NLP Project 1

Chosen Topic: *Customer Review Classification for Vitamins and Supplements*
Students: Fatih Arslan (B211202023), Yusuf İnan (B211202070)
Code Repository: GitHub - NLP_Product_Review_Classification

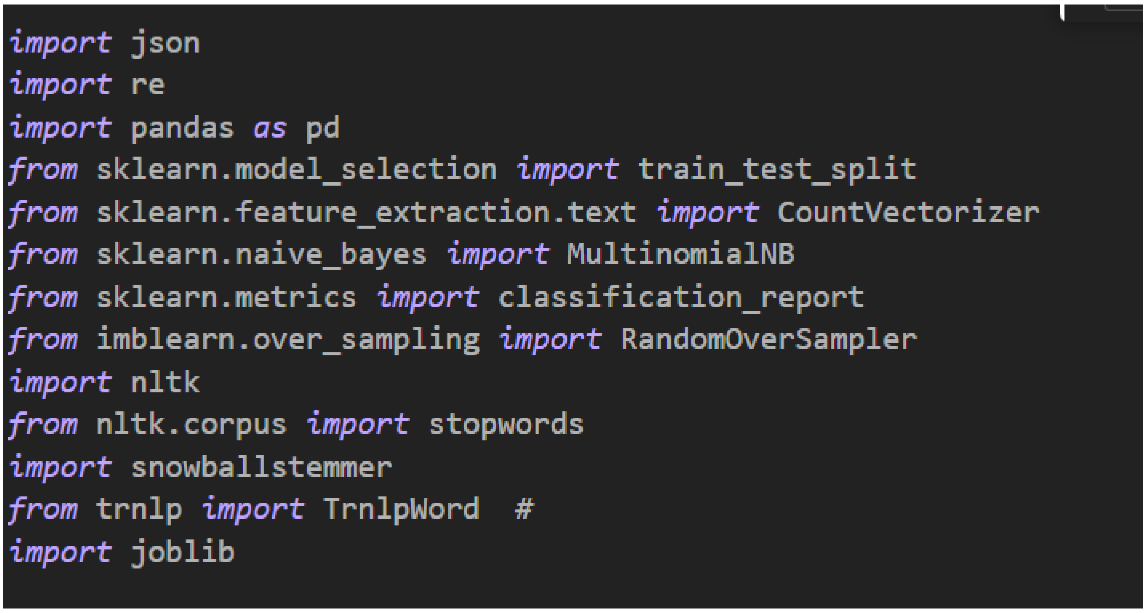
In this project, we focused on classifying customer reviews related to vitamins and supplements. We first got a dataset containing customer feedback and then developed a Python-based NLP application to analyze it.

Our preprocessing pipeline included **text normalization and cleaning, tokenization, lemmatization, and stemming** to prepare the textual data for analysis. We employed both the **Bag-of-Words** and **Word2Vec** vectorization methods to represent text numerically, allowing us to compare their effectiveness in classification performance.

For the classification task, we utilized the **Naïve Bayes** algorithm. We evaluated and compared the performance of each vectorization approach using various **performance metrics** and supported our findings with **visual representations and graphs**.

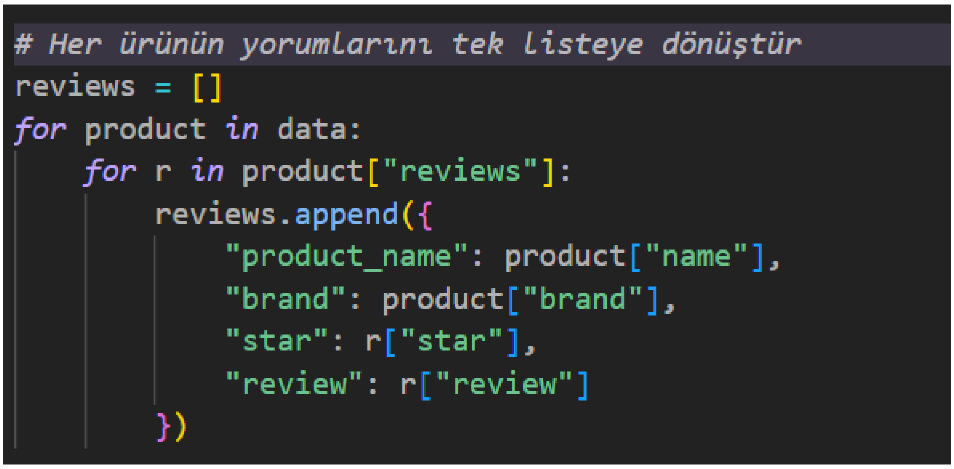
This project demonstrates the application of fundamental NLP techniques in sentiment classification and highlights the impact of different vectorization methods on model accuracy and interpretability.

