Sentiment Analysis of Amazon Product Reviews

Tools

- Python
- Scikit-Learn
- NLTK
- NumPy, Pandas, Matplotlib

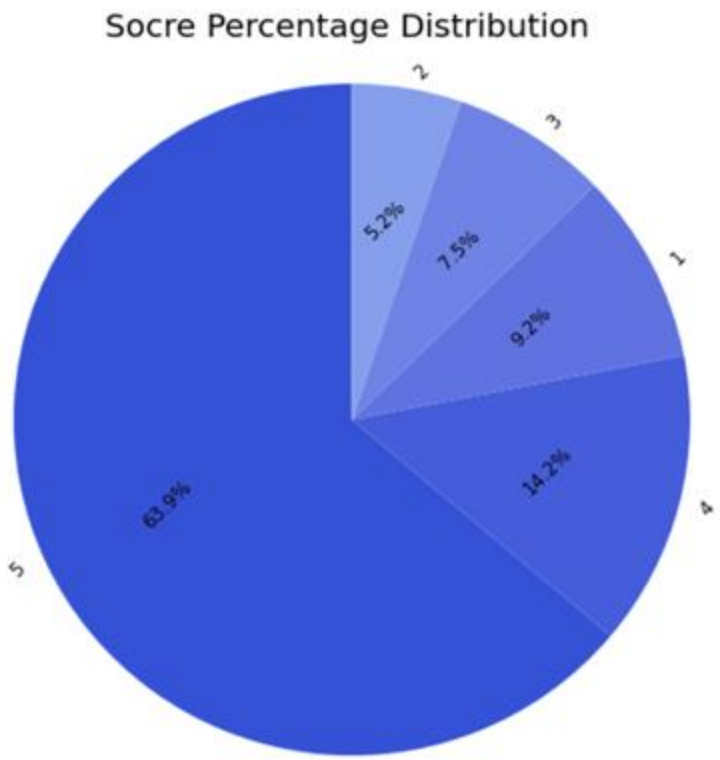
Dataset: Amazon Product Reviews

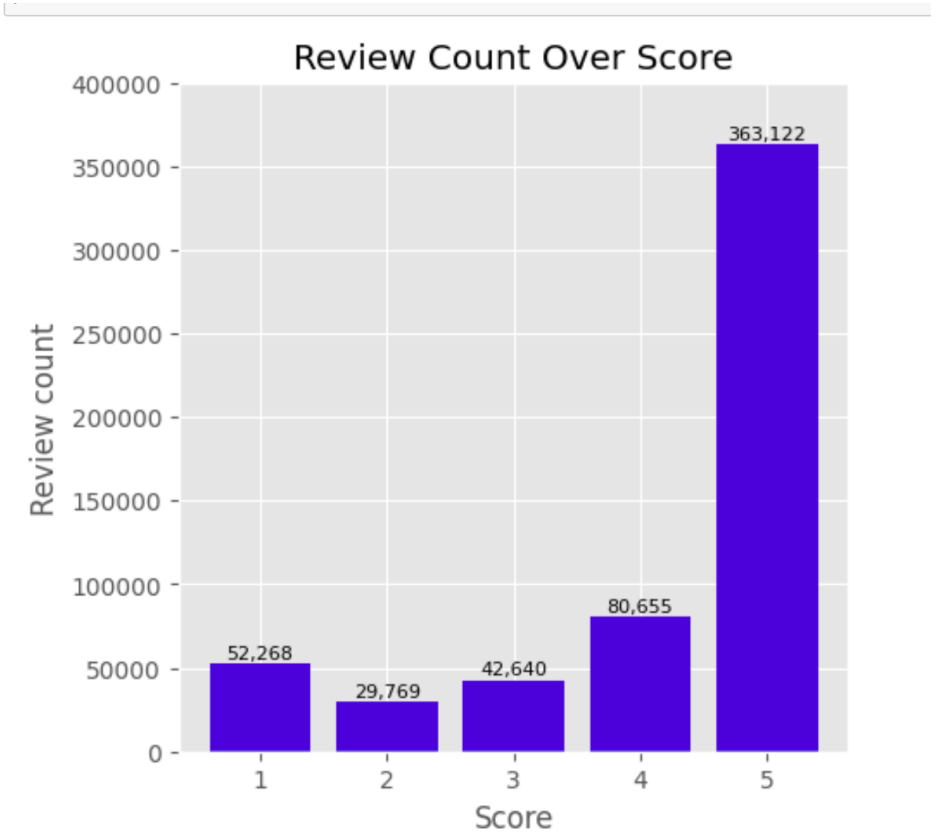
- Amazon fine food reviews
- 568,454 reviews
- 74,258 products
- 256,059 users
- Data source: https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews

