Assignment 8: DT

- Apply Decision Tree Classifier(Decision Tree)(sample) on these features sets:
  - Set 1: categorical, numerical features + dense(text) (BOW with binarized vectorizer)+ dense(TfIDF)
  - Set 2: categorical, numerical features + sparse (BoW with binarized vectorizer)+ sparse(TfIDF)+ Project Text proposal feature, where you replace each word in the essay with its corresponding glove vector(300 dim vector)+ Project Text proposal feature, where you replace each word in the essay with its corresponding TfIDF weighted glove vector(300 dim vector)
- Hyperparameters of model:
  - You need to plot the performance vs depth and min_samples_split(you can take any reasonable range for both parameters). We should see the trade-off

  AND (i.e.) in case **sample split** 10 you can **depth** and (i.e.) or **Set Review** can have given the maximum (which sample split is the depth, you can find it in the same plot) for **depth** and **min_samples_split**

OR

  - You need to plot the performance of model both on train data and cross validation data for each hyper parameter, like shown in the figure.

  - Along with plotting ROC curve, you need to print the confusion matrix with proper labels for train and test data.

Confusion Matrix		
	Predicted: No	Predicted: Yes
Actual: No	TN = ?	FP = ?
Actual: Yes	FN = ?	TP = ?

  - Once after you plot the confusion matrix with the best hyper parameter, you need to do the following with it:
    - Find the Most Important 20 features from different feature groups (categorical, numerical, text) and print them in the console/notebook
    - Show the pretty table format output of prettytable library or Matplotlib table format output of prettytable library
    - Once after you found the best hyper parameters, you need to train your model with it and find the AUC on test data and plot the ROC curve on test data with them.

  - Once after you found the best hyper parameters, you need to train your model with it, and find the AUC on test data and plot the ROC curve on test data with them.

  - Summarize the results in the following table:

Vectorizer	Model	Hyper parameter	AUC
BOW	DT	depth: 10, min_samples_split: 500	0.7
TfIDF	DT	depth: 10, min_samples_split: 500	0.72
AVG W2V	DT	depth: 10, min_samples_split: 500	0.74
TfIDF W2V	DT	depth: 10, min_samples_split: 500	0.76

  - Once after you plot the confusion matrix with the best hyper parameter, you need to do the following with it:
    - Find the Most Important 20 features from different feature groups (categorical, numerical, text) and print them in the console/notebook
    - Show the pretty table format output of prettytable library or Matplotlib table format output of prettytable library

Note: when you train for the Decision Tree(hyper parameter tuning) you can DO not train, plot hyperparameter tuning in future.

- You need to summarize the results you have obtained in the form of a table as shown below

Vectorizer	Model	Hyper parameter	AUC
Set 1 categorical, numerical features + BoW + TfIDF	DT	depth: 10, min_samples_split: 500	0.7
Set 2 categorical, numerical features + BoW + TfIDF + Project Text proposal feature	DT	depth: 10, min_samples_split: 500	0.72

In Conclusion

- You need to summarize the results you have obtained in the form of a table as shown below

Vectorizer	Model	Hyper parameter	AUC
BOW	DT	depth: 10, min_samples_split: 500	0.7
TfIDF	DT	depth: 10, min_samples_split: 500	0.72
AVG W2V	DT	depth: 10, min_samples_split: 500	0.74
TfIDF W2V	DT	depth: 10, min_samples_split: 500	0.76

- Your task is to analyze the TruncatedSVD for TfIDF W2V vectors(for example you can take 5000 dimensions) and then you can perform the same steps 2 to 6 as mentioned above.

In [1]:
```
%matplotlib inline
import warnings
import sqlite3
import pandas as pd
import numpy
import nltk
import string
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction import TfidfVectorizer
from sklearn.feature_extraction import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer
from nltk.stem.porter import SnowballStemmer

import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

from gensim.models.word2vec import Word2Vec
from gensim.models.keyedvectors import KeyedVectors
import pickle

from tqdm.auto import tqdm
import os

!wget -O glove_vectors.zip quit
from zipfile import ZipFile
ZipFile('glove_vectors.zip').extractall()
!ls -l glove.6B.300d.txt
!rm glove_vectors.zip
```

```
warnings.filterwarnings("ignore")
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipped /root/nltk_data/corpus/stopwords.zip.
--2020-05-23 05:54:07--  http://quit/
Resolving quit (quit)... 172.217.194.121
Connecting to quit (quit):80... failed: Connection refused.
-rw-r--r-- 1 root root 1037962819 Mar  2  2015 glove.6B.300d.txt
rm: cannot remove 'glove_vectors.zip': No such file or directory
```

In [2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted; to force remount, run drive.mount("/content/drive", force_remount=True).
```

In [3]:
```
!ls -l /content/drive/My\ Drive/assignments/SVM/data/
```

```
total 9604
-rwx------ 1 root root 1053183488 Jan  1  2020 'Copy of train.csv'
-rwx------ 1 root root 1053183488 Jan  1  2020 train.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stemmed.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stemmed_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stemmed_lemmatized_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_lemmatized_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed_lemmatized_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords_stemmed.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords_stemmed_lemmatized.csv
```

In [4]:
```
!cp -r /content/drive/My\ Drive/assignments/SVM/data/ /content/
```

In [5]:
```
!ls -l /content/data/
```

```
total 9604
-rwx------ 1 root root 1053183488 Jan  1  2020 'Copy of train.csv'
-rwx------ 1 root root 1053183488 Jan  1  2020 train.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stemmed.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stemmed_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stemmed_lemmatized_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_cleaned_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_lemmatized_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed_lemmatized_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stemmed_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords_lemmatized.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords_stemmed.csv
-rwx------ 1 root root  109680128 Jan  1  2020 train_data_stopwords_stemmed_lemmatized.csv
```

In [6]:
```
!ls -l /content/data/train.csv
```

