från recensionerna. I detta kapitel redogörs för de olika stegen i processen. Först delas recensionerna in i positiva och negativa baserat på deras betyg. Därefter identifieras vilket språk som är mest frekvent förekommande i materialet, och endast recensioner skrivna på detta språk används för vidare bearbetning. Efter språkfiltreringen förbereds recensionerna för NLP genom en serie språktekniska steg, inklusive textnormalisering, tokenisering, borttagning av stop words och lemmatisering. Slutligen extraheras ordparen från de bearbetade recensionerna, vilket möjliggör en analys av de vanligaste teman och återkommande uttrycken i kundernas feedback.

3.2.1 Steg 1: Fördelning av positiva och negativa recensioner

För att kunna analysera sentiment, det vill säga om en recension är positiv eller negativ, används recensionens gradering som kriterium. En gradering på tre, fyra eller fem stjärnor räknas som en positiv recension, medan ett eller två stjärnor räknas som negativ.

Figur 12 visar fördelningen på recensionernas betyg för flygbolaget EasyJet. Det här ger en tydlig bild över hur stor del av recensionerna kommer att vara positiva och hur många kommer att vara negativa.

	Gradering
1	4856
5	374
2	140
4	87
3	80

Figur 12 Fördelning av graderingen på recensionerna

För att avgöra utlåtandet går programmet igenom varje recension och undersöker dess gradering. Med hjälp av en *for*-loop och en *if*-sats kontrolleras varje betyg, och resultatet, “positiv” eller “negativ”, sparas i en lista. Denna lista läggs sedan som en ny kolumn i Pandas-tabellen, vilket gör det enkelt att filtrera recensioner efter sentiment i kommande steg. Figur 13 visar *for*-loopen för markering av recensionernas utlåtande. Figur 14 visar Pandas-tabellen med nya kolumnen, ”Utlåtande”.

```
pos_neg = []
for row in reviewsDF['Gradering']:
    if row >= 3:
        #positive
        pos_neg.append("Positiv")
    else:
        #negative
        pos_neg.append("Negativ")
```

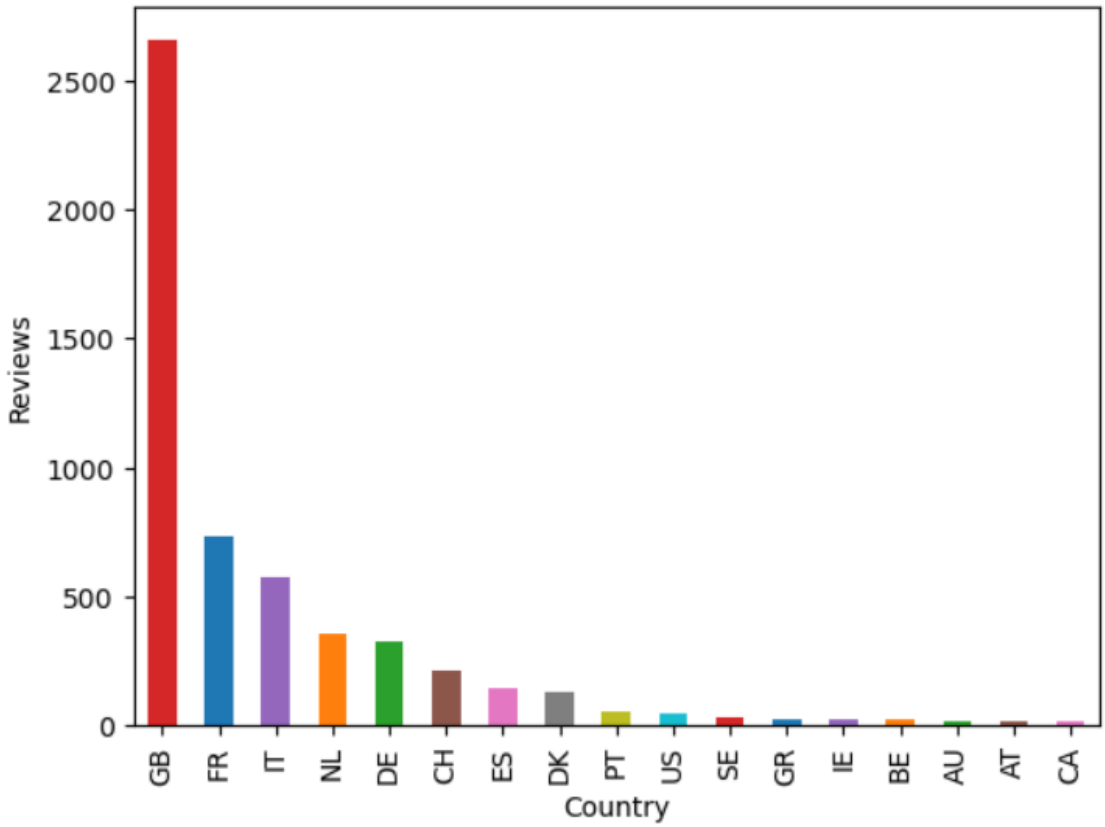
Figur 13 For-loop för att markera recensionernas utlåtande

	Titel	Brödtext	Gradering	Plats	Datum	Webbsida	Utlåtande
0	Never again	Never again. Marketing is not true - cheap fli...	1	DE	2024-10-07	Trustpilot	Negativ
1	Maurice one of the cabin crew on our...	Maurice one of the cabin crew on our flight EZ...	5	GB	2024-10-06	Trustpilot	Positiv
2	Flug 2 Std vor Abflug storniert, keine Alterna...	Flug wurde 2 Std vor Abflug storniert, kein Er...	1	DE	2024-10-05	Trustpilot	Negativ
3	Demuetigung	...ich fühle mich immer noch ungeheuer gedemüt...	1	DE	2024-10-07	Trustpilot	Negativ
4	Autoverhuur: niet doen!!!	Wij hadden een vlucht geboekt en daarbij een a...	1	NL	2024-10-07	Trustpilot	Negativ

Figur 14 Tabellen med nya kolumnen 'Utlåtande'

3.2.2 Steg 2: Utredning av mest förekommande recensionsspråk

Eftersom NLP-verktyg ofta är språkberoende behöver man identifiera vilket språk som dominerar i datamängden. I detta projekt används recensionernas platsinformation som en första ledtråd, till exempel om flera recensioner kommer från Storbritannien, antyder det på att engelska är vanligt. I figur 15 kan man se att Storbritannien har skrivit överlägset mest recensioner för EasyJet. Från detta skulle man kunna härleda att engelska skulle vara det vanligaste språket i recensionerna.



Figur 15 Graf som visar frekvensen av recensioner i olika länder

För att säkerställa vilket språk recensionerna är skrivna på används Python-biblioteket *langdetect*, som kan identifiera språk för korta textstycken. Varje recension körs genom språkanalysen och resultatet, till exempel "en" för engelska, sparas i en ny kolumn i tabellen. Figur 16 visar *for*-loopen som granskar recensionernas språk med langdetect. (PyPI, 2021)

```
languages = []
for x in range(reviewsDF.size):
    try:
        language = ld.detect(reviewsDF['Brödtext'][x])
        languages.append(language)
    except LD_EXC:
        languages.append("error")
    except:
        pass
```

Figur 16 For-loop som granskar varje recensions språk

Eftersom många NLP-tekniker fungerar bäst på ett språk i taget, väljs det mest förekommande språket (i detta fall engelska) för vidare bearbetning. Alla recensioner på andra språk filtreras bort, vilket förenklar processen och förbättrar kvaliteten på analysen. Figur 17 visar recensionstabellen med nya kolumnen språk.