Introduction

- **Problem:** Building models which can identify the sentiment (positive or negative)
- **Scope of work:**
  - Analyzing the input text data and the corresponding response variables (ratings)
  - Performing basic pre-processing to prepare the data for modeling
  - Featurizing the reviews text
  - Building machine learning models to classify text as either positive or negative sentiment (1 or 0)

Data Exploration





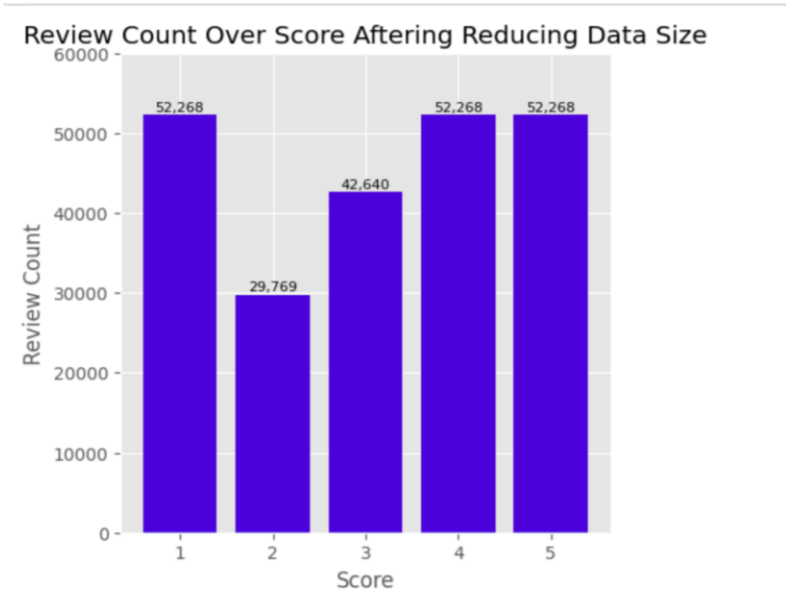
Methodology

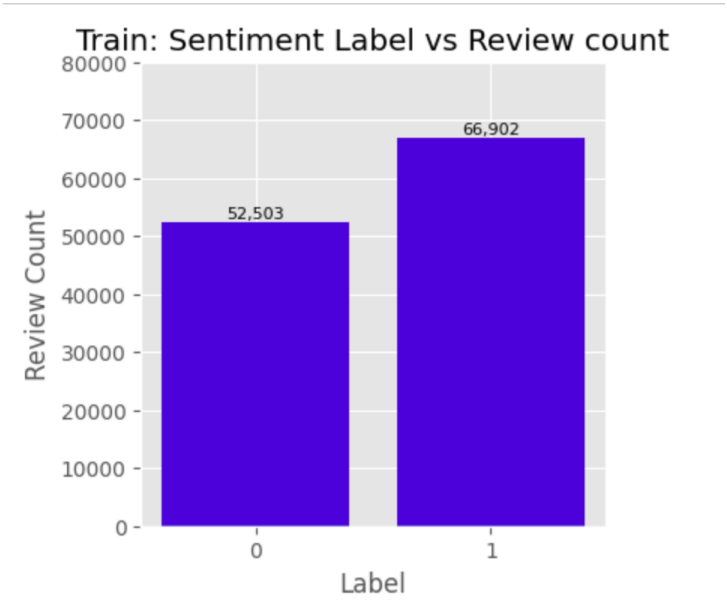
- Data pre-processing
- Machine learning models
- Confusion matrices