```
-rwx------ 1 root root 1053183488 Jan  1  2020 /content/data/train.csv
```

In [7]:
```
!ls -l /content/data/train_data.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data.csv
```

In [8]:
```
!ls -l /content/data/train_data_cleaned.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned.csv
```

In [9]:
```
!ls -l /content/data/train_data_cleaned_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed.csv
```

In [10]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

In [11]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

In [12]:
```
!ls -l /content/data/train_data_cleaned_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stopwords.csv
```

In [13]:
```
!ls -l /content/data/train_data_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized.csv
```

In [14]:
```
!ls -l /content/data/train_data_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized_stopwords.csv
```

In [15]:
```
!ls -l /content/data/train_data_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed.csv
```

In [16]:
```
!ls -l /content/data/train_data_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized.csv
```

In [17]:
```
!ls -l /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

In [18]:
```
!ls -l /content/data/train_data_stemmed_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_stopwords.csv
```

In [19]:
```
!ls -l /content/data/train_data_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords.csv
```

In [20]:
```
!ls -l /content/data/train_data_stopwords_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_lemmatized.csv
```

In [21]:
```
!ls -l /content/data/train_data_stopwords_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed.csv
```

In [22]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

In [23]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

In [24]:
```
!ls -l glove.6B.300d.txt
```

```
-rwx------ 1 root root 1037962819 Mar  2  2015 glove.6B.300d.txt
```

In [25]:
```
!ls -l glove_vectors
```

```
ls: cannot access 'glove_vectors': No such file or directory
```

In [26]:
```
!ls -l glove_vectors.zip
```

```
ls: cannot access 'glove_vectors.zip': No such file or directory
```

In [27]:
```
!ls -l /content/data/train.csv
```

```
-rwx------ 1 root root 1053183488 Jan  1  2020 /content/data/train.csv
```

In [28]:
```
!ls -l /content/data/train_data.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data.csv
```

In [29]:
```
!ls -l /content/data/train_data_cleaned.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned.csv
```

In [30]:
```
!ls -l /content/data/train_data_cleaned_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed.csv
```

In [31]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

In [32]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

In [33]:
```
!ls -l /content/data/train_data_cleaned_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stopwords.csv
```

In [34]:
```
!ls -l /content/data/train_data_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized.csv
```

In [35]:
```
!ls -l /content/data/train_data_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized_stopwords.csv
```

In [36]:
```
!ls -l /content/data/train_data_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed.csv
```

In [37]:
```
!ls -l /content/data/train_data_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized.csv
```

In [38]:
```
!ls -l /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

In [39]:
```
!ls -l /content/data/train_data_stemmed_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_stopwords.csv
```

In [40]:
```
!ls -l /content/data/train_data_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords.csv
```

In [41]:
```
!ls -l /content/data/train_data_stopwords_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_lemmatized.csv
```

In [42]:
```
!ls -l /content/data/train_data_stopwords_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed.csv
```

In [43]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

In [44]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

In [45]:
```
!ls -l glove.6B.300d.txt
```

```
-rwx------ 1 root root 1037962819 Mar  2  2015 glove.6B.300d.txt
```

In [46]:
```
!ls -l glove_vectors
```

```
ls: cannot access 'glove_vectors': No such file or directory
```

In [47]:
```
!ls -l glove_vectors.zip
```

```
ls: cannot access 'glove_vectors.zip': No such file or directory
```

In [48]:
```
!ls -l /content/data/train.csv
```

```
-rwx------ 1 root root 1053183488 Jan  1  2020 /content/data/train.csv
```

In [49]:
```
!ls -l /content/data/train_data.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data.csv
```

In [50]:
```
!ls -l /content/data/train_data_cleaned.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned.csv
```

In [51]:
```
!ls -l /content/data/train_data_cleaned_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed.csv
```

In [52]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

In [53]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

In [54]:
```
!ls -l /content/data/train_data_cleaned_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stopwords.csv
```

In [55]:
```
!ls -l /content/data/train_data_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized.csv
```

In [56]:
```
!ls -l /content/data/train_data_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized_stopwords.csv
```

In [57]:
```
!ls -l /content/data/train_data_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed.csv
```

In [58]:
```
!ls -l /content/data/train_data_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized.csv
```

In [59]:
```
!ls -l /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

In [60]:
```
!ls -l /content/data/train_data_stemmed_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_stopwords.csv
```

In [61]:
```
!ls -l /content/data/train_data_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords.csv
```

In [62]:
```
!ls -l /content/data/train_data_stopwords_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_lemmatized.csv
```

In [63]:
```
!ls -l /content/data/train_data_stopwords_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed.csv
```

In [64]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

In [65]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

In [66]:
```
!ls -l glove.6B.300d.txt
```

```
-rwx------ 1 root root 1037962819 Mar  2  2015 glove.6B.300d.txt
```

In [67]:
```
!ls -l glove_vectors
```

```
ls: cannot access 'glove_vectors': No such file or directory
```

In [68]:
```
!ls -l glove_vectors.zip
```

```
ls: cannot access 'glove_vectors.zip': No such file or directory
```

In [69]:
```
!ls -l /content/data/train.csv
```

```
-rwx------ 1 root root 1053183488 Jan  1  2020 /content/data/train.csv
```

In [70]:
```
!ls -l /content/data/train_data.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data.csv
```

In [71]:
```
!ls -l /content/data/train_data_cleaned.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned.csv
```

In [72]:
```
!ls -l /content/data/train_data_cleaned_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed.csv
```