	Titel	Brödtext	Gradering	Plats	Datum	Webbsida	Utlåtande	Språk
0	Efficient flight with pleasant crew.	Boarding seemed to run smoothly. Cabin crew we...	4	GB	2024-10-15	Trustpilot	Positiv	en
1	Eazyjet = communication nulle en cas de pépin	Mon expérience avec EasyJet a été extrêmement ...	1	FR	2024-10-15	Trustpilot	Negativ	fr
2	Schlechter geht nicht !!!	Schlechter geht nicht !!!Flugstorno während de...	1	DE	2024-10-14	Trustpilot	Negativ	de
3	Één hele grote oplichterij	Één hele grote oplichterij. De handbagage was ...	1	NL	2024-10-13	Trustpilot	Negativ	nl
4	Handgepäcksabzocke	Was für eine Abzocke. Am Gate wurde vor dem Be...	1	DE	2024-10-14	Trustpilot	Negativ	de

Figur 17 Recensionstabellen med nya kolumnen språk

3.2.3 Steg 3: NLP förberabetning

Innan en text kan analyseras med NLP måste den förbehandlas. Det innebär att texten städas och omvandlas till ett format som datorer kan tolka på ett effektivt sätt. Följande tekniker används i denna process:

- Textnormalisering
- Tokenisering
- Stop word removal
- Lemmatisering

Många av dessa tekniker kommer att implementeras med NLTK-biblioteket. NLTK är ett Python-bibliotek för NLP på engelska. NLTK kan implementeras i programmet med koden *import nltk*. (NLTK, 2024)

Textnormalisering är ett viktigt förbearbetningssteg i NLP eftersom det standardiserar och rengör rå textdata. Först rensas texten från specialtecken och skiljetecken, och alla bokstäver görs små. Detta gör att ord som "Service" och "service" behandlas som samma ord. Normaliseringen görs med hjälp av RegEx (Regular Expressions), som identifierar och tar bort oönskade tecken från texten.

RegEx är ett sätt att söka och manipulera text. Det använder sig av mönster för att söka igenom texten. Figur 18 visar Python-koden för RegEx. Med mönstret *[^a-zA-Z0-9\\s]* hittas alla tecken i recensionerna som inte är alfanumeriska. Med *sub()*-funktionen kan alla sådana tecken bytas ut till en tom sträng. Med andra ord tas specialtecken bort från

recensionerna helt och hållet. I figur 19 körs *lower()*-funktionen på recensionen. Recensionen förvandlas så att alla bokstäver är små. (Mistry, 2024)

```
clean_text = re.sub(r'[^a-zA-Z0-9\s]', '', review)
```

Figur 18 RegEx-funktion för textnormalisering

```
review = clean_text.lower()
```

Figur 19 lower()-funktionen används för att göra alla bokstäver små

Efter normalisering delas texten upp i mindre enheter, så kallade tokens, vanligtvis enskilda ord. Detta görs med funktionen *word_tokenize()* från NLTK-biblioteket. Tokenisering är grunden för all vidare analys, då det tillåter programmet att arbeta med varje ord för sig. Figur 20 visar koden för tokeniseringen av recensionerna. (Jain, 2024)

```
word_tokens = word_tokenize(review)
```

Figur 20 word_tokenize-funktionen används för att tokenisera recensionerna

Vanliga ord som inte tillför någon särskild betydelse, som “och”, “är”, “att” tas bort från texten. Dessa kallas stop words. Genom att rensa bort dessa kan analysen fokusera på mer betydelsebärande ord som bättre speglar kundernas upplevelser. En färdig lista med stop words laddas ner från NLTK. Figur 21 visar implementeringen av *stop_words* listan, samt *for*-loopen som jämför stop orden med recensionens tokens. (Jain, 2024)

```
stop_words = set(stopwords.words('english'))
    for w in word_tokens:
        if w not in stop_words:
            filtered_sentence.append(w)
```

Figur 21 Stop orden tas bort från recensionerna

Slutligen reduceras orden till sina grundformer med hjälp av lemmatisering. Till exempel omvandlas "hoppade", "hoppar" och "hoppat" till "hoppa". Detta görs med funktionen *WordNetLemmatizer()* i NLTK. Lemmatisering minskar variationen i texten och gör det lättare att räkna förekomster av ord och ordpar. I figur 22 visas hur *WordNetLemmatizer()* initialiseras, varefter recensionernas tokens körs igenom en *for*-loop. I *for*-loopen lemmatiseras recensionernas alla tokens. (Jain, 2024)

```
wl = WordNetLemmatizer()
lemma_words = []
for i in word_tokens:
  lemma_words.append(wl.lemmatize(i))
```

Figur 22 Orden i recensionerna lemmatiseras

Alla stegen, textnormalisering, tokenisering, stop word removal och lemmatisering kombineras i en funktion som bearbetar varje recension. Resultatet blir en lista med rensade och förenklade ord som sparas i en ny kolumn i tabellen under namnet Tokeniserad. Nedan är ett exempel på en recension före och efter NLP-förbearbetning.

Shocking company, make excuses for not paying expenses owed to you for their flight delays!, not the first time I have had issues with their service. I should have learned my lesson last time and book with an airline who cares about their customers.

[‘shocking’, ‘company’, ‘make’, ‘excuse’, ‘paying’, ‘expense’, ‘owed’, ‘flight’, ‘delay’, ‘first’, ‘time’, ‘issue’, ‘service’, ‘learned’, ‘lesson’, ‘last’, ‘time’, ‘book’, ‘airline’, ‘care’, ‘customer’]

Figur 23 visar recensionstabellen med nya kolumnen *Tokeniserad* som innehåller bearbetade versionen av recensionernas brödtext.