Libraries we used:



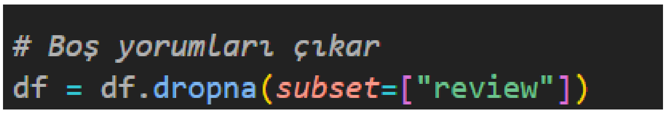
```
import json
import re
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
from imblearn.over_sampling import RandomOverSampler
import nltk
from nltk.corpus import stopwords
import snowballstemmer
from trnlp import TrnlpWord  #
import joblib
```

Collects all product reviews into a single list with product name, brand, rating, and review text.



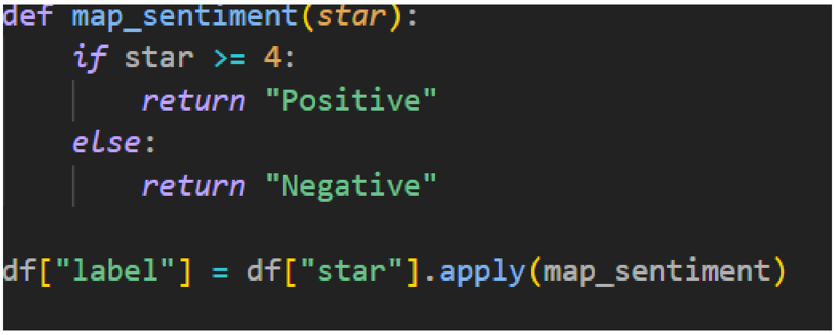
```
# Her ürünün yorumlarını tek listeye dönüştür
reviews = []
for product in data:
    for r in product["reviews"]:
        reviews.append({
            "product_name": product["name"],
            "brand": product["brand"],
            "star": r["star"],
            "review": r["review"]
        })
```

Removes rows where the review text is missing.



```
# Boş yorumları çıkar
df = df.dropna(subset=["review"])
```

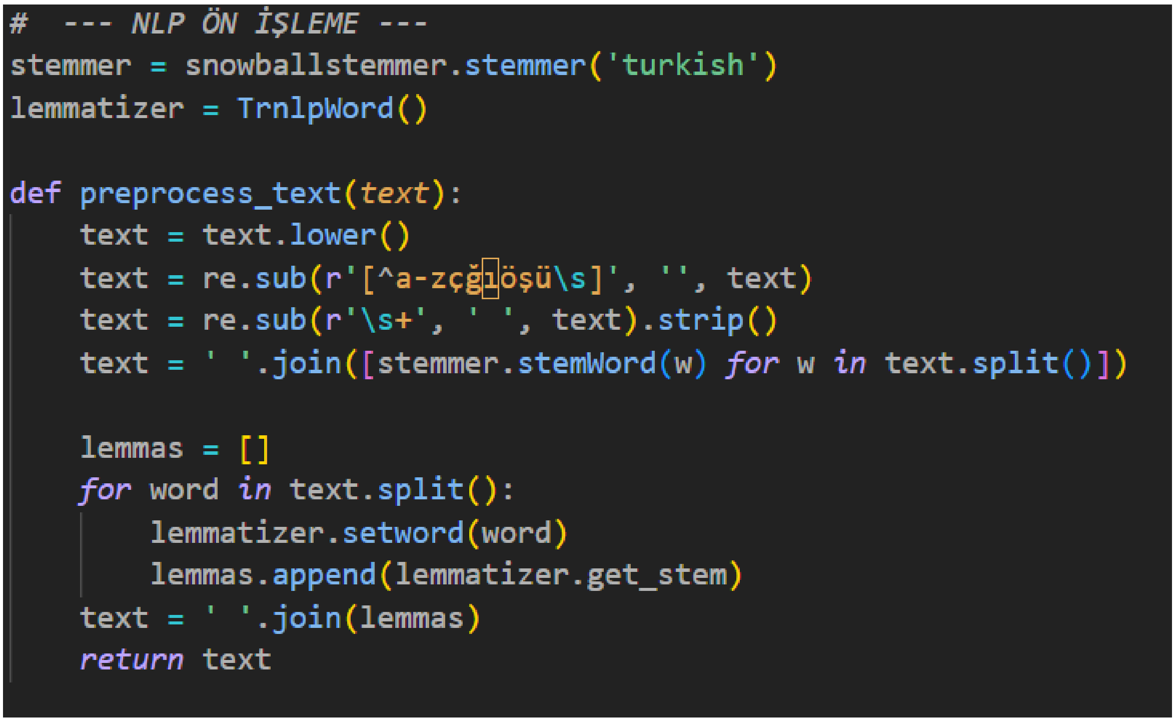
Converts numeric ratings into sentiment labels — 4 or 5 stars = *Positive*, otherwise *Negative*.



```
def map_sentiment(star):
    if star >= 4:
        return "Positive"
    else:
        return "Negative"

df["label"] = df["star"].apply(map_sentiment)
```