In [73]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

In [74]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

In [75]:
```
!ls -l /content/data/train_data_cleaned_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stopwords.csv
```

In [76]:
```
!ls -l /content/data/train_data_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized.csv
```

In [77]:
```
!ls -l /content/data/train_data_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized_stopwords.csv
```

In [78]:
```
!ls -l /content/data/train_data_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed.csv
```

In [79]:
```
!ls -l /content/data/train_data_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized.csv
```

In [80]:
```
!ls -l /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

In [81]:
```
!ls -l /content/data/train_data_stemmed_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_stopwords.csv
```

In [82]:
```
!ls -l /content/data/train_data_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords.csv
```

In [83]:
```
!ls -l /content/data/train_data_stopwords_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_lemmatized.csv
```

In [84]:
```
!ls -l /content/data/train_data_stopwords_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed.csv
```

In [85]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

In [86]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

In [87]:
```
!ls -l glove.6B.300d.txt
```

```
-rwx------ 1 root root 1037962819 Mar  2  2015 glove.6B.300d.txt
```

In [88]:
```
!ls -l glove_vectors
```

```
ls: cannot access 'glove_vectors': No such file or directory
```

In [89]:
```
!ls -l glove_vectors.zip
```

```
ls: cannot access 'glove_vectors.zip': No such file or directory
```

In [90]:
```
!ls -l /content/data/train.csv
```

```
-rwx------ 1 root root 1053183488 Jan  1  2020 /content/data/train.csv
```

In [91]:
```
!ls -l /content/data/train_data.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data.csv
```

In [92]:
```
!ls -l /content/data/train_data_cleaned.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned.csv
```

In [93]:
```
!ls -l /content/data/train_data_cleaned_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed.csv
```

In [94]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

In [95]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

In [96]:
```
!ls -l /content/data/train_data_cleaned_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stopwords.csv
```

In [97]:
```
!ls -l /content/data/train_data_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized.csv
```

In [98]:
```
!ls -l /content/data/train_data_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized_stopwords.csv
```

In [99]:
```
!ls -l /content/data/train_data_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed.csv
```

In [100]:
```
!ls -l /content/data/train_data_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized.csv
```

In [101]:
```
!ls -l /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

In [102]:
```
!ls -l /content/data/train_data_stemmed_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_stopwords.csv
```

In [103]:
```
!ls -l /content/data/train_data_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords.csv
```

In [104]:
```
!ls -l /content/data/train_data_stopwords_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_lemmatized.csv
```

In [105]:
```
!ls -l /content/data/train_data_stopwords_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed.csv
```

In [106]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

In [107]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

In [108]:
```
!ls -l glove.6B.300d.txt
```

```
-rwx------ 1 root root 1037962819 Mar  2  2015 glove.6B.300d.txt
```

In [109]:
```
!ls -l glove_vectors
```

```
ls: cannot access 'glove_vectors': No such file or directory
```

In [110]:
```
!ls -l glove_vectors.zip
```

```
ls: cannot access 'glove_vectors.zip': No such file or directory
```

In [111]:
```
!ls -l /content/data/train.csv
```

```
-rwx------ 1 root root 1053183488 Jan  1  2020 /content/data/train.csv
```

In [112]:
```
!ls -l /content/data/train_data.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data.csv
```

In [113]:
```
!ls -l /content/data/train_data_cleaned.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned.csv
```

In [114]:
```
!ls -l /content/data/train_data_cleaned_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed.csv
```

In [115]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

In [116]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

In [117]:
```
!ls -l /content/data/train_data_cleaned_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stopwords.csv
```

In [118]:
```
!ls -l /content/data/train_data_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized.csv
```

In [119]:
```
!ls -l /content/data/train_data_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized_stopwords.csv
```

In [120]:
```
!ls -l /content/data/train_data_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed.csv
```

In [121]:
```
!ls -l /content/data/train_data_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized.csv
```

In [122]:
```
!ls -l /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

In [123]:
```
!ls -l /content/data/train_data_stemmed_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_stopwords.csv
```

In [124]:
```
!ls -l /content/data/train_data_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords.csv
```

In [125]:
```
!ls -l /content/data/train_data_stopwords_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_lemmatized.csv
```

In [126]:
```
!ls -l /content/data/train_data_stopwords_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed.csv
```

In [127]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized.csv
```

In [128]:
```
!ls -l /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords_stemmed_lemmatized_stopwords.csv
```

In [129]:
```
!ls -l glove.6B.300d.txt
```

```
-rwx------ 1 root root 1037962819 Mar  2  2015 glove.6B.300d.txt
```

In [130]:
```
!ls -l glove_vectors
```

```
ls: cannot access 'glove_vectors': No such file or directory
```

In [131]:
```
!ls -l glove_vectors.zip
```

```
ls: cannot access 'glove_vectors.zip': No such file or directory
```

In [132]:
```
!ls -l /content/data/train.csv
```

```
-rwx------ 1 root root 1053183488 Jan  1  2020 /content/data/train.csv
```

In [133]:
```
!ls -l /content/data/train_data.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data.csv
```

In [134]:
```
!ls -l /content/data/train_data_cleaned.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned.csv
```

In [135]:
```
!ls -l /content/data/train_data_cleaned_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed.csv
```