	Titel	Brödtext	Gradering	Plats	Datum	Webbsida	Utlåtande	Engelska	Tokeniserad
0	Efficient flight with pleasant crew.	Boarding seemed to run smoothly. Cabin crew we...	4	GB	2024-10-15	Trustpilot	Positiv	True	[boarding, seemed, run, smoothly, cabin, crew,...
1	Eazyjet = communication nulle en cas de pépin	Mon expérience avec EasyJet a été extrêmement ...	1	FR	2024-10-15	Trustpilot	Negativ	False	[mon, exprience, avec, easyjet, extrmement, dc...
2	Schlechter geht nicht !!!	Schlechter geht nicht !!!Flugstorno während de...	1	DE	2024-10-14	Trustpilot	Negativ	False	[schlechter, geht, nicht, flugstorno, whrend, ...
3	Één hele grote oplichterij	Één hele grote oplichterij. De handbagage was ...	1	NL	2024-10-13	Trustpilot	Negativ	False	[n, hele, grote, oplichterij, de, handbagage, ...
4	Handgepäcksabzocke	Was für eine Abzocke. Am Gate wurde vor dem Be...	1	DE	2024-10-14	Trustpilot	Negativ	False	[wa, fr, eine, abzocke, gate, wurde, vor, dem,...
5	Incapable de faire un effort commercial	Obligé d'annuler nos vacances pour cause de ma...	1	FR	2024-10-15	Trustpilot	Negativ	False	[oblig, dannuler, vacances, pour, cause, de, m...
6	Car rental scam with Easyjet	I did rent a car with Easyjet, with full insur...	1	GB	2024-10-15	Trustpilot	Negativ	True	[rent, car, easyjet, full, insurance, trkiye, ...
7	Flight cancelled left completely alone to resolve	Flight cancelled no contact available or help ...	1	GB	2024-10-13	Trustpilot	Negativ	True	[flight, cancelled, contact, available, help, ...
8	EasyJet/Goldcar Deception	We travelled from Gatwick to Bordeaux a few mo...	1	GB	2024-10-15	Trustpilot	Negativ	True	[travelled, gatwick, bordeaux, month, ago, pla...
9	A fuir !	Très mauvaise compagnie dont les agents ne pre...	1	FR	2024-10-14	Trustpilot	Negativ	False	[trs, mauvaise, compagnie, dont, le, agent, ne...

Figur 23 Recensionstabellen med nya kolumnen Tokeniserad

3.2.4 Steg 4: Extrahering av ordpar

Efter att recensionerna förberetts med NLP-tekniker delas de upp i två nya DataFrames, en för positiva recensioner och en för negativa. Endast recensioner på engelska används, och detta filtreras fram med Pandas-funktionen *loc()*. I figur 24 visas koden för filtreringen. En DataFrame skapas för både negativa och positiva recensioner på engelska. (Pandas, 2024c)

```
ENGpos_DF = reviewsDF.loc[(reviewsDF["Språk"] == 'en') & (reviewsDF['Utlåtande'] == "Positiv")]
ENGneg_DF = reviewsDF.loc[(reviewsDF["Språk"] == 'en') & (reviewsDF['Utlåtande'] == "Negativ")]
```

Figur 24 Engelska recensionerna delas i nya DataFrames för positiva och negativa recensioner

Nästa steg är att skapa två korpusar, en för varje grupp. En korpus är en samling av alla tokens (ord) från recensionerna, sparade i sin ursprungliga ordning. Detta är viktigt för att kunna analysera ordpar (bigrams), eftersom ordens placering i förhållande till varandra spelar roll. Figur 25 visar en självkonstruerad Python-funktion som skapar korpusen från Tokeniserad-kolumnen i DataFramen. DataFramen körs i en *for*-loop som tar varje token från Tokeniserad och lägger till dem i korpusen.

```python
def create_corpus(df):
    corpus = []
    for i in range(df.shape[0]):
        try:
            corpus += df['Tokeniserad'][i]
        except KeyError:
            pass
    return corpus
```

Figur 25 Python-funktion för skapandet av en korpus

För att skapa ordpar används funktionen *bigrams()* från NLTK, som automatiskt parar ihop ord som står bredvid varandra i korpusen. Detta görs både för den positiva och negativa gruppen. Resultatet är två listor med tusentals ordpar. Figur 26 visar exempel på negativa ordparen. Man kan se att nästa ordpar börjar med det sista ordet i tidigare ordparet. (Tutorialspoint, u.å.)

```
[('rent', 'car'),
 ('car', 'easyjet'),
 ('easyjet', 'full'),
 ('full', 'insurance'),
 ('insurance', 'trkiye'),
 ('trkiye', 'agency'),
 ('agency', 'insisting'),
 ('insisting', 'pay'),
 ('pay', 'full'),
 ('full', 'insurance'),
 ('insurance', 'cant'),
 ('cant', 'carand'),
 ('carand', 'suddenly'),
 ('suddenly', 'card'),
 ('card', 'machine'),
 ('machine', 'working'),
 ('working', 'depositsi'),
```

Figur 26 Lista på bigrams i negative korpusen

Slutligen används *Counter()*-funktionen i modulen *collections* för Python för att räkna hur ofta varje ordpar förekommer. Programmet visar de tio mest frekventa bigrams i varje grupp, vilket ger värdefulla insikter i vad kunderna ofta nämner i sina recensioner, både i positiv och negativ bemärkelse. Figur 27 visar de 10 vanligaste ordparen av negativa korpusen. Siffran bredvid ordparen berättar frekvensen av ordparet. (Python, u.å.)

```
[(('customer', 'service'), 395),
 (('easy', 'jet'), 285),
 (('flight', 'cancelled'), 129),
 (('cabin', 'bag'), 120),
 (('flight', 'delayed'), 94),
 (('cancelled', 'flight'), 79),
 (('even', 'though'), 69),
 (('return', 'flight'), 69),
 (('2', 'hour'), 67),
 (('easyjet', 'holiday'), 65)]
```

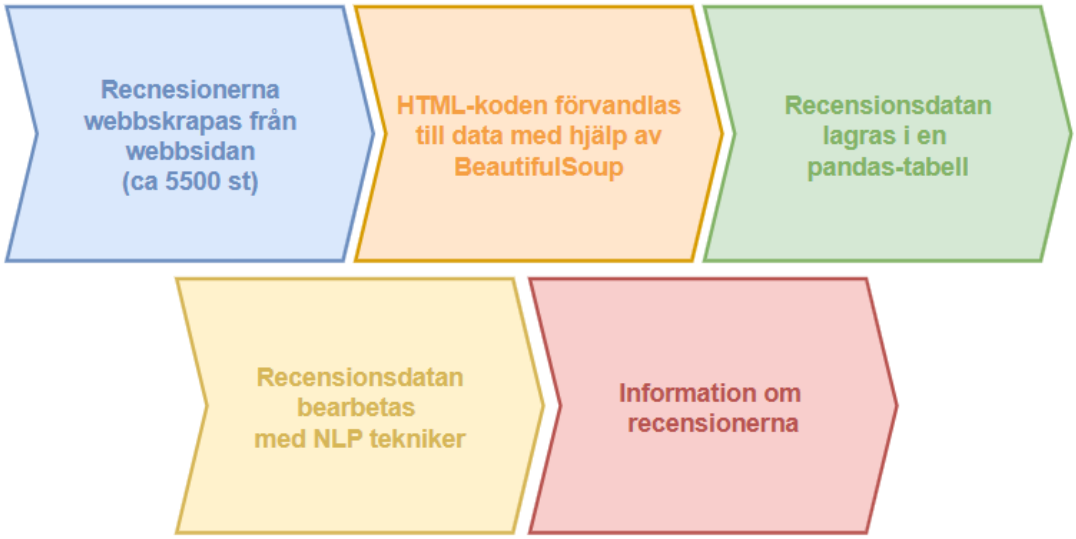
Figur 27 Lista på de 10 mest förekommande bigrams i negativa korpusen

4 Resultat

I detta kapitel kommer undersökningens resultat presenteras. Resultatet från både programvarans funktionalitet, samt de data som programmet gett kommer att visas.

