

Review Sentiment Prediction

Binary Classification Problem

Prepared By: Hiren Kelaiya

Index

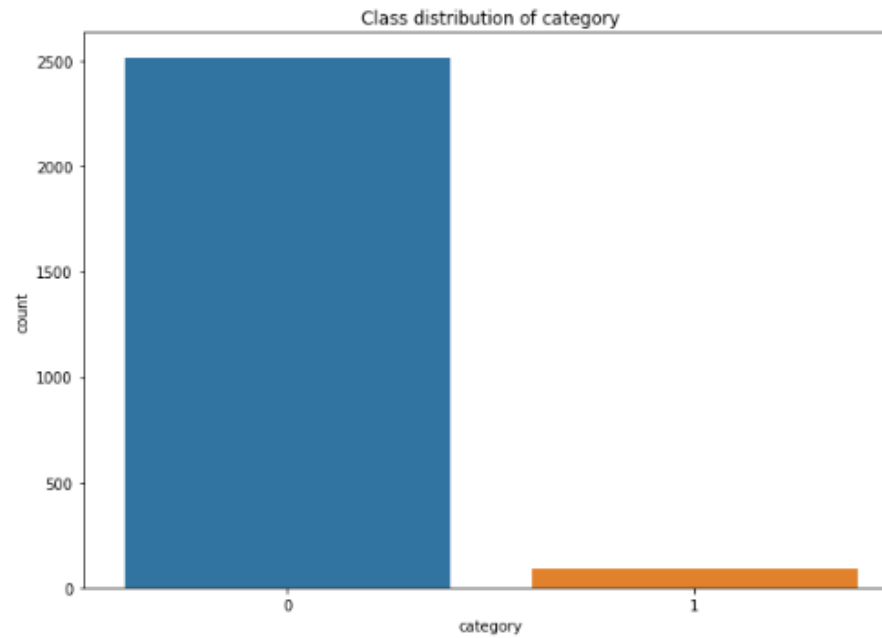
- Initial Data Observation
- Exploratory Data Analysis (EDA)
- Feature Encoding
- Class imbalance handling and Train/Test split
- Machine Learning Models
- Evaluation metric
- Conclusion

Initial Data Observation

- Training data has 3 features: **id, text and category**
- Here text is already converted into numbers but given in string format
- Category is the sentiments containing positive (1) and negative (0) classes
- Training data has 2598 entries while testing data has 866 entries
- None of the features have missing values
- Target variable has imbalanced class distribution
- Scaling is not required as all the numerical features are in similar scale

Exploratory Data Analysis (EDA)

- Distribution of Target Classes



1. Heavily class imbalance is observed. From the above figure, we can clearly see that class 1 (positive sentiment) is very less than class 0 (negative sentiment).

Feature Encoding

- Here, following 3 different NLP techniques are tried to vectorize the text feature:
 1. **Count Vectorizer** converts a collection of text documents to a matrix of token counts
 2. **TFIDF Vectorizer** converts a collection of raw documents to a matrix of TF-IDF features.
 3. **Hash Vectorizer** converts a collection of text documents to a matrix of token occurrences

Class imbalance handling and Train/Test split

- The target variable ***category*** has imbalance between two classes.
- Class **1** needs to be up-sampled in order to make it equivalent to the class **0**. **SMOTE** technique is used to handle class imbalance.
- Here, **80:20** splitting of training and validation data lead to higher ***precision score*** compare to other splits.

Machine Learning Models

- Tried following few classifiers and selected the best performing model for further tuning and testing.
 1. Logistic Regression
 2. Random Forest
 3. XGBoost
 4. Stochastic Gradient Descent (SGD) Classifier
- Out of these models, **Logistic Regression** outperformed all others with having **precision score of 0.96** on test data.

Evaluation metric

- For this problem, **precision score with micro average** is used to evaluate the model on test data.
- Precision attempts to answer the following question: What proportion of positive identifications was actually correct? The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.
- The best value is 1 and the worst value is 0.
- Precision = $TP / (TP + FP)$

Conclusion

- This is the binary classification problem with imbalanced target classes.
- The given feature ***text*** is already tokenized and converted into numeric values.
- ***id*** is removed from the data due to unimportance in predicting target variable.
- **Logistic Regression** performs better compare to other known classifiers with **Precision** of **0.96** on test data.

Thank you