In [136]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized.csv
```

In [137]:
```
!ls -l /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stemmed_lemmatized_stopwords.csv
```

In [138]:
```
!ls -l /content/data/train_data_cleaned_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_cleaned_stopwords.csv
```

In [139]:
```
!ls -l /content/data/train_data_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized.csv
```

In [140]:
```
!ls -l /content/data/train_data_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_lemmatized_stopwords.csv
```

In [141]:
```
!ls -l /content/data/train_data_stemmed.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed.csv
```

In [142]:
```
!ls -l /content/data/train_data_stemmed_lemmatized.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized.csv
```

In [143]:
```
!ls -l /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_lemmatized_stopwords.csv
```

In [144]:
```
!ls -l /content/data/train_data_stemmed_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stemmed_stopwords.csv
```

In [145]:
```
!ls -l /content/data/train_data_stopwords.csv
```

```
-rwx------ 1 root root 109680128 Jan  1  2020 /content/data/train_data_stopwords.csv
```

In [146]:
```
!ls -l /content/data/train_data_stopwords_lemmatized.csv
```

```
-rwx
```

```
In [14]: print('school_teacher_prefix').value_counts()
```

```
Out[14]: Mrs.    38955
         Ms.     28854
         Mr.     10648
         Teacher  2360
         Dr.        13
         Name:    5942, dtype: int64
```

```
In [15]: print('school_teacher_prefix').isnull().sum()
```

```
Out[15]: 0
```

```
In [16]: print('school_teacher_prefix').fillna('Mrs.').value_counts()
```

```
Out[16]: Mrs.    38955
         Ms.     28854
         Mr.     10648
         Teacher  2360
         Dr.        13
         Name:    5942, dtype: int64
```

Preprocessing categorical features : school_state

```
In [17]: print('school_state').value_counts()
```

```
Out[17]: CA    15388
         NY     8828
         TX     7274
         FL     6761
         MN     5175
         IN     5091
         MI     5023
         OH     4954
         PA     4916
         IL     4875
         GA     4635
         WA     4350
         WI     4340
         NC     4185
         MA     3960
         AZ     3867
         CO     3745
         SC     3171
         OR     2960
         AL     2634
         NV     2489
         UT     2405
         MO     2397
         NJ     2342
         KY     2304
         TN     2300
         LA     2295
         VA     2161
         AR     1628
         MD     1526
         OK     1521
         CT     1324
         UT     1142
         NM     1057
         MS     1055
         HI     1051
         ME     1004
         KS     995
         NE     961
         ID     892
         DC     814
         WV     797
         NH     764
         AK     731
         DE     570
         RI     507
         MT     491
         SD     466
         ND     392
         VT     356
         WY     353
         Name: school_state, dtype: int64
```

```
In [18]: print('school_state').isnull().sum()
```

```
Out[18]: 0
```

```
In [19]: print('school_state').fillna('CA').value_counts()
```

```
Out[19]: CA    15388
         NY     8828
         TX     7274
         FL     6761
         MN     5175
         IN     5091
         MI     5023
         OH     4954
         PA     4916
         IL     4875
         GA     4635
         WA     4350
         WI     4340
         NC     4185
         MA     3960
         AZ     3867
         CO     3745
         SC     3171
         OR     2960
         AL     2634
         NV     2489
         UT     2405
         MO     2397
         NJ     2342
         KY     2304
         TN     2300
         LA     2295
         VA     2161
         AR     1628
         MD     1526
         OK     1521
         CT     1324
         UT     1142
         NM     1057
         MS     1055
         HI     1051
         ME     1004
         KS     995
         NE     961
         ID     892
         DC     814
         WV     797
         NH     764
         AK     731
         DE     570
         RI     507
         MT     491
         SD     466
         ND     392
         VT     356
         WY     353
         Name: school_state, dtype: int64
```

```
In [20]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[20]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [21]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[21]: 1464073
```

```
In [22]: resource_data_gb.head(2)
```

```
Out[22]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [23]: print('school_state').isnull().sum()
```

```
Out[23]: 0
```

3 Preparing data for models

```
In [24]: print('project_grade_category').value_counts()
```

```
Out[24]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtype: int64
```

```
In [25]: print('project_grade_category').isnull().sum()
```

```
Out[25]: 0
```

```
In [26]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[26]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [27]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[27]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [28]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[28]: 1464073
```

```
In [29]: resource_data_gb.head(2)
```

```
Out[29]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [30]: print('school_state').isnull().sum()
```

```
Out[30]: 0
```

3 Preparing data for models

```
In [31]: print('project_grade_category').value_counts()
```

```
Out[31]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [32]: print('project_grade_category').isnull().sum()
```

```
Out[32]: 0
```

```
In [33]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[33]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [34]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[34]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [35]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[35]: 1464073
```

```
In [36]: resource_data_gb.head(2)
```

```
Out[36]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [37]: print('school_state').isnull().sum()
```

```
Out[37]: 0
```

```
In [38]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[38]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [39]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[39]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [40]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[40]: 1464073
```

```
In [41]: resource_data_gb.head(2)
```

```
Out[41]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [42]: print('school_state').isnull().sum()
```

```
Out[42]: 0
```

```
In [43]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[43]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [44]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[44]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [45]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[45]: 1464073
```

```
In [46]: resource_data_gb.head(2)
```

```
Out[46]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [47]: print('school_state').isnull().sum()
```

```
Out[47]: 0
```

```
In [48]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[48]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [49]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[49]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [50]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[50]: 1464073
```

```
In [51]: resource_data_gb.head(2)
```

```
Out[51]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [52]: print('school_state').isnull().sum()
```

```
Out[52]: 0
```

```
In [53]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[53]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [54]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[54]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [55]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[55]: 1464073
```

```
In [56]: resource_data_gb.head(2)
```

```
Out[56]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [57]: print('school_state').isnull().sum()
```

```
Out[57]: 0
```

```
In [58]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[58]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [59]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[59]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [60]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[60]: 1464073
```

```
In [61]: resource_data_gb.head(2)
```

```
Out[61]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [62]: print('school_state').isnull().sum()
```