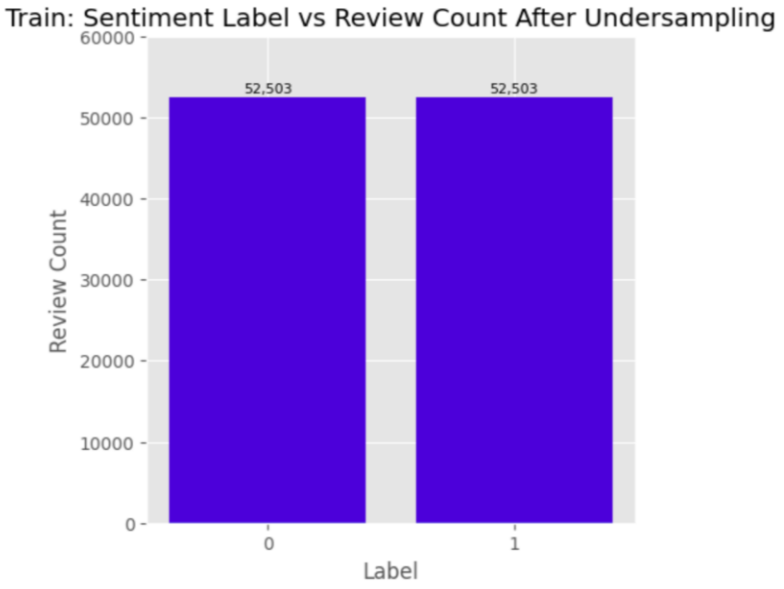
Data Pre-processing

- Deleting redundant columns
- Random resampling without replacement
- Preparaing train validation and testing data
- Standardize the ratings for sentiment analysis
- Converting words to lower case
- Tokenizing the words
- Removing the special characters and the stop words
- Lemmetizing the words
- Making random undersampling to make data balanced
- Vectorizing the features
- Normalize the data