4.1 Funktionalitet

Syftet med arbetet var att skapa en programvara som kan webbskrapa kundrecensioner och hitta de oftast förekommande ordparen i dem. Den färdiga programvaran skrapar recensioner, varefter den råa HTML-koden bearbetas till recensionsdata. Därefter lagras den färdiga datan i en tabell. Till sist använder programvaran NLP tekniker för att extrahera de oftast förekommande ordparen i recensionerna. Figur 28 visar stegen som programvaran tar för att nå det önskade resultatet.



Figur 28 Programvarans olika skeden

4.2 Data

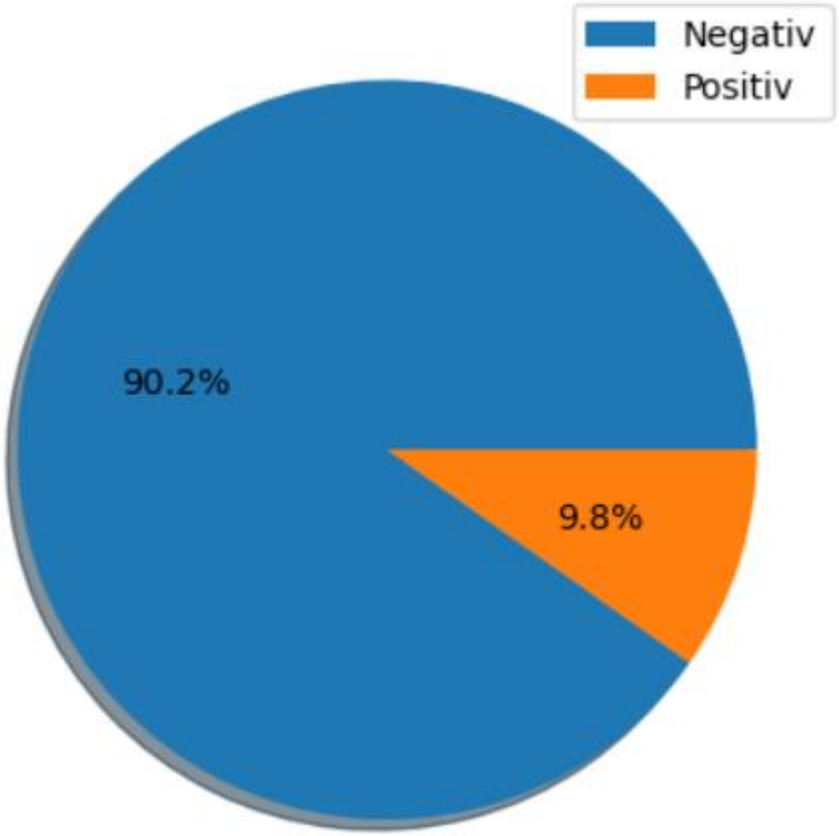
Det huvudsakliga målet var att hitta de oftast förekommande ordparen i både positiva och negativa recensioner. I detta kapitel kommer även annan statistik presenteras för att ge kontext åt resultatet. Data från två olika företags recensioner kommer att redovisas för att ge en bättre insikt i hurdan data kan fås med hjälp av denna programvara. Data som presenteras är:

- Fördelning på positiva och negativa recensioner.
- Fördelning på recensioner per språk.

- De oftast förekommande ordparen i positiva och negativa recensioner.

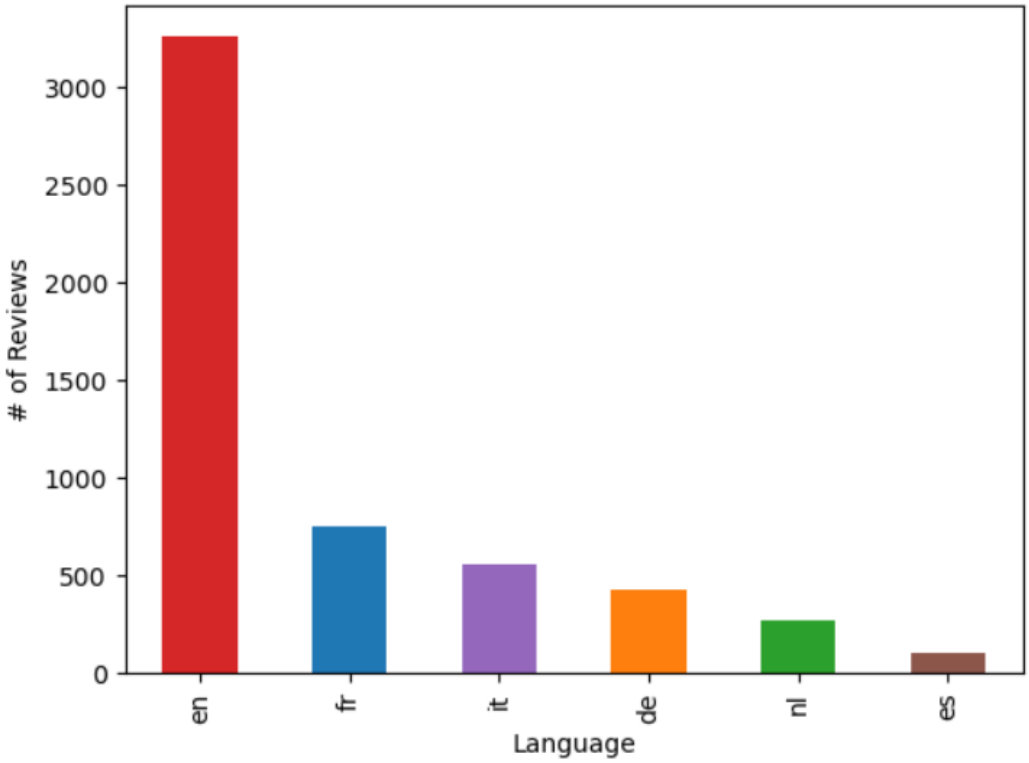
4.2.1 EasyJet

Det första företaget är EasyJet, ett brittiskt flygbolag. Totala mängden recensioner var 5537. Recensionerna delades enligt deras utlåtande i negativa och positiva recensioner. Recensionerna för EasyJet var överväldigande negativ. 90,2% av recensionerna var negativa och 9,8% var positiva. Detta betyder att 4996 recensioner är negativa, medan endast 541 recensioner är positiva. Figur 29 visar fördelningen på recensionerna hos EasyJet enligt utlåtande.



