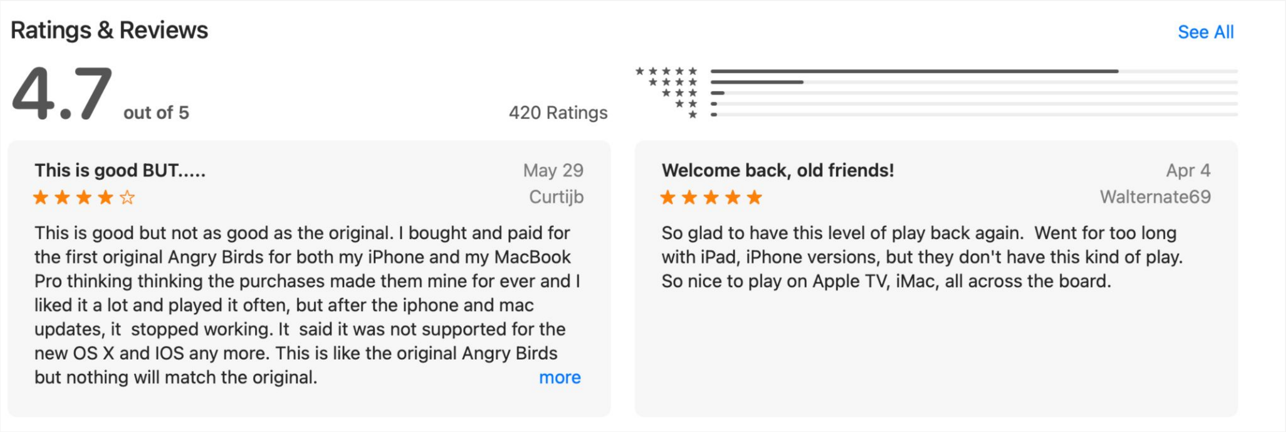
App Store
Fake Review Detection

Hande Gulbagci Dede, PhD

Online Reviews

- Online reviews are an important and inevitable aspect of e-commerce.
- Similar to online reviews, before downloading an app, users often read through the reviews.



The Problem

- Positive reviews promote the download and sales of applications.
- Some app developers mislead users by posting fake and high-rating reviews.
    - Misguide users into making wrong decisions
    - Cause economic damage to other application owners
- **Goal:** Build a classifier that can accurately classify the App Store review as genuine and fake

Data

- The Apple App Store reviews dataset is created by Martens and Maalej (2019).

Features Data

- 19 features
- 8696 unique users
- 5624 unique apps
- Review labels as fake and real

Reviews Data

- 31 features
- 16000 reviews (8000 fake & 8000 real)
- 10 features related to reviews such as review body, title, posting time

Applications Data

- 31 features
- Meta-data of apps
- 5563 unique apps
- 61 apps info is missing

Data Wrangling

Regular Steps

- Shape, Data Types, NaNs, Duplicates etc.

Dropped Columns

- Columns with just unique value
- Identical columns (i.e.contentAdvisoryRating ,trackContentRating)
- Non-imputable columns (Advisories columns with 10.7% NaNs)
- Columns have same mission (i.e. Genres, GenresId)
- Irrelevant data (i.e creator name)

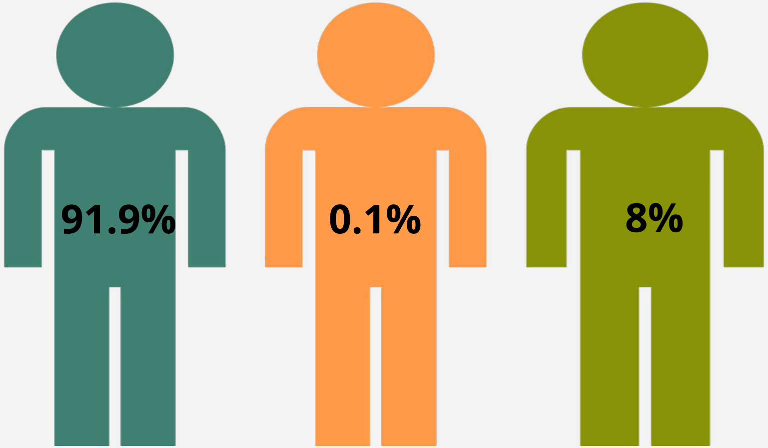
Create New Features

- Price ⇨ Paid - Free
- Version ⇨ Major version (i.e. 8.7.26 ⇨ 8)
- Datetime Objects ⇨ Year, Month, Day
- languageCodesISO2A ⇨ # of languages
- supportedDevices ⇨ # of supported devices

Merge DataFrames

- All features dataframe
- Ready for EDA!

