

## Predicting the ratings of Coursera course reviews

### Abstract

This project will be based on binary text classification problem to predict the ratings of Coursera reviews from reviewer's text content a dataset that can be found in Kaggle. The dataset with 15000 observation and 2000 features. Divided as 80% training and 20% testing. I have used logistic regression, SVM, Random Forest and Naive Bayes. The dataset suffers from imbalanced classes which was handled by using SMOTE and ADASYN. CountVectorizer and TF-IDF have been used as word embedding. The results show in imbalanced classes, a close performance for accuracy, precision, recall and F1 score for logistic regression if the embedding was done with CountVectorizer and TF-IDF. After applying the SMOTE, CountVectorizer slightly performed better than TF-IDF. For the rest of algorithms, their performances were compared on balanced classes with SMOTE and CountVectorizer. The results between the other models show that RF outperformed the rest. Logistic regression comes next then SVM, finally Naïve bayes.

### Design

Two datasets will be used one for reviews and the other one for the courses after merging these two datasets we will have 8 features and about 1454711 instances they will only be used in EDA. In building the models I used dataset with 15000 observation and 2000 features due problems in resource. For the classification problem I will only use reviews column and I will create a new column for sentiment which will be used as label. To create the label, I used the rating to transfer the problem to binary classification. Each reviewer can rate from 1 to 5, these rates will be separated into two classes:

- **High rating (class 1):** if rating  $\geq 3$
- **Low rating (class 0):** if rating  $< 3$

## Algorithms

I have used logistic regression, SVM, Random Forest and Naive Bayes. The dataset suffers from imbalanced classes which was handled by using SMOTE and ADASYN. CountVectorizer and TF-IDF have been used as word embedding. The results showing in the tables below:

Table1: CountVectorizer Vs TF-IDF logistic regression model with imbalanced classes

	LR_CV	LR1-TFIDF
<b>Accuracy</b>	0.981	0.981
<b>Precision</b>	0.982	0.981
<b>Recall</b>	0.999	1.000
<b>F1 Score</b>	0.990	0.990

Table 2: CountVectorizerand Vs TF-IDF logistic regression model with balanced classes using SMOTE

	LRCV- balanced_class_smote	LRTFIDF- balanced_class_smote
<b>Accuracy</b>	0.910	0.884
<b>Precision</b>	0.992	0.988
<b>Recall</b>	0.916	0.892
<b>F1 Score</b>	0.952	0.938

Table 3: Total Results

	LR_CV	LR1-TFIDF	LR-balanced_ADASYN	LRCV-balanced_class_smote	LRTFIDF-balanced_class_smote	SVM_balanced_smote	RF_balaanced_smote	Bayes_balanced_smote
<b>Accuracy</b>	0.981	0.981	0.887	0.910	0.884	0.887	0.912	0.883
<b>Precision</b>	0.982	0.981	0.989	0.992	0.988	0.987	0.985	0.984
<b>Recall</b>	0.999	1.000	0.894	0.916	0.892	0.896	0.924	0.895
<b>F1 Score</b>	0.990	0.990	0.939	0.952	0.938	0.939	0.954	0.937

## Tools

- Numpy and Pandas for data manipulation
- Scikit-learn for modeling, confusion matrix and feature extraction
- Matplotlib and Seaborn for plotting
- imblearn to handle imbalanced classes

**Communication:** Slides, plotting seaborn libraries in python