```
Out[62]: 0
```

```
In [63]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[63]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [64]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[64]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [65]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[65]: 1464073
```

```
In [66]: resource_data_gb.head(2)
```

```
Out[66]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [67]: print('school_state').isnull().sum()
```

```
Out[67]: 0
```

```
In [68]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[68]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [69]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[69]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [70]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[70]: 1464073
```

```
In [71]: resource_data_gb.head(2)
```

```
Out[71]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [72]: print('school_state').isnull().sum()
```

```
Out[72]: 0
```

```
In [73]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[73]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [74]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[74]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [75]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[75]: 1464073
```

```
In [76]: resource_data_gb.head(2)
```

```
Out[76]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [77]: print('school_state').isnull().sum()
```

```
Out[77]: 0
```

```
In [78]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[78]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [79]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[79]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [80]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[80]: 1464073
```

```
In [81]: resource_data_gb.head(2)
```

```
Out[81]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [82]: print('school_state').isnull().sum()
```

```
Out[82]: 0
```

```
In [83]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[83]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [84]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[84]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [85]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[85]: 1464073
```

```
In [86]: resource_data_gb.head(2)
```

```
Out[86]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [87]: print('school_state').isnull().sum()
```

```
Out[87]: 0
```

```
In [88]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[88]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [89]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[89]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [90]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[90]: 1464073
```

```
In [91]: resource_data_gb.head(2)
```

```
Out[91]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [92]: print('school_state').isnull().sum()
```

```
Out[92]: 0
```

```
In [93]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[93]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [94]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[94]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [95]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[95]: 1464073
```

```
In [96]: resource_data_gb.head(2)
```

```
Out[96]:                price  quantity
         id
         p000000   459.56         7
         p000001   515.89        21
```

```
In [97]: print('school_state').isnull().sum()
```

```
Out[97]: 0
```

```
In [98]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[98]: Grades PreK-2    44232
         Grades 3-5     37137
         Grades 6-8     16923
         Grades 9-12    10963
         Name:    5942, dtyp: int64
```

```
In [99]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[99]:   id                            description quantity  price
           0  p233245  Educational technology for classrooms        2  149.00
           1  p069063             Mobile classroom storage        1   99.67
```

```
In [100]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[100]: 1464073
```

```
In [101]: resource_data_gb.head(2)
```

```
Out[101]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [102]: print('school_state').isnull().sum()
```

```
Out[102]: 0
```

```
In [103]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[103]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [104]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[104]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [105]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[105]: 1464073
```

```
In [106]: resource_data_gb.head(2)
```

```
Out[106]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [107]: print('school_state').isnull().sum()
```

```
Out[107]: 0
```

```
In [108]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[108]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [109]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[109]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [110]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[110]: 1464073
```

```
In [111]: resource_data_gb.head(2)
```

```
Out[111]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [112]: print('school_state').isnull().sum()
```

```
Out[112]: 0
```

```
In [113]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[113]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [114]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[114]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [115]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[115]: 1464073
```

```
In [116]: resource_data_gb.head(2)
```

```
Out[116]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [117]: print('school_state').isnull().sum()
```

```
Out[117]: 0
```

```
In [118]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[118]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [119]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[119]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [120]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[120]: 1464073
```

```
In [121]: resource_data_gb.head(2)
```

```
Out[121]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [122]: print('school_state').isnull().sum()
```

```
Out[122]: 0
```

```
In [123]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[123]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [124]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[124]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [125]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[125]: 1464073
```

```
In [126]: resource_data_gb.head(2)
```

```
Out[126]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [127]: print('school_state').isnull().sum()
```

```
Out[127]: 0
```

```
In [128]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[128]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [129]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[129]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [130]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[130]: 1464073
```

```
In [131]: resource_data_gb.head(2)
```

```
Out[131]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [132]: print('school_state').isnull().sum()
```

```
Out[132]: 0
```

```
In [133]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[133]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [134]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[134]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [135]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[135]: 1464073
```

```
In [136]: resource_data_gb.head(2)
```

```
Out[136]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [137]: print('school_state').isnull().sum()
```

```
Out[137]: 0
```

```
In [138]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[138]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [139]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[139]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [140]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[140]: 1464073
```

```
In [141]: resource_data_gb.head(2)
```

```
Out[141]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [142]: print('school_state').isnull().sum()
```

```
Out[142]: 0
```

```
In [143]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[143]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [144]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[144]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [145]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[145]: 1464073
```

```
In [146]: resource_data_gb.head(2)
```

```
Out[146]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [147]: print('school_state').isnull().sum()
```

```
Out[147]: 0
```

```
In [148]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[148]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [149]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[149]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [150]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[150]: 1464073
```

```
In [151]: resource_data_gb.head(2)
```

```
Out[151]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [152]: print('school_state').isnull().sum()
```

```
Out[152]: 0
```

```
In [153]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[153]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [154]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[154]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [155]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[155]: 1464073
```

```
In [156]: resource_data_gb.head(2)
```

```
Out[156]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [157]: print('school_state').isnull().sum()
```

```
Out[157]: 0
```

```
In [158]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[158]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [159]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[159]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [160]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[160]: 1464073
```

```
In [161]: resource_data_gb.head(2)
```

```
Out[161]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [162]: print('school_state').isnull().sum()
```

```
Out[162]: 0
```

```
In [163]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[163]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [164]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[164]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [165]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[165]: 1464073
```

```
In [166]: resource_data_gb.head(2)
```

```
Out[166]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [167]: print('school_state').isnull().sum()
```

```
Out[167]: 0
```

```
In [168]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[168]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [169]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[169]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [170]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[170]: 1464073
```

```
In [171]: resource_data_gb.head(2)
```

```
Out[171]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [172]: print('school_state').isnull().sum()
```