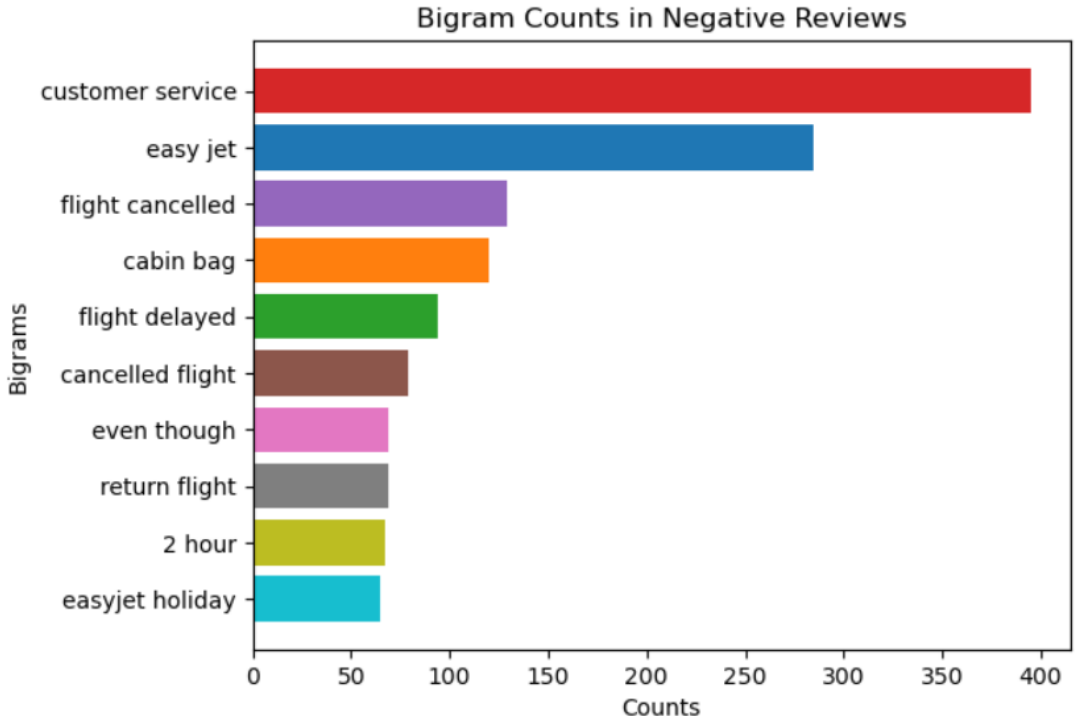
Figur 29 EasyJet: recensionernas utlåtande

Språkfördelningen var också ensidig. Över tre fjärdedelar av recensionerna var på engelska med 3256 recensioner. På andra plats var franska med ca 700 recensioner. Detta var en motivation för att använda engelskspråkiga recensioner i slutliga undersökningen. Figur 30 visar ett stapeldiagram med mängden recensioner per språk med över 100 recensioner för EasyJet. Y-axeln visar mängden recensioner och x-axeln visar språk. Språken på x-axeln från vänster till höger är: engelska, franska, italienska, tyska, holländska och spanska.

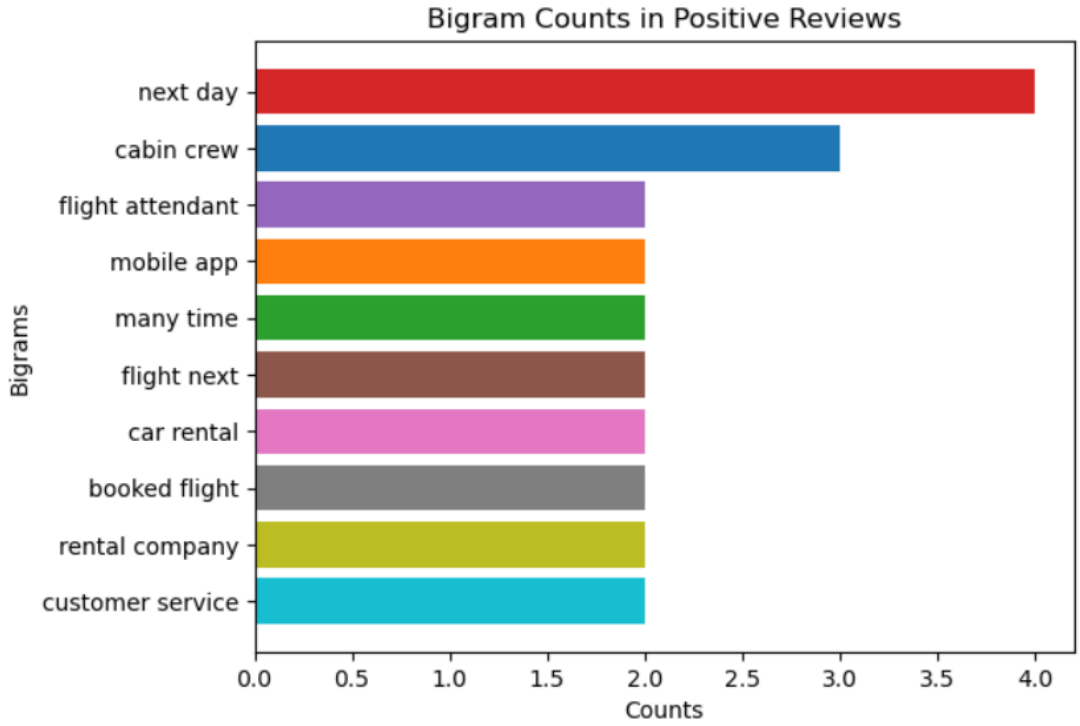


Figur 30 EasyJet: språk med flest recensioner

Till näst presenteras de tio oftast förekommande ordparen (bigrams) i recensionerna för EasyJet. Figur 31 visar ordparen i negativa recensionerna och figur 32 visar ordparen i positiva recensionerna. Figurerna är stapeldiagram. Y-axeln visar ordpar och x-axeln visar frekvensen av ordpar.



Figur 31 EasyJet: Mest förekommande ordpar i negativa recensioner



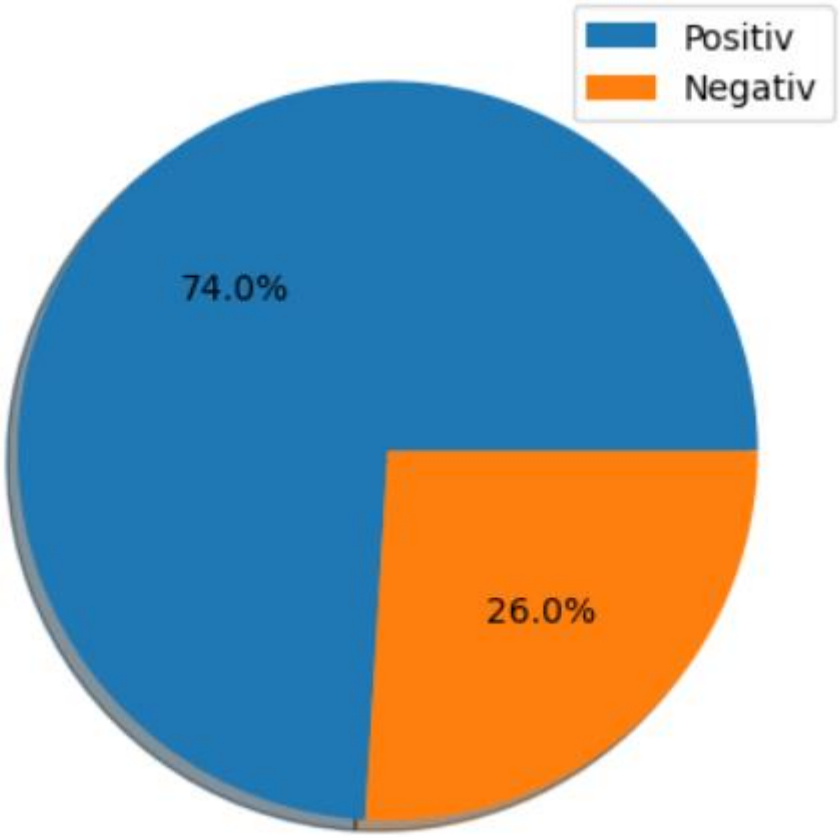
Figur 32 EasyJet: Mest förekommande ordpar i positiva recensioner

Negativa recensionernas vanligaste ordpar är ”customer service” som förekommer 395 gånger. I positiva recensionerna är ”next day” det vanligaste ordparet, men den

förekommer endast 4 gånger. Detta beror på den låga mängden positiva recensioner, vilket gör det mindre sannolikt att ordpar förekommer. ”Customer service” förekommer i de tio mest förekommande ordparen för både negativa och positiva recensioner.