8696 Unique Users



No fake reviews | **At least 1 fake review** | **All reviews are fake**

5624 Unique Apps

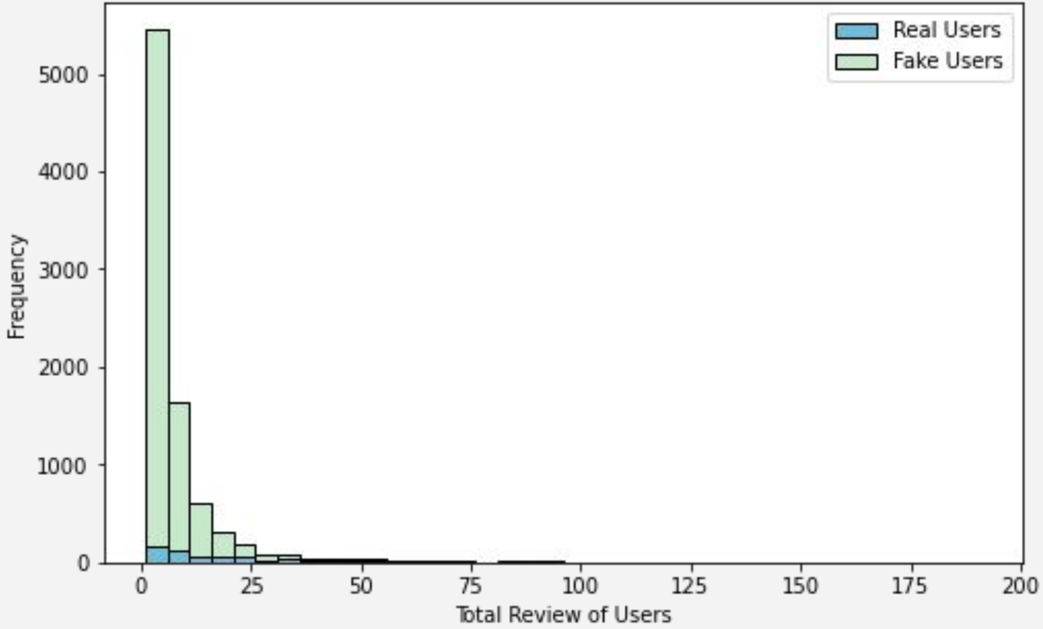
66.4% | **1.2%** | **32.4%**

No fake reviews | **At least 1 fake review** | **All reviews are fake**

Labeling criteria: If an app/user has at least one fake review, this app/user is unreliable.