```
Out[172]: 0
```

```
In [173]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[173]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [174]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[174]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [175]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[175]: 1464073
```

```
In [176]: resource_data_gb.head(2)
```

```
Out[176]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [177]: print('school_state').isnull().sum()
```

```
Out[177]: 0
```

```
In [178]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[178]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [179]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[179]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [180]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[180]: 1464073
```

```
In [181]: resource_data_gb.head(2)
```

```
Out[181]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [182]: print('school_state').isnull().sum()
```

```
Out[182]: 0
```

```
In [183]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[183]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [184]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[184]:   id                            description quantity  price
            0  p233245  Educational technology for classrooms        2  149.00
            1  p069063             Mobile classroom storage        1   99.67
```

```
In [185]: # Joining resource data per school_id to train data
# Grouping resource data per school_id to get one row per school_id
resource_data_gb = resource_data.groupby('id').agg({'price': 'sum', 'quantity': 'sum'})[['price', 'quantity']]
print(len(resource_data_gb))
```

```
Out[185]: 1464073
```

```
In [186]: resource_data_gb.head(2)
```

```
Out[186]:                price  quantity
          id
          p000000   459.56         7
          p000001   515.89        21
```

```
In [187]: print('school_state').isnull().sum()
```

```
Out[187]: 0
```

```
In [188]: print('project_grade_category').fillna('Grades PreK-2').value_counts()
```

```
Out[188]: Grades PreK-2    44232
          Grades 3-5     37137
          Grades 6-8     16923
          Grades 9-12    10963
          Name:    5942, dtyp: int64
```

```
In [189]: # or use the one of the project using resource.csv file
resource_data.head(2)
```

```
Out[189]:   id                            description quantity  price
            
```

```
In [48]: print(df[df['Rate'] ==
         0].sample(5,random_state=1))
         df[df['Rate'] ==
         0].sample(5,random_state=1)
```

```
Stdout:
      Unnamed: 0                                          review  Rate  review_length  sentiment
1342        1342  Great place to be when you are in Bangalore.   0              51          1
1557        1557                                       Crust is not very good.   0              23          0
1907        1907                      Not tasty and texture was just nasty.   0              41          0
1909        1909  The crust is not crispy and still not to my ...   0              80          0
1999        1999                         Stopped by during the weekend for dinner and the place was almost empty.   0              93          0
```

```
Stdout:
      Unnamed: 0                                          review  Rate  review_length  sentiment
1342        1342  Great place to be when you are in Bangalore.   0              51          1
1557        1557                                       Crust is not very good.   0              23          0
1907        1907                      Not tasty and texture was just nasty.   0              41          0
1909        1909  The crust is not crispy and still not to my ...   0              80          0
1999        1999                         Stopped by during the weekend for dinner and the place was almost empty.   0              93          0
```

```
In [49]: # Plot the Bivariate distribution
         ax = sns.jointplot(data=df,x='review_length',
         y='Rate',kind='hist')
```

```
In [50]: # Plot the pairwise bivariate distribution
         sns.pairplot(df,hue='sentiment')
```

```
In [51]: # Plot the count plot for author's name
         plt.figure(figsize=(10,5))
         sns.countplot(x='Rate',data=df)
```



```
In [52]: # Choose target feature for classification and pritnt
         top 10 values
         review_vc = df['review'].value_counts[:10]
         print('No. of unique values in review
         feature : {}'.format(df['review'].nunique()))
         review_vc
```

```
Stdout:
No. of unique values in review feature : 1000
```

```
Out[52]: review
         Great place to be when you are in Bangalore.                                                    1
         The crust is not crispy and still not to my ...                                                 1
         The texture of the crust is not crispy and ...                                                  1
         Not taste fresh and kept on serving old stuff.                                                  1
         Not tasty and texture was just nasty.                                                           1
         Not a good place to go to for a family gathe...                                                 1
         Not a place to be at your best.                                                                 1
         Not a place to be at your best mood.                                                            1
         Not a place to be at your best time.                                                            1
         Not a place to be when you are in a good mood.                                                  1
         Name: count, dtype: int64
```

```
In [53]: # Choose target feature for classification and pritnt
         top 10 values
         review_vc = df['review'].value_counts[:10]
         print('No. of unique values in review
         feature : {}'.format(df['review'].nunique()))
         review_vc
```

```
Stdout:
No. of unique values in review feature : 1000
```

```
Out[53]: review
         Great place to be when you are in Bangalore.                                                    1
         The crust is not crispy and still not to my ...                                                 1
         The texture of the crust is not crispy and ...                                                  1
         Not taste fresh and kept on serving old stuff.                                                  1
         Not tasty and texture was just nasty.                                                           1
         Not a good place to go to for a family gathe...                                                 1
         Not a place to be at your best.                                                                 1
         Not a place to be at your best mood.                                                            1
         Not a place to be at your best time.                                                            1
         Not a place to be when you are in a good mood.                                                  1
         Name: count, dtype: int64
```

```
In [54]: # Plot the top 10 values
         review_vc.plot(kind='bar')
```

```
In [55]: # Plot the top 10 values
         review_vc.plot(kind='barh')
```

```
In [56]: # Plot the top 10 values
         review_vc.plot(kind='pie')
```

```
In [57]: # Plot the top 10 values
         review_vc.plot(kind='area')
```

```
In [58]: # Plot the top 10 values
         review_vc.plot(kind='line')
```

4.2.2 Applying Decision Tree on TFIDF BOV