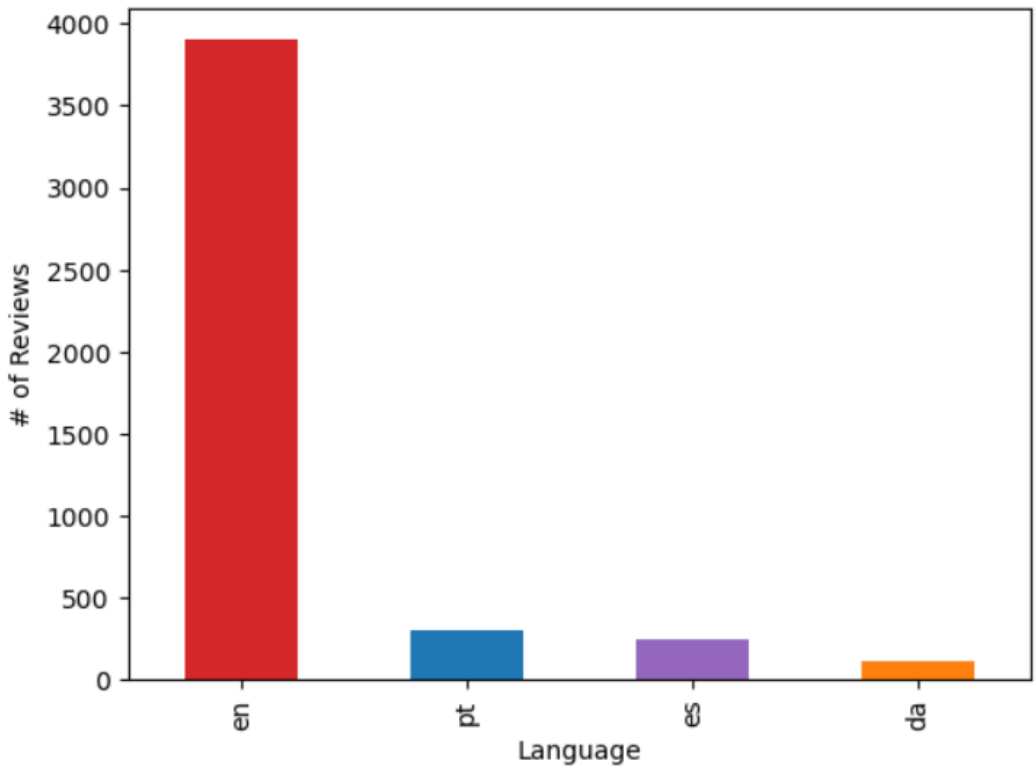
4.2.2 FlixBus

Det andra företaget är FlixBus, ett tyskt företag som ordnar busstrafik. Totala mängden recensioner var 5156. Recensionerna för FlixBus var mera positiva än för EasyJet. 26% av recensionerna var negativa och 74% var positiva, d.v.s 1339 recensioner är negativa och 3817 recensioner är positiva. Figur 33 visar fördelningen av negativa och positiva recensioner.



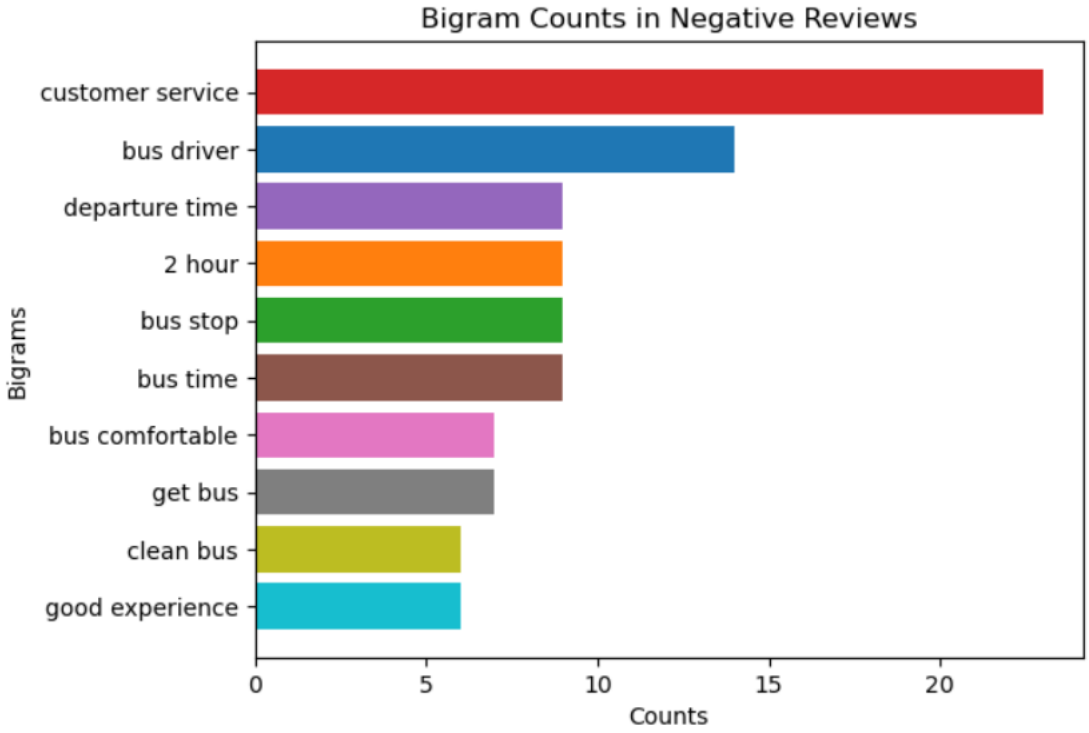
Figur 33 FlixBus: recensionernas utlåtande

Flest recensioner var på engelska med 3902 recensioner och på andra plats var portugisiska med 297 recensioner. Figur 34 visar ett stapeldiagram med mängden recensioner per språk med över 100 recensioner för FlixBus. Y-axeln visar mängden recensioner och x-axeln visar språk. Språken på x-axeln från vänster till höger är: engelska, portugisiska, spanska och danska.