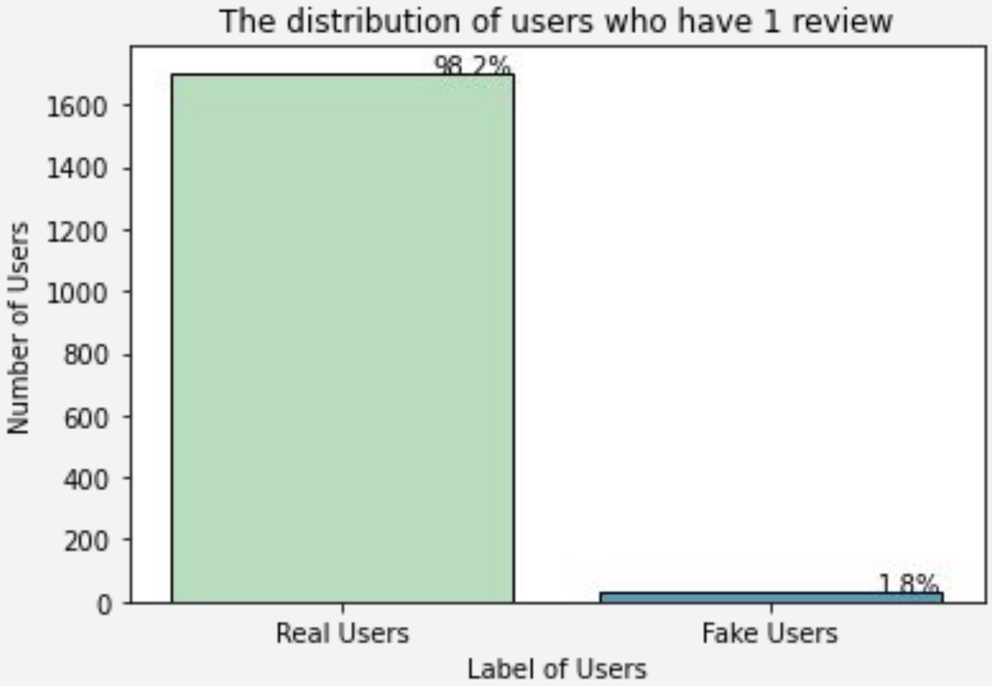
Users' Total Reviews

Fake users ($\bar{X}$ = *31.6*) have posted more reviews than real users ($\bar{X}$ = *6.85*). This difference is statistically significant (*t* = *-29.3*, *p* < *.05*).