```
In [59]: # Import the required libraries
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.metrics import
         confusion_matrix,classification_report,accuracy_score,
         f1_score,recall_score,precision_score
         from sklearn.model_selection import
         train_test_split,GridSearchCV,RandomizedSearchCV
         from sklearn.feature_selection import
         SelectKBest,chi2,mutual_info_classif,mutual_info_regression
         from sklearn.preprocessing import
         MinMaxScaler,StandardScaler,Normalizer,LabelEncoder
         from sklearn.pipeline import Pipeline
         from sklearn.ensemble import
         RandomForestClassifier,AdaBoostClassifier,
         GradientBoostingClassifier,ExtraTreesClassifier,
         StackingClassifier,VotingClassifier
         from sklearn.naive_bayes import
         MultinomialNB,BernoulliNB
         from sklearn.svm import SVC,LinearSVC
         from sklearn.neighbors import
         KNeighborsClassifier,RadiusNeighborsClassifier
         from sklearn.linear_model import
         LogisticRegression,SGDClassifier,
         PassiveAggressiveClassifier,RidgeClassifier
         from sklearn.model_selection import
         cross_val_score,cross_val_predict
         from sklearn.decomposition import
         TruncatedSVD,PCA,KernelPCA
         from sklearn.manifold import
         TSNE,MDS,Isomap,LocalOptimalEmbedding
         from sklearn.cluster import
         KMeans,MiniBatchKMeans,AgglomerativeClustering,
         SpectralClustering,DBSCAN,OPTICS,Birch,MeanShift
         from sklearn.mixture import
         GaussianMixture,BayesianGaussianMixture
         from sklearn.metrics import
         silhouette_score,silhouette_samples,
         calinski_harabasz_score,davies_bouldin_score
         from sklearn.preprocessing import
         OrdinalEncoder,OneHotEncoder,LabelBinarizer,
         MultiLabelBinarizer,PolynomialFeatures,
         PowerTransformer,QuantileTransformer,
         KBinsDiscretizer,FunctionTransformer,
         KernelCenterer,StandardScaler,MinMaxScaler,
         MaxAbsScaler,RobustScaler,Normalizer
         from sklearn.impute import
         SimpleImputer,KNeighborsImputer,IterativeImputer
         from sklearn.base import
         BaseEstimator,TransformerMixin,
         RegressorMixin,ClassifierMixin
         from sklearn.tree import
         DecisionTreeClassifier,DecisionTreeRegressor,
         plot_tree,export_graphviz,export_text
         from sklearn.ensemble import
         ExtraTreeClassifier,ExtraTreeRegressor
         from sklearn.utils import
         resample,shuffle,class_weight,
         compute_class_weight,compute_sample_weight
         from scipy import
         stats,linalg,sparse,interpolate,
         integrate,optimize,special,misc
         import numpy as np,pandas as pd,
         matplotlib as mpl,seaborn as sns,
         matplotlib.pyplot as plt,
         plotly,plotly.express as px,
         plotly.graph_objs as go,plotly.figure_factory
         as ff,plotly.offline as pyoff,
         plotly.tools as pytl,cufflinks as cf,
         wordcloud,os,sys,warnings,
         pickle,joblib,gzip,gensim,
         nltk,spacy,re,string,
         collections,Counter,
         datetime,time,
         pprint,itertools,
         sqlite3,sqlalchemy,
         pymysql,psycopg2,
         pyodbc,pymongo,
         dask,ipyparallel,
         numba,numpy.linalg,
         numpy.ma,numpy.random,
         numpy.fft,numpy.polynomial,
         numpy.lib,numpy.testing,
         numpy.core,numpy.matlib,
         numpy.linalg.linalg,
         numpy.linalg.lapack_lite,
         numpy.linalg.mtrec,
         numpy.linalg._umath_linalg,
         numpy.linalg._ufuncs,
         numpy.linalg._ufuncs_cxx,
         numpy.linalg._pocketfft_internal,
         numpy.linalg._pocketfft_umath,
         numpy.linalg.lapack_lite,
         numpy.linalg.mtrec,
         numpy.linalg.setup,
         numpy.linalg.tests,
         numpy.linalg.__init__,
         numpy.linalg.__pyinstaller.__init__,
         numpy.linalg.__pyinstaller.hook-numpy.linalg,
         numpy.linalg.__pycache__.__init__.cpython-37,
         numpy.linalg.__pycache__._umath_linalg.cpython-37,
         numpy.linalg.__pycache__.setup.cpython-37,
         numpy.linalg.__pycache__.__init__.cpython-37,
         numpy.linalg.__pycache__.__init__.cpython-37,
         numpy.linalg.__pycache__.__init__
```

In [20]: /home/kaggle/input/kaggle/working/generated/sklearn.model_selection.GridSearchCV.html

```
%cat /home/kaggle/input/kaggle/working/generated/sklearn.model_selection.GridSearchCV.html
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

param = {
    'n_estimators': [100, 200],
    'criterion': ['gini', 'entropy'],
    'max_depth': [2, 3, 4, 5, 6, 7, 8],
    'max_features': ['sqrt', 'log2', None],
    'bootstrap': [True, False]
}

logreg = GridSearchCV(estimator=RF, \n\n param=param,
                      cv=StratifiedKFold(n_splits=5, random_state=42, shuffle=True),\n\n scoring='roc_auc',return_train_score=True)

print(f"Fitting {len(param['n_estimators'])} folds for each of {len(param['criterion'])} candidates, totalling {len(param['n_estimators'])*len(param['criterion'])*5} fits")

cv_results = pd.DataFrame(RS.cv_results_)
cv_results.to_csv('GridSearchCV_results.csv')