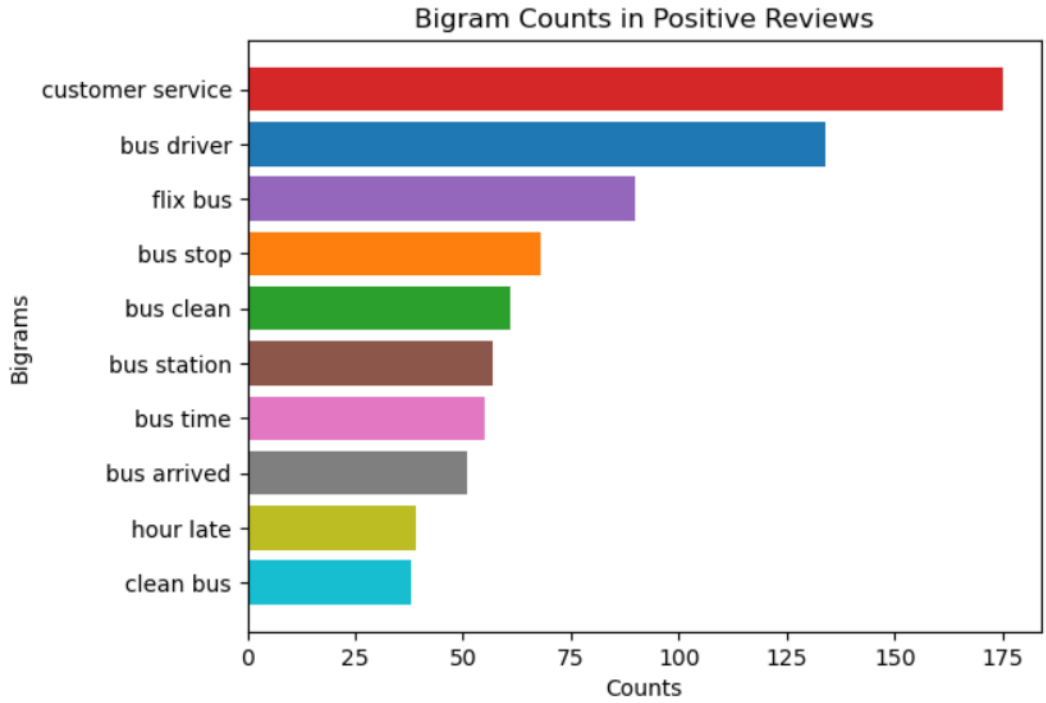


Figur 34 FlixBus: språk med flest recensioner

Till näst presenteras de tio vanligaste ordparen (bigrams) i recensionerna för FlixBus. Figur 35 visar ordparen i negativa recensionerna och figur 36 visar ordparen i positiva recensionerna. Figurerna är stapeldiagram. Y-axeln visar ordpar och x-axeln visar frekvensen av ordpar.



Figur 35 FlixBus: Mest förekommande ordpar i negativa recensioner



Figur 36 FlixBus: Mest förekommande ordpar i positiva recensioner

För FlixBus är de två mest förekommande ordparen exakt samma för både negativa och positiva recensioner; ”customer service” och ”bus driver”. I negativa recensioner förekom dessa ordpar 23 och 14 gånger, medan i positiva recensioner förekom de 175 och

134 gånger respektive. Eftersom FlixBus hade mera positiva recensioner är mängden ordpar också betydligt större. Många andra ordpar förekommer också i de tio mest förekommande ordparen före både negativa och positiva recensioner.

5 Diskussion

I detta kapitel kommer lärdomsprovets metodik och resultat diskuteras från en kritisk synvinkel. Eventuella brister i forskningen och fortsatta forskningsrekommendationer kommer att presenteras, men även saker som gick som förväntat kommer att nämnas.

Syftet med arbetet var att effektivt hitta ofta förekommande ordpar från kundrecensioner. Programvaran som jag utvecklade nådde detta mål med bra effektivitet. Webbrecensionerna skrapas ned med BeautifulSoup, varefter de arrangeras i en pandas-tabell. Därefter implementeras de diverse NLP-teknikerna för att extrahera ordparen från recensionerna. Även om programvaran fungerade som förväntat, så fungerar den endast för en webbsida, Trustpilot. Jag kunde ha utvidgat forskningen till flera olika webbsidor för att se hur man kan skrapa från flera olika webbsidor med en webbskrapare. Detta skulle ha ökat på komplexiteten av programvaran, men också gett mera mångsidiga data att extrahera ordpar från. Över lag så fungerar webbskraparen som förväntat och jag anser att det är en fungerande process för att hämta data från webben, men den kunde utvecklas vidare för att ge mångsidigare resultat.

Även om webbskraparen fungerade som förväntat, var resultaten som jag fick lite överraskande. För det första var ordparens mängd mindre än förväntat i vissa fall. I positiva recensionernas ordpar för EasyJet var frekvensen för det mest förekommande ordparet endast 4. 4 ordpar är ett för litet urval för att göra någon slags meningsfull slutsats av kundnöjdhet. Den låga mängden ordpar berodde dels till att det fanns få positiva recensioner för EasyJet till att börja med och att det filtrerades bort recensioner på andra språk än engelska. Detta kunde ha undvikts med en större mängd skrapade recensioner, vilket kunde ha nåtts till exempel genom att utvidga skraparen att fungera på flera webbsidor, som tidigare diskuterats. Man kunde även ha använt sig av recensioner på andra språk, vilket skulle ha gett ett större urval av recensioner. I detta projekt valde jag att endast använda mig av ett språk, eftersom jag tänkte det skulle minimera komplexiteten. Tillägg av andra språk skulle potentiellt krävt användning av flera NLP-verktyg för de olika språken, medan med endast engelska språket räckte ett verktyg.

En annan brist med resultatet kunde sägas vara validiteten på recensionerna. Webbskraparen som jag skrapat ser inte skillnad på en äkta recension, en bot eller en recension som är ett resultat av ’review bombing’; ett fenomen där flera personer koordinerat ger negativa recensioner till en tjänst eller ett företag. Trustpilot har regler för vem kan skriva en recension, men utan något behov att bevisa att man använt ett företags produkt eller tjänst kan vem som helst skriva en recension om vad som helst. För det här problemet måste man mest lite på webbsidan själv att radera recensioner som inte är äkta, men troligtvis finns det ändå recensioner som slipper igenom filternätet. Ofta är dessa oäkta recensioner dock korta eller helt utan text, i vilket fall det inte påverkar forskningens resultat på ett meningsfullt sätt.

Ordparen som förekom i både negativa och positiva recensioner var ibland samma. Detta kan bero på att ordparen i sig är neutrala och kan förekomma i både positiva och negativa recensioner, i vilket fall det inte är ett problem. Det är dock möjligt att en negativ recension av misstag markerats som positiv och vice versa. På Trustpilot kan en användare ge utlåtande från 1–5 på en produkt eller tjänst. En användare kan av misstag ha gett 5/5 när de menade ge 1/5. Detta gör att data blir skevat, eftersom programvaran kommer att markera recensionen som positiv, fast innehållet av den är negativ. Genom att använde sofistikerade NLP-verktyg kunde jag ha avgjort sentimentet i recensionerna utan att använda användarnas utlåtande från Trustpilot. Detta skulle potentiellt gjort ordparen noggrannare till deras verkliga sentiment.

En annan möjlighet för att både negativa och positiva recensioner var ibland samma kan vara att bigrams inte ger tillräckligt information. Ifall jag hade använt mig av n-grams av högre grad, som trigrams, kunde man ha fått mera kontext i resultatet. Detta skulle ha krävt en mycket större mängd recensioner för att få en tillräckligt stor frekvens av trigrams.