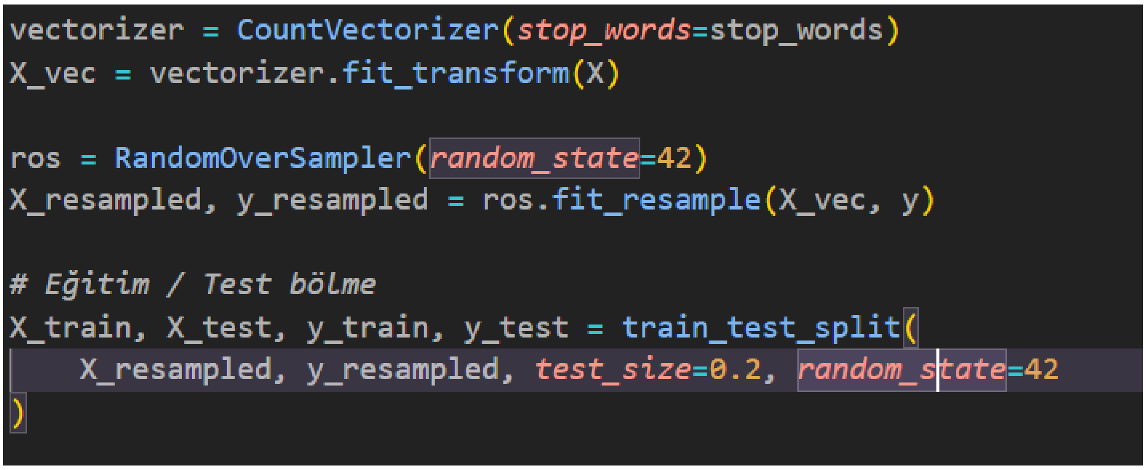
Cleans and normalizes text by:

- Lowercasing
- Removing special characters
- Applying stemming and lemmatization for Turkish words.



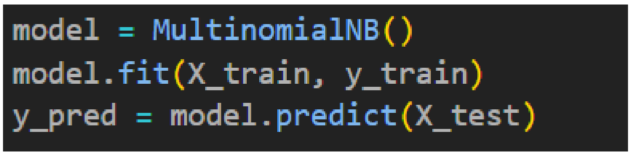
```
#  --- NLP ÖN İŞLEME ---
stemmer = snowballstemmer.stemmer('turkish')
lemmatizer = TrnlpWord()

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zçğıöşü\s]', '', text)
    text = re.sub(r'\s+', ' ', text).strip()
    text = ' '.join([stemmer.stemWord(w) for w in text.split()])

    lemmas = []
    for word in text.split():
        lemmatizer.setword(word)
        lemmas.append(lemmatizer.get_stem)
    text = ' '.join(lemmas)
    return text
```

Balances the dataset by duplicating minority class samples to prevent bias.



