

# Predicting the Ratings of Coursera Course Reviews

(NLP Problem)

Amany Mohammed Al Luhaybi

[amluhaybi@uqu.edu.sa](mailto:amluhaybi@uqu.edu.sa)

# Motivation

- This project will be based on binary text classification problem to predict the ratings of Coursera reviews from reviewer's text content.
- This project could be used as part of bigger applications. It can be included in a personal website for university's professors as external resources for their courses and it will only have the courses from Coursera with high rates.

# Preparing the Dataset

**Dataset source:** [https://www.kaggle.com/imuhammad/course-reviews-on-coursera?select=Coursera\\_reviews.csv](https://www.kaggle.com/imuhammad/course-reviews-on-coursera?select=Coursera_reviews.csv)

- Two datasets will be used one for reviews and the other one for the courses.
- Merging of these two datasets is performed resulted in 8 features and about 1454711 instances.
- Create new column for the label with name sentiment
- Drop one column that is not needed even for EDA.
- cleaning the reviews column as it will be used later: Removing punctuation, numbers, extra whitespace, stop-words and Converting text to lower case
- Basic cleaning like Handling missing data in review column with fill in it with 'good high rate', 'bad low rate' based on the rating of that user
- Removing duplicates.

# Sample of the dataset

	reviews	reviewers	date_reviews	rating	course_id	name	institution	course_url
0	Pretty dry, but I was able to pass with just t...	By Robert S	Feb 12, 2020	4	google-cbrs-cpi-training	Become a CBRS Certified Professional Installer...	Google - Spectrum Sharing	<a href="https://www.coursera.org/learn/google-cbrs-cpi...">https://www.coursera.org/learn/google-cbrs-cpi...</a>
1	would be a better experience if the video and ...	By Gabriel E R	Sep 28, 2020	4	google-cbrs-cpi-training	Become a CBRS Certified Professional Installer...	Google - Spectrum Sharing	<a href="https://www.coursera.org/learn/google-cbrs-cpi...">https://www.coursera.org/learn/google-cbrs-cpi...</a>
2	Information was perfect! The program itself wa...	By Jacob D	Apr 08, 2020	4	google-cbrs-cpi-training	Become a CBRS Certified Professional Installer...	Google - Spectrum Sharing	<a href="https://www.coursera.org/learn/google-cbrs-cpi...">https://www.coursera.org/learn/google-cbrs-cpi...</a>
3	A few grammatical mistakes on test made me do ...	By Dale B	Feb 24, 2020	4	google-cbrs-cpi-training	Become a CBRS Certified Professional Installer...	Google - Spectrum Sharing	<a href="https://www.coursera.org/learn/google-cbrs-cpi...">https://www.coursera.org/learn/google-cbrs-cpi...</a>

# EDA: The questions in MVP

## Q1: Which institution had the most courses?

```
# Which institution had the most courses?  
inst_high_Courses = df.institution.value_counts()  
inst_high_Courses|
```

```
University of Michigan  
DeepLearning.AI  
Google  
University of Pennsylvania  
IBM  
  
Google - Spectrum Sharing  
Peter the Great St. Petersburg Polytechnic University  
University of New Mexico  
GitLab  
Advancing Women in Product  
Name: institution, Length: 132, dtype: int64
```

57194

40329

26936

25763

24975

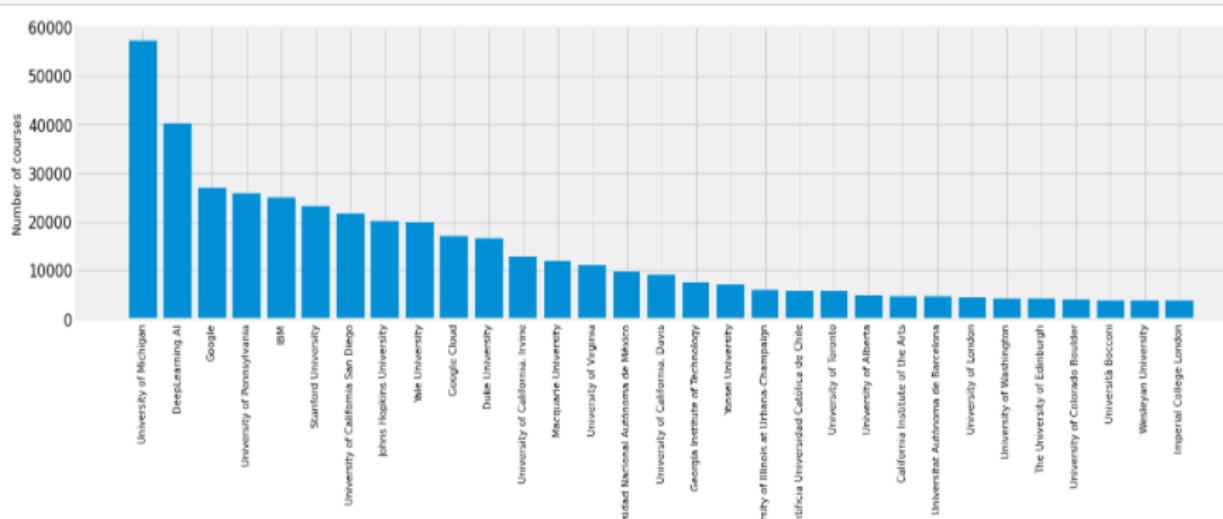
...