Även om resulterande data inte var perfekt, kan man ändå få en översiktlig bild på kundernas behov och beröm. Ordparen kan användas genom att ge avsikter till en kundservice chatbot. Ordpar som ”flight cancelled” och ”flight delayed” kan ges en avsikt som ’flygproblem’, medan ”bus clean” kan ges avsikten ’beröm’. Chatboten kan då tränas att reagera på ett visst sätt ifall en kund triggar någon av avsikterna. Till exempel ifall en kund skriver att ”Mitt flyg försenades med 24 timmar” så vet boten att det handlar om ett flygproblem och kan hjälpa kunden på rätt sätt.

5.1 Slutsatser

Syftet med detta lärdomsprov var att utveckla en metod för att effektivt extrahera ofta förekommande ordpar från webbskrapade kundrecensioner. Den huvudsakliga forskningsfrågan var: *Hur kan man effektivt hitta ofta förekommande ordpar från kundrecensioner på webben?* Resultaten visar att en kombination av webbskrapning med BeautifulSoup och NLP-tekniker med NLTK är ett effektivt och genomförbart sätt att uppnå detta mål.

Genom att skrapa recensioner från Trustpilot och bearbeta dem med tekniker som textnormalisering, tokenisering, lemmatisering och n-gram-analys kunde programvaran identifiera de vanligaste teman i både positiva och negativa recensioner. Programvaran fungerade enligt förväntan och lyckades extrahera värdefull information, trots vissa begränsningar.

Styrkan i arbetet låg i metodens enkelhet och möjligheten att skapa ett arbetsflöde från datainsamling till analys. Begränsningarna omfattade dock ett snävt urval av datakällor, språkfilter som uteslöt icke-engelska recensioner samt den potentiella partiskheten av

användargenererat innehåll. Dessa faktorer påverkade datamängdens bredd och validitet.

För framtida utveckling rekommenderas en utvidgning av webbskrapningen till flera recensionsplattformar och stöd för fler språk. Dessutom kunde vidare bearbetning av sentimentet direkt från texten, snarare än endast baserat på användarnas betyg, förbättra analysens noggrannhet. De extraherade ordparen kan utgöra en värdefull grund för vidare arbete med att utveckla och träna kundtjänst-chatbotar.

Arbetet visar att kombinerad användning av webbskrapning och NLP erbjuder en effektiv metod för att analysera stora mängder kundfeedback och utvinna praktiskt användbara insikter.

Källor

Cloudflare. (u.å.). *What is HTTP?*. https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/

Crummy. (u.å.). *Beautiful Soup Documentation*. https://www.crummy.com/software/BeautifulSoup/bs4/doc/

GeeksforGeeks. (1 maj 2024). *Generate bigrams with NLTK*. https://www.geeksforgeeks.org/generate-bigrams-with-nltk/

GeeksforGeeks. (8 april 2025). *Natural Language Processing (NLP) – Overview*. https://www.geeksforgeeks.org/natural-language-processing-overview/#nlp-tasks

Heath, A. (22 mars 2023). *Web Scraping 101: Tools, Techniques and Best Practices*. Medium. https://medium.com/geekculture/web-scraping-101-tools-techniques-and-best-practices-417e377fbeaf

Jain, A. (2 februari 2024). *All about Tokenization, Stop words, Stemming and Lemmatization in NLP*. Medium https://medium.com/@abhishekjainindore24/all-about-tokenization-stop-words-stemming-and-lemmatization-in-nlp-1620ffaf0f87

Johansson, R. (2023). *The One Spider To Rule Them All*. [Examensarbete, Kungliga Tekniska högskolan]. DiVA. https://www.diva-portal.org/smash/get/diva2:1774210/FULLTEXT01.pdf

Khder, M. A. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. *Int. J. Advance Soft Compu. Appl*, *13*(3), 146–147. 10.15849/IJASCA.211128.11

Lacasa, L. A., Dévora-Pajares, M., Zbib, R., & Fabregat, H. (2024). Intent Classification Methods for Human Resources Chatbots. *Procesamiento del Lenguaje Natural*, 73, 109-111. 10.26342/2024-73-8

Malik, P., Mittal, V., Nautiyal, L., & Ram, M. (2022). NLP techniques, tools, and algorithms for data science. *De Gruyter Series on the Applications of Mathematics in Engineering and Information Sciences*, 11, 123-141. 10.1515/9783110734652-006

Mistry, T. (5 april 2024). *Text-PreProcessing — Removing Punctuation and Special Characters*. Medium. https://medium.com/@mistrytejasm/text-preprocessing-removing-punctuation-and-special-characters-e3de4cece082

NLTK. (19 augusti 2024). *Natural Language Toolkit*. https://www.nltk.org/

Pandas. (20 september 2024a). *pandas documentation*. https://pandas.pydata.org/docs/index.html

Pandas. (2024b). *pandas.DataFrame.dropna*. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dropna.html

Pandas. (2024c). *pandas.DataFrame.loc*. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html

Pandas. (2024d). *pandas.DataFrame.transpose*. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.transpose.html#pandas.DataFrame.transpose

Patwardhan, N., Marrone, S., & Sansone, C. (2023). Transformers in the Real World: A Survey on NLP Applications. *Information*, *14*(4), 5-12. 10.3390/info14040242

PyPI. (7 maj 2021). *langdetect 1.0.9*. https://pypi.org/project/langdetect/

PyPI. (29 maj 2024). *requests*. https://pypi.org/project/requests/

Python. (u.å.). *collections — Container datatypes*. https://docs.python.org/3/library/collections.html

Rakovic, S. (2020). *Upphovsrättsligt intrång genom webbskrapning*. [Examensarbete, Lunds universitet]. Lund University Publications. https://lup.lub.lu.se/student-papers/record/9033376/file/9038950.pdf

Sapardic, J. (15 april 2025). *What Are Chatbot Intents: Classification, Use Cases, and Training Tips*. Tidio. https://www.tidio.com/blog/chatbot-intents/

Subex. (30 september 2023). *What is a corpus?*. https://www.subex.com/article/corpus/

Sulcas, A. (11 april 2025). *BeautifulSoup Tutorial - How to Parse Web Data With Python*. Oxylabs. https://oxylabs.io/blog/beautiful-soup-parsing-tutorial

Rawat, V. (28 augusti 2023). *A Must-Know Guide On Information Retrieval in NLP*. Pickl.AI. https://www.pickl.ai/blog/information-retrieval-in-nlp/

Tutorialspoint. (u.å.). *Python – Bigrams*. https://www.tutorialspoint.com/python_text_processing/python_bigrams.htm

Vasilis, T. (6 maj 2024). *Beautiful Soup: find by class (Python tutorial)*. Apify. https://blog.apify.com/beautifulsoup-find-by-class/

W3Schools. (u.å.-a). *HTML Introduction*. https://www.w3schools.com/html/html_intro.asp

W3Schools. (u.å.-b). *Python Requests get() Method*. https://www.w3schools.com/python/ref_requests_get.asp

W3Schools. (u.å.-c). *Python requests.Response Object*. https://www.w3schools.com/python/ref_requests_response.asp

WebHarvy. (u.å.). *What is Web Scraping used for ?* https://www.webharvy.com/articles/web-scraper-use-cases.html