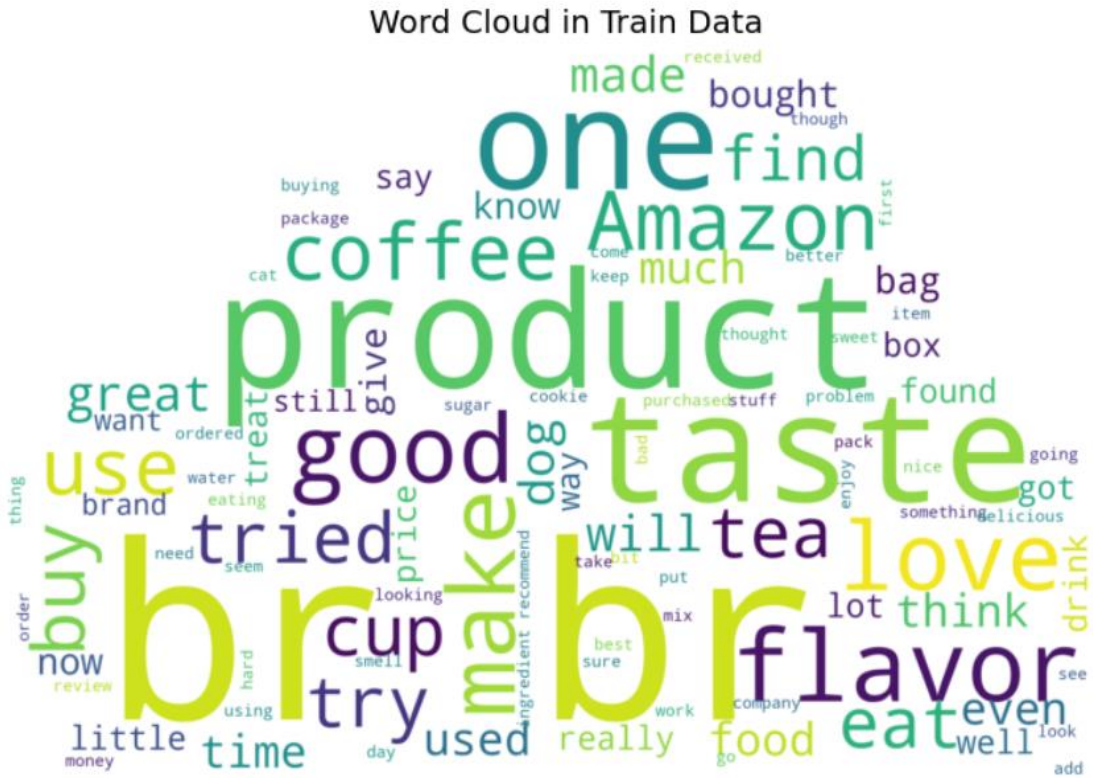
Data Preprocessing







Word Cloud


Word Cloud in Train Data
one
product
br
br
taste
flavor
love
good
make
tried
coffee
Amazon
find
made
bought
know
say
buying
package
cat
keep
much
bag
item
box
found
great
want
ordered
still
give
treat
sugar
cookie
purchased
stuff
problem
use
brand
water
eating
dog
way
pack
nice
going
got
something
delicious
buy
price
need
seem
will
tea
take
put
looking
lot
think
drink
order
now
review
smell
hard
cup
try
mix
best
sure
ingredient
recommend
see
company
work
using
little
time
money
day
used
really
food
eat
even
well
look
add
go
received
though
first
come
better
thought
sweet
bad
enjoy
thing

Machine Learning Models

- Bernoulli Naive Bayes
- Complement Naive Bayes
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest

Model Training and Evaluation

- I split the data into train and test sets. Then, I further split the training set into train and validation sets.
- Train the models using training set and tune hyperparameters using validation set.
- Model accuracies, precision scores, recall scores and F1-scores have been used to evaluate the model

Model Training and Evaluation (Cont...)

	Best Parameters
Logistic Regression:	C=5, max_iter=30, tol=0.001
Linear Support Vector Classifier	C=0.1, max_iter=30, tol=0.01
Bernoulli Naive Bayes	alpha=0.1
Complement Naive Bayes	Alpha=10
Random Forest Classifier	Criterion="gini", max_depth=150, n_estimators=200

Model Training and Evaluation (Cont...)

TF-IDF(ngram=(1,1))	Accuracy	Precision	Recall	F1-Score
Logistic Regression:	86.65%	89.36%	86.48%	87.90%
Linear Support Vector Classifier	87.29%	89.95%	87.03%	88.47%
Bernoulli Naive Bayes	83.39%	82.93%	88.59%	85.66%
Complement Naive Bayes	84.26%	87.69%	83.65%	85.62%
Random Forest Classifier	**88.63%**	89.69%	90.04%	89.87%

Model Training and Evaluation (Cont...)

TF-IDF(ngram=(1,2))	Accuracy	Precision	Recall	F1-Score
Logistic Regression:	90.69%	92.78%	90.42%	91.58%
Linear Support Vector Classifier	**91.15%**	92.84%	91.24%	92.03%
Bernoulli Naive Bayes	87.98%	85.84%	94.05%	89.76%
Complement Naive Bayes	88.57%	91.81%	87.39%	89.54%
Random Forest Classifier	88.84%	89.23%	91.07%	90.14%

Model Training and Evaluation (Cont...)

TF-IDF(ngram=(1,3))	Accuracy	Precision	Recall	F1-Score
Logistic Regression:	90.11%	92.27%	98.88%	91.06%
Linear Support Vector Classifier	**91.21%**	92.03%	91.29%	92.09%
Bernoulli Naive Bayes	88.29%	85.18%	95.76%	90.16%
Complement Naive Bayes	89.29%	92.83%	87.65%	90.16%
Random Forest Classifier	88.49%	88.51%	91.30%	89.89%

Conclusion

- In conclusion, I cleaned up and featurized an Amazon reviews dataset and built Logistic Regression, Linear Support Vector Classifier, Bernoulli Naive Bayes, Complement Naive Bayes, and Random Forest Classifier to predict sentiment (positive or negative).
- I found out that **Linear Support Vector Classifier** model got the highest accuracy (91.21) and outperformed among the proposed models.

References

- https://www.researchgate.net/publication/331291125_Sentiment_analysis_and_opinion_mining_applied_to_scientific_paper_reviews
- https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2
- https://www.researchgate.net/publication/325756171_Sentiment_analysis_on_large_scale_Amazon_product_reviews
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3886135
- https://ieeexplore.ieee.org/document/8376299