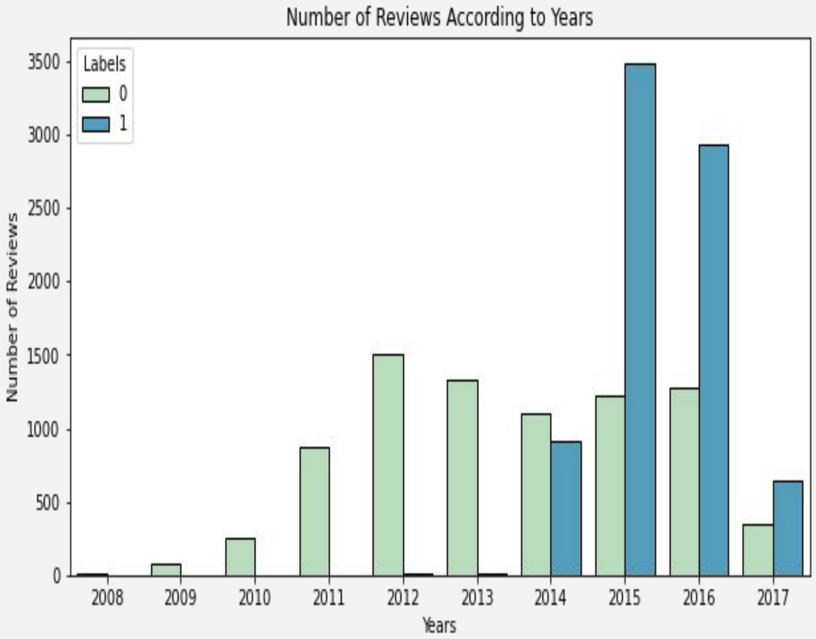


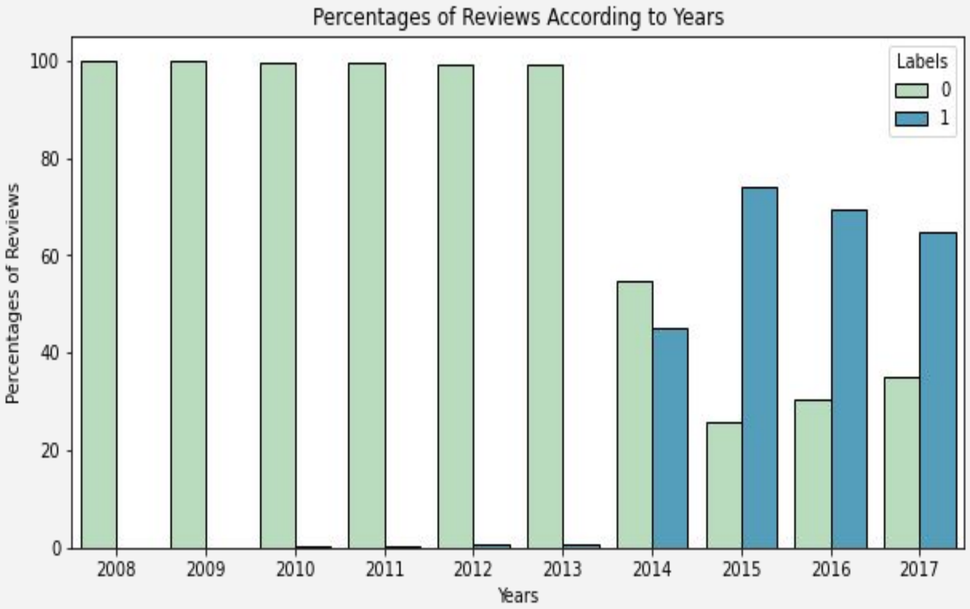
Users' with Just One Review

- 1734 users (19.9%) out of total users
- Important indicator of not being fake

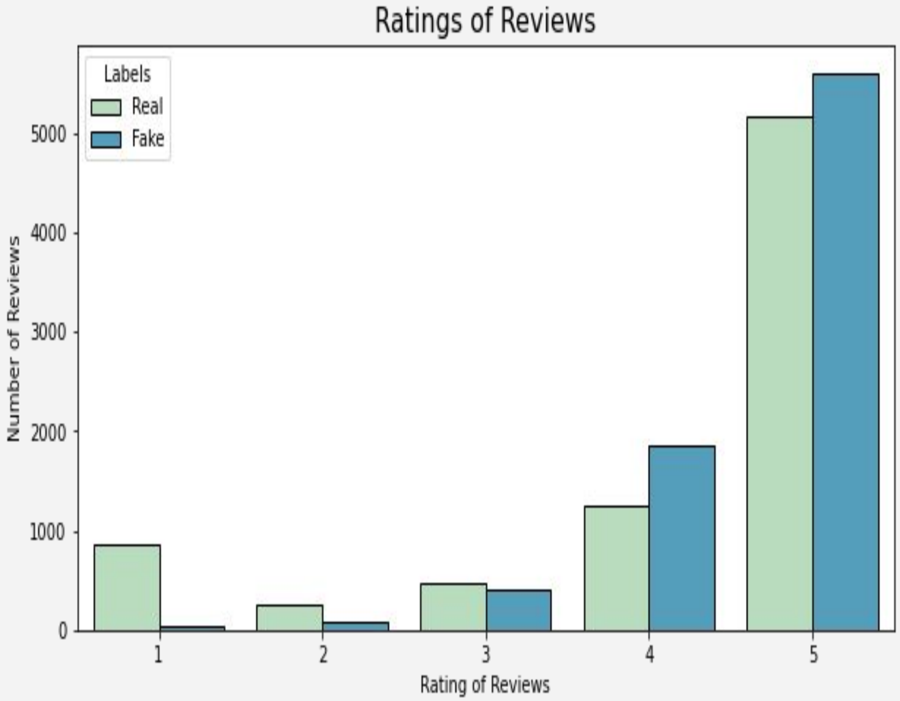


Distribution of Reviews According to Years



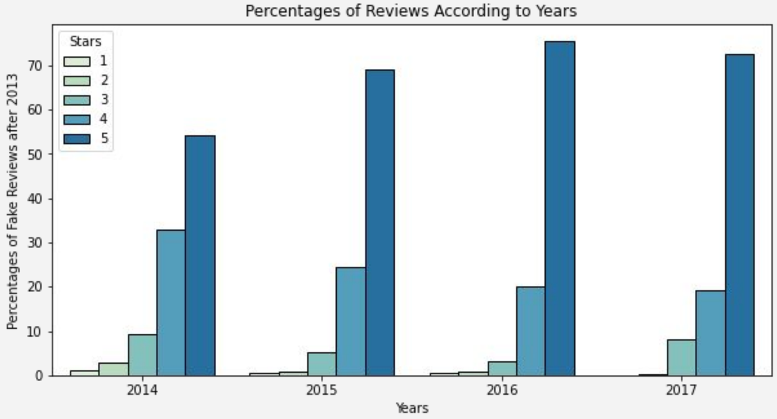


Rating of Reviews

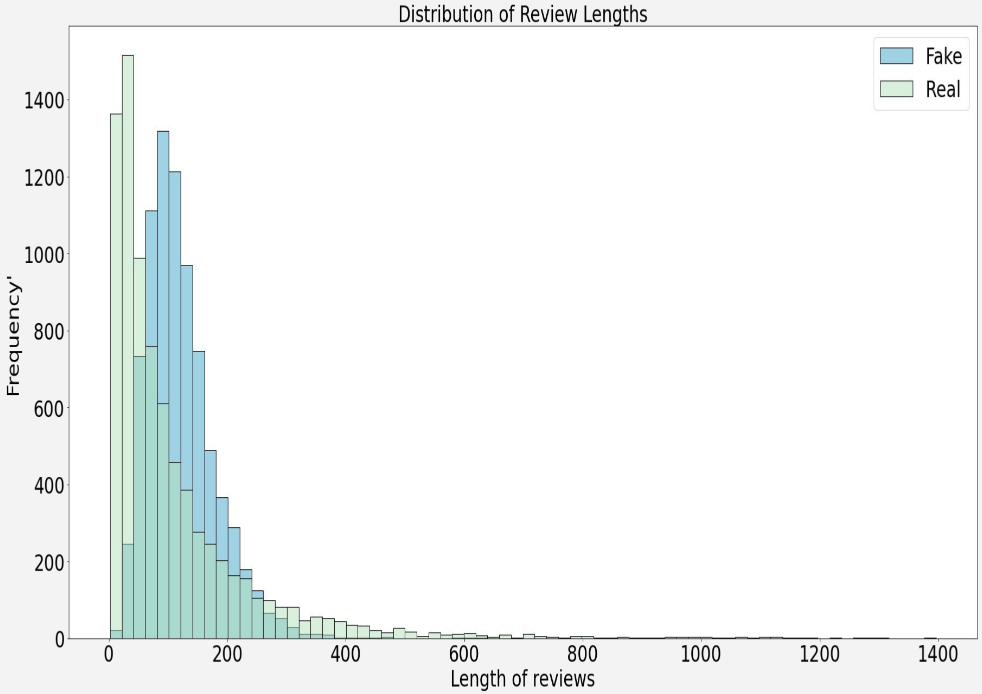


Unreliable apps can get fake reviews for two reasons:

- Promote their application, or
- Sabotage rival applications.



Length of Reviews



- The review body length is longer in fake reviews ($\bar{X}$=*121.18*) than in real ones ($\bar{X}$=*111.83*).
- 35.1% of real reviews' review length is 40 or less.
- The 3.15% of fake reviews' lengths are 40 or less.
- Fake reviews may be written longer to be more convincing, but in real life, reviews are shorter.

Language Detection

94.2% of review text,
98.7% of app descriptions
were written in English.

Meta Features

- Generate new features from text
  - # Punctuation,
  - # Words
  - # Average length of words etc.

Text Cleaning

- Convert into lowercase
- Removed items:
  - Punctuation,
  - HTML,
  - Emoji,
  - URL,
  - White spaces,
  - Text in the square brackets, and numbers

Translation

Non-English reviews and descriptions were translated into English.