X_train_auc= logreg.predict_proba(X_train)[:,1]
X_test_auc= logreg.predict_proba(X_test)[:,1]
print('Train auc =',roc_auc_score(y_train,X_train_auc))
print('Test auc =',roc_auc_score(y_test,X_test_auc))
cm = confusion_matrix(y_test,y_pred)
print('Confusion matrix\n',cm)

plt.plot(fpr2,tpr2,'r',lw=2,label='Train ROC')
plt.plot([0,1],[0,1],'k--',lw=2,label='Random chance')
plt.xlim([-0.05,1.05])\n\n plt.ylim([-0.05,1.05])\n\n plt.rcParams['font.size'] = 12\n\n plt.title('ROC curve')
plt.xlabel('False Positive Rate (1 - Specificity)')
plt.ylabel('True Positive Rate (Sensitivity)')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()

print(RS.best_params_)
```

```
Fitting 5 folds for each of 2 candidates, totalling 10 fits
Train auc = 0.9999999999999999
Test auc = 0.9999999999999999
Confusion matrix
 [[1000    0]
 [   0 1000]]
{'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': None, 'n_estimators': 200}
```

In [21]: /home/kaggle/input/kaggle/working/GridSearchCV_results.csv

```
pd.read_csv("/home/kaggle/input/kaggle/working/GridSearchCV_results.csv")
```

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_bootstrap	param_criterion	param_max_depth	param_max_features	param_n_estimators	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score	split0_train_score	split1_train_score	split2_train_score	split3_train_score	split4_train_score	mean_train_score	std_train_score
0	0.147952	0.007630	0.013243	0.000244	True	gini	8	None	200	{'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': None, 'n_estimators': 200}	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
1	0.146455	0.007748	0.013011	0.000243	True	gini	8	sqrt	200	{'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 200}	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000

1 rows × 21 columns

Tuning the performance of the model on test data, plotting ROC Curves

In [22]: /home/kaggle/input/kaggle/working/GridSearchCV_results.csv

```
pd.read_csv("/home/kaggle/input/kaggle/working/GridSearchCV_results.csv")
```

	Unnamed: 0	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_bootstrap	param_criterion	param_max_depth	param_max_features	param_n_estimators	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score	split0_train_score	split1_train_score	split2_train_score	split3_train_score	split4_train_score	mean_train_score	std_train_score
0	0	0.147952	0.007630	0.013243	0.000244	True	gini	8	None	200	{'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': None, 'n_estimators': 200}	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
1	1	0.146455	0.007748	0.013011	0.000243	True	gini	8	sqrt	200	{'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 200}	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000

2 rows × 21 columns

Tuning the performance of the model on test data, plotting ROC Curves

In [23]: /home/kaggle/input/kaggle/working/GridSearchCV_results.csv

```
pd.read_csv("/home/kaggle/input/kaggle/working/GridSearchCV_results.csv")
```

	Unnamed: 0	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_bootstrap	param_criterion	param_max_depth	param_max_features	param_n_estimators	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score	split0_train_score	split1_train_score	split2_train_score	split3_train_score	split4_train_score	mean_train_score	std_train_score
0	0	0.147952	0.007630	0.013243	0.000244	True	gini	8	None	200	{'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': None, 'n_estimators': 200}	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
1	1	0.146455	0.007748	0.013011	0.000243	True	gini	8	sqrt	200	{'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 200}	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000

2 rows × 21 columns

Confusion Matrix

In [24]: /home/kaggle/input/kaggle/working/generated/sklearn.metrics.confusion_matrix.html

```
%cat /home/kaggle/input/kaggle/working/generated/sklearn.metrics.confusion_matrix.html
from sklearn.metrics import confusion_matrix
cnf = confusion_matrix(y_test,y_pred)\n\n print('Train confusion matrix\n',cnf)

print('Train confusion matrix')
pd.DataFrame(data=np.array([cnf[0,0],cnf[0,1],cnf[1,0],cnf[1,1]]).reshape(2,2),\n\n index=['Actual - Yes','Actual - No'],\n\n columns=['Predicted - Yes','Predicted - No'])\n\n print('Test confusion matrix')
pd.DataFrame(data=np.array([cnf[0,0],cnf[0,1],cnf[1,0],cnf[1,1]]).reshape(2,2),\n\n index=['Actual - Yes','Actual - No'],\n\n columns=['Predicted - Yes','Predicted - No'])
```

```
Train confusion matrix
 [[1000    0]
 [   0 1000]]
Train confusion matrix
Test confusion matrix
```

	Predicted - Yes	Predicted - No
Actual - Yes	1000	0
Actual - No	0	1000

```
Test confusion matrix
```

	Predicted - Yes	Predicted - No
Actual - Yes	1000	0
Actual - No	0	1000

```
The maximum value of tpr*(1-fpr) is 0.9999999999999999, for threshold 0.995
[1.000 0.000]
[1.000 0.000]
```

```
Train confusion matrix
```

	Predicted - Yes	Predicted - No
Actual - Yes	1000	0
Actual - No	0	1000

```
Train confusion matrix
```

	Predicted - Yes	Predicted - No
Actual - Yes	1000	0
Actual - No	0	1000

```
Test confusion matrix
```

	Predicted - Yes	Predicted - No
Actual - Yes	1000	0
Actual - No	0	1000

```
Test confusion matrix
```

	Predicted - Yes	Predicted - No
Actual - Yes	1000	0
Actual - No	0	1000

2. Summary

Comparing performance of all the models

In [25]: /home/kaggle/input/kaggle/working/Summary

```
%cat /home/kaggle/input/kaggle/working/Summary
pd.DataFrame(data=np.array(['Random Forest','Hyper-parameter','NO',
                            'Random Forest','Hyper-parameter','YES',
                            'Random Forest','Hyper-parameter','Tuning the performance of the model on test data, plotting ROC Curves',
                            'Random Forest','Hyper-parameter','Confusion Matrix',
                            'Random Forest','Hyper-parameter','Comparing performance of all the models
```