33

33

6

```
# Using some institution because the size of data is huge  
#we can see that University of Michigan has the hightes number of courses  
y = df.institution.value_counts().values[0:31]  
x = df.institution.value_counts().index[0:31]  
plt.figure(figsize=(20,5))  
plt.bar(x,y)  
plt.xlabel("institution")  
plt.ylabel("Number of courses")  
plt.xticks(rotation='vertical',size=10)  
plt.yticks(size=15)  
plt.show()
```



# EDA: The questions in MVP

## Q2) Which course had the highest rates by reviewers?

```
#which course had the highest rates by reviewers?  
high_rated_courses=df.groupby(['name'])['rating'].mean().sort_values(ascending = False)
```

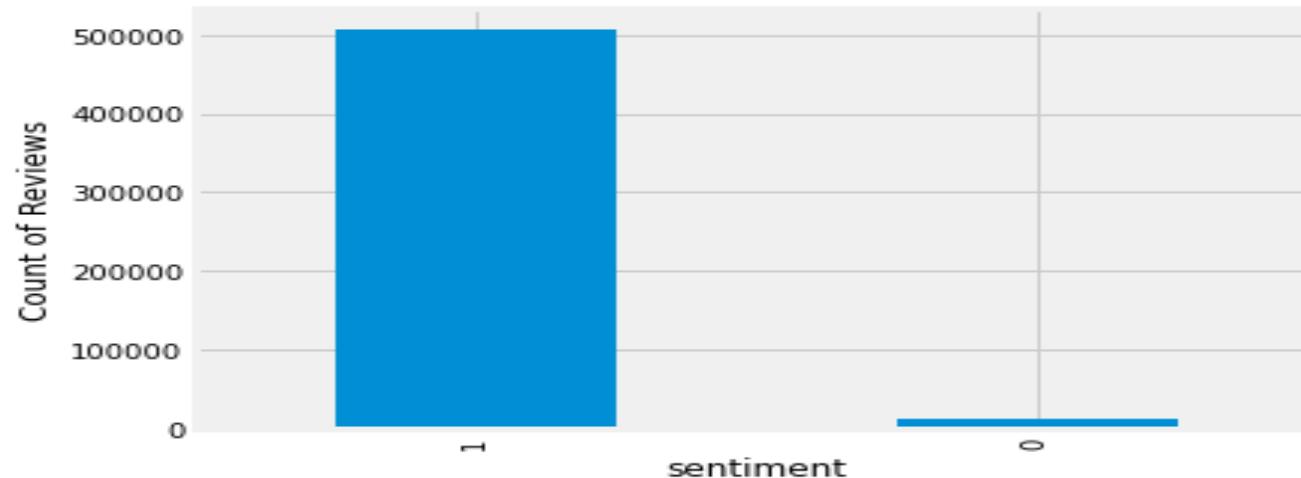
```
high_rated_courses
```

name	rating
Entreprise et changement climatique	5.000000
How to Manage a Remote Team	5.000000
Building Modern Python Applications on AWS	5.000000
Removing Barriers to Change	5.000000
El Abogado del Futuro: Legaltech y la Transformación Digital del Derecho	4.968000
	...
The Introduction to Quantum Computing	3.204545
Epigenetic Control of Gene Expression	1.000000
Entrepreneurship Strategy: From Ideation to Exit	1.000000
Social and Economic Networks: Models and Analysis	1.000000
Curanderismo: Traditional Healing Using Plants	1.000000
Name: rating, Length: 683, dtype: float64	

# Imbalanced classes

- Discovering imbalance in dataset will be dealt with in two ways

```
#discovering imbalance in dataset
ax = df.sentiment.value_counts().plot(kind='bar')
plt.xlabel("sentiment")
plt.ylabel("Count of Reviews")
# the data is Highly imbalanced will be dealt with in the final submission
Text(0, 0.5, 'Count of Reviews')
```



```
#discovering imbalance in dataset
df['sentiment'].value_counts()
1    507233
0     12695
Name: sentiment, dtype: int64
```

# Algorithms

- Due the huge size of the dataset over 1m and the limited resources, I had to reduce the size of observation to 15000 and restricted the features to 2000.
- I have used only the reviews and sentiment columns.
- I have used logistic regression, SVM, Random Forest and Naive Bayes.
- I have used CountVectorizer and TF-IDF as word embedding and Create a logistic regression model to compare their performances. And run the experiment on imbalanced and balanced classes using SMOTE classes.
- Other algorithms were compared with only balanced classes using SMOTE and CountVectorizer as word embedding.

# Results: CountVectorizer and Vs TF-IDF logistic regression model with imbalanced classes

	LR_CV	LR1-TFIDF
Accuracy	0.981	0.981
Precision	0.982	0.981
Recall	0.999	1.000
F1 Score	0.990	0.990

- The results show a close results in all the metrics for this model if the embedding was done with CountVectorizer Vs TF-IDF

# Results: CountVectorizer and Vs TF-IDF logistic regression model with balanced classes using SMOTE

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

- The results after balancing the classes shows that slightly CountVectorizer performed better than TF-IDF

# The Total results of all models

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

- The results between the other models show that RF outperformed the rest. Logistic regression comes next then SVM , finally Naïve bayes.