Similarity between text and description

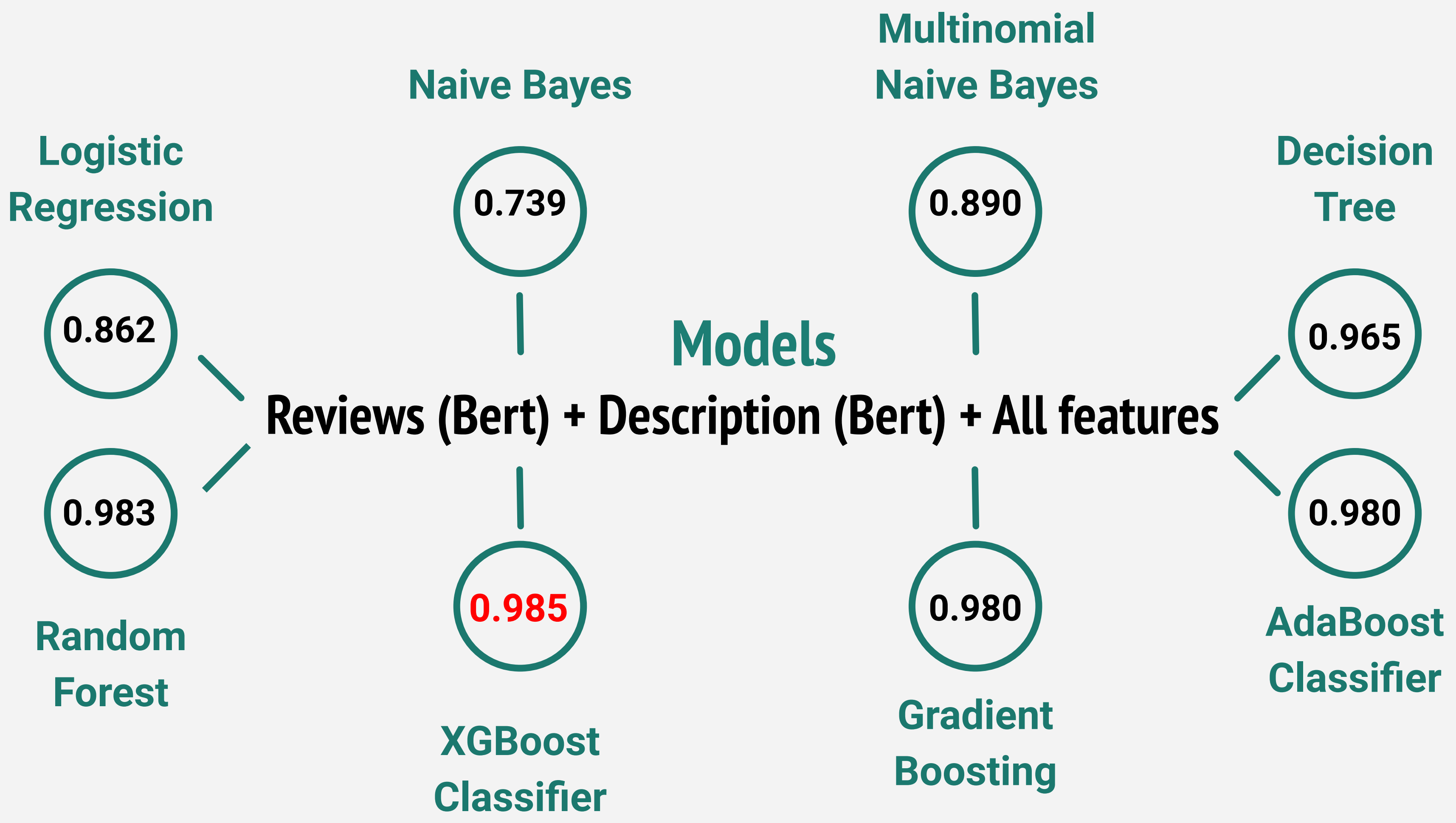
- Stop words were removed
- Tokenization
- Stemming
- TF-IDF vectorizer
- Cosine similarity