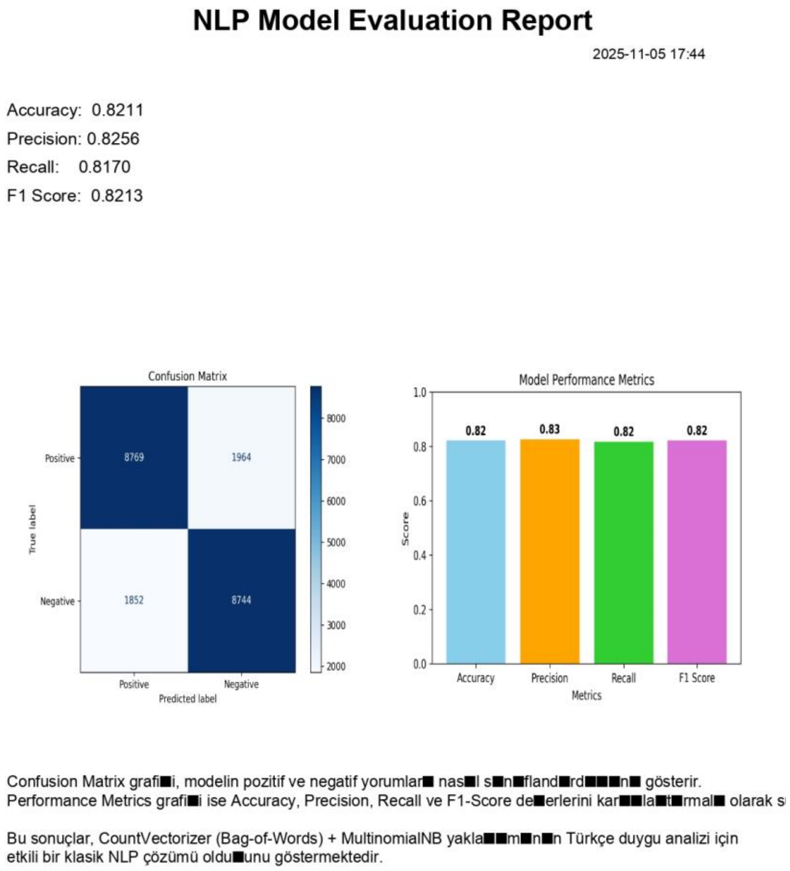
```
vectorizer = CountVectorizer(stop_words=stop_words)
X_vec = vectorizer.fit_transform(X)

ros = RandomOverSampler(random_state=42)
X_resampled, y_resampled = ros.fit_resample(X_vec, y)

# Eğitim / Test bölme
X_train, X_test, y_train, y_test = train_test_split(
    X_resampled, y_resampled, test_size=0.2, random_state=42
)
```

Trains a Naive Bayes classifier on training data and predicts labels for test data.



```
model = MultinomialNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Results when we use **Bag-of-Words** method;



NLP Model Evaluation Report

2025-11-05 17:44

Accuracy: 0.8211
Precision: 0.8256
Recall: 0.8170
F1 Score: 0.8213

Confusion Matrix grafi■i, modelin pozitif ve negatif yorumlar■ nas■l s■n■fland■rd■■■n■ gösterir.
Performance Metrics grafi■i ise Accuracy, Precision, Recall ve F1-Score de■erlerini kar■■la■t■rmal■ olarak s

Bu sonuçlar, CountVectorizer (Bag-of-Words) + MultinomialNB yakla■■m■n■n Türkçe duygu analizi için etkili bir klasik NLP çözümü oldu■unu göstermektedir.

Results when when we use **word2vec** method;

NLP Model Evaluation Report

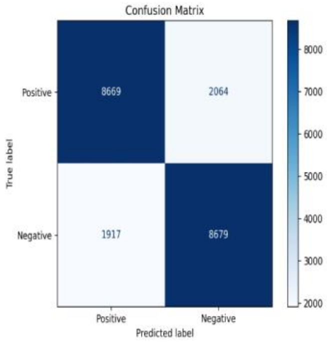
2025-11-05 18:05

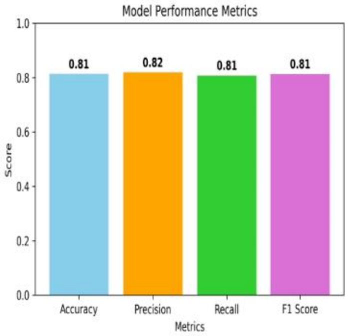
Accuracy: 0.8134

Precision: 0.8189

Recall: 0.8077

F1 Score: 0.8133





The Confusion Matrix chart shows how the model classifies positive and negative reviews.
The Performance Metrics chart presents a comparative view of Accuracy, Precision, Recall, and F1-Score values

These results demonstrate that the CountVectorizer (Bag-of-Words) + MultinomialNB approach
is an effective classical NLP solution for Turkish sentiment analysis.