Convert into Vectors

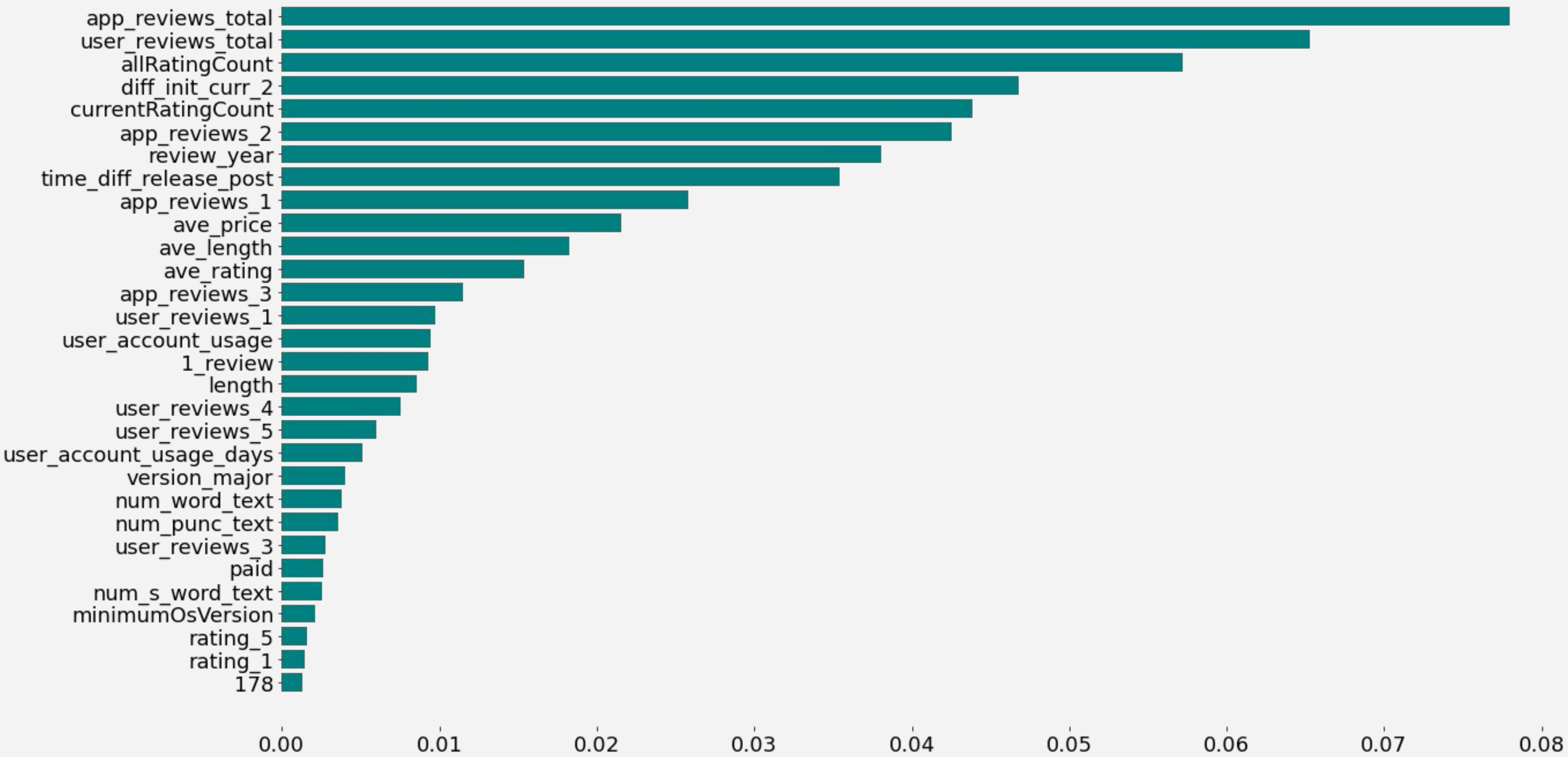
GloVe: Word vector representation method
Sentence-BERT: Context of a word in a sentence

Final Checks before Modelling

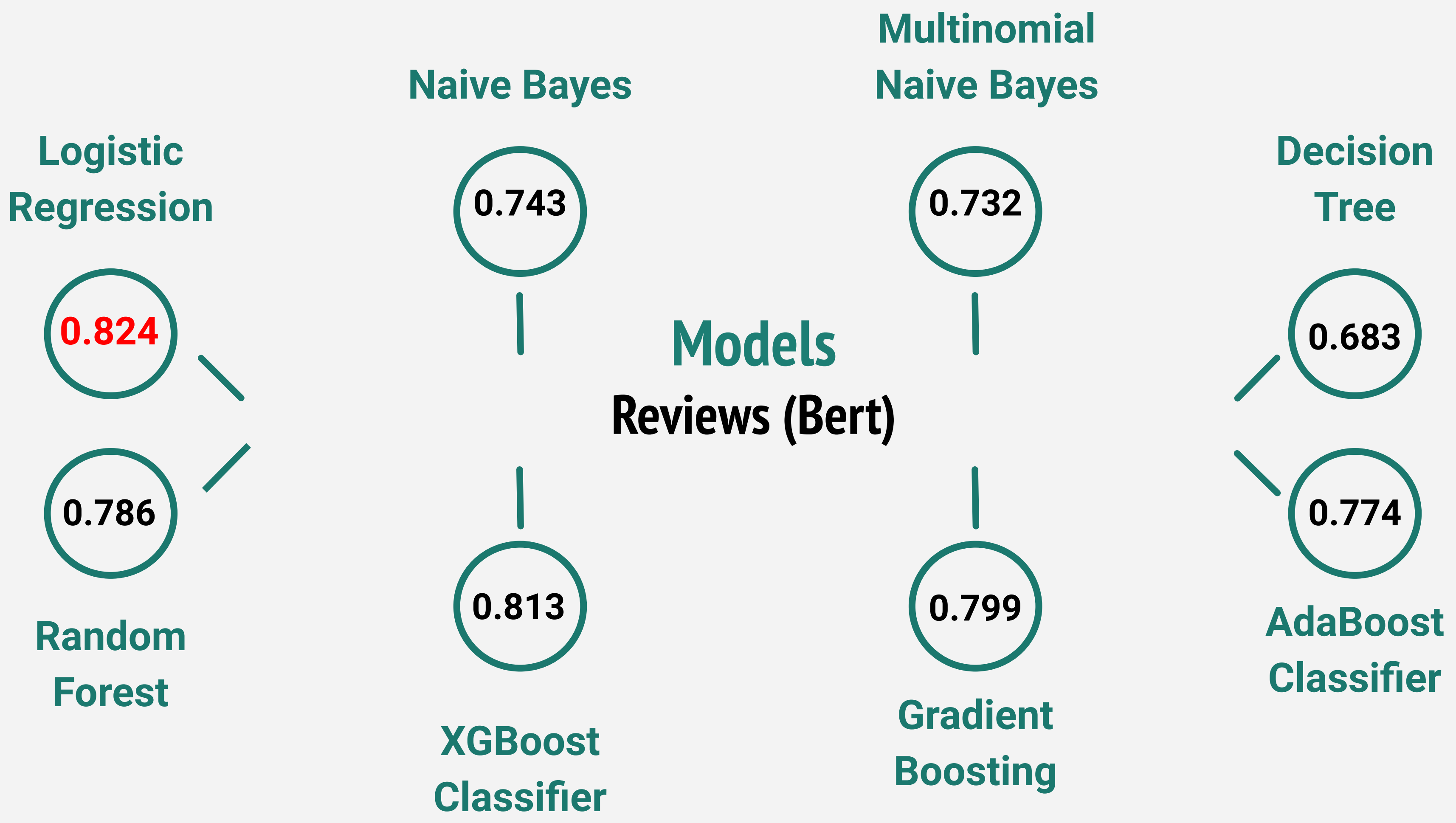
- Train-test split
- Encoding
- Imputing NaNs


Models
Reviews (Bert) + Description (Bert) + All features
Logistic Regression
0.862
Naive Bayes
0.739
Multinomial Naive Bayes
0.890
Decision Tree
0.965
Random Forest
0.983
XGBoost Classifier
0.985
Gradient Boosting
0.980
AdaBoost Classifier
0.980

Feature Importances



Random Forest with All Features - Feature Importance Plot -


Models
Reviews (Bert)
Logistic Regression
0.824
Naive Bayes
0.743
Multinomial Naive Bayes
0.732
Decision Tree
0.683
Random Forest
0.786
XGBoost Classifier
0.813
Gradient Boosting
0.799
AdaBoost Classifier
0.774

- Features related to users, reviews, and apps play a significant role in detecting fake reviews.
- Models with just review text gave worse metrics than models with all features.
- As expected, BERT works better than GloVe.
- Assuming we do not have any data about reviews, users, and apps, we have just reviews.
- Optimize models with review text (BERT).

Hypermeter Optimization

Model	Optimization Type	Mean AUC	Performance on Unseen Test
Logistic Regression	Grid Search	.901	.824
XGBoost Classifier	Random Search	.884	.811
Gradient Boosting		.879	0.811
Random Forest		.858	.784

Conclusion

- XGBoost classifier identifies fake reviews with an AUC/ROC value of 98%.
- Information about users, applications, and reviews should be considered in fake review detection.
- Factors such as the business problem, the aim of the model, and text type should be considered when deciding which vectorization method to use.

For questions, suggestions and feedback

Contact info:

handegulbagci@gmail.com

https://github.com/hangulde