Movie rating score prediction

Team Members: Zhitao Gan (Owen), Chengyang Miao

Dataset: https://snap.stanford.edu/data/web-Movies.html

Goal: The goal of this project is to predict the movie rating from Amazon Movie Reviews by using the available features

- Data overview
  - Clean data
  - Add new features to the data
- Analysis on numerical columns
  - Apply KNeighborsClassifier and decision tree model
- Analysis on text column
  - Text data clean
  - Apply LinearSVC model and KNeighborsClassifier model

Data overview

- ProductID - unique identifier for the product
- UserId - unique identifier for the user
- profileName - name of the user
- Helpfulness -fraction of users who found the review helpful
- Time - timestamp for the review
- Summary - brief summary of the review
- Text - text of the review

Data conversion from .txt file to .csv file

```
product/productId: B003AI2VGA
review/userId: A141HP4LYPWMSR
review/profileName: Brian E. Erland "Rainbow Sphinx"
review/helpfulness: 7/7
review/score: 3.0
review/time: 1182729600
review/summary: "There Is So Much Darkness Now ~ Com
review/text: Synopsis: On the daily trek from Juarez, Mexic

product/productId: B003AI2VGA
review/userId: A328S9RN3U5M68
review/profileName: Grady Harp
review/helpfulness: 4/4
review/score: 3.0
review/time: 1181952000
review/summary: Worthwhile and Important Story Hampe
review/text: THE VIRGIN OF JUAREZ is based on true even

product/productId: B003AI2VGA
review/userId: A1I7QGUDP043DG
review/profileName: Chrissy K. McVay "Writer"
review/helpfulness: 8/10
review/score: 5.0
review/time: 1164844800
review/summary: This movie needed to be made.
review/text: The scenes in this film can be very disquieting
```

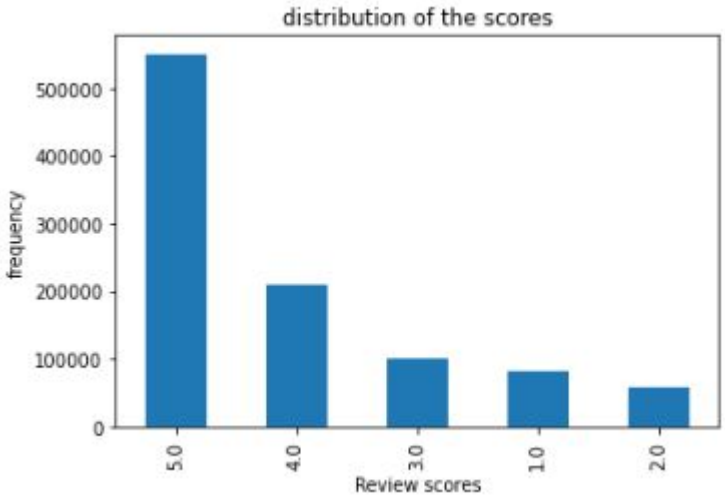
?

```python
cols = [
    'ProductId',
    'Userid',
    'Profilename',
    'Helpfulness',
    'Score',
    'Time',
    'Summary',
    'Text'
    ]

output_file = open(output_file_path, 'w', encoding='utf8')
w = csv.writer(output_file)
w.writerow(cols) # write table header first

def write_row(doc):
    w.writerow([doc.get(col) for col in cols])

count = 0
doc = {}
for line in input_file:
    line = line.strip()
    if line == '':
        write_row(doc)
        doc = {}
        count += 1

    else:
        idx = line.find(':')
        key, value = tuple([line[:idx], line[idx+1:]])
        key = key.strip().replace('/', '_').lower()
        value = value.strip()
        doc[key] = value
```

Data

	product_productid	review_userid	review_profilename	review_helpfulness	review_score	review_time	review_summary	review_text
0	B003AI2VGA	A141HP4LYPWMSR	Brian E. Erland "Rainbow Sphinx"	7/7	3.0	1182729600	"There Is So Much Darkness Now ~ Come For The ...	Synopsis: On the daily trek from Juarez, Mexic...
1	B003AI2VGA	A328S9RN3U5M68	Grady Harp	4/4	3.0	1181952000	Worthwhile and Important Story Hampered by Poo...	THE VIRGIN OF JUAREZ is based on true events s...

Data distribution



Preliminary analysis/exploration

- Useful feature: Text, Summary, Helpfulness, Scores

- People's opinions on movie are subjective mainly depending on personal preferences and experiences. However,
  - Good movies $\longrightarrow$ Higher scores
  - Strict users $\longrightarrow$ lower scores

Feature extraction

- Average_product_score (average rating scores of the movie)

```
Average_product_score=df[['ProductId','Score']].groupby(df['ProductId']).aggregate({'Score': 'mean'})
df = df.merge(Average_product_score, how='left',on='ProductId')
```

- Average_User_score (average rating scores of the user)

```
Average_User_score=df[['UserId','Score']].groupby(df['UserId']).aggregate({'Score': 'mean'})
df = df.merge(Average_User_score, how='left',on='UserId')
```

- Text + summary
  - Merge text and summary into one column

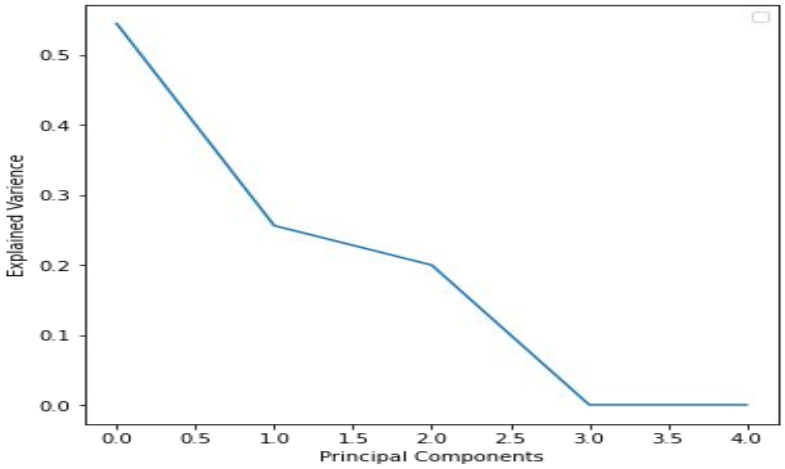
```
df['text + summary']=df['Text'] + ' ' + df['Summary']
df=df.drop(columns=['Text','Summary'])
```

Data overview

	ProductId	UserId	Helpfulness	Score	Time	Average_product_score	Average_User_score	text + summary
0	B003AI2VGA	A141HP4LYPWMSR	1.0	3.0	1182729600	2.857143	4.144766	Synopsis: On the daily trek from Juarez, Mexic...
1	B003AI2VGA	A328S9RN3U5M68	1.0	3.0	1181952000	2.857143	4.131435	THE VIRGIN OF JUAREZ is based on true events s...
2	B003AI2VGA	A1I7QGUDP043DG	0.8	5.0	1164844800	2.857143	4.700441	The scenes in this film can be very disquietin...
3	B003AI2VGA	A1M5405JH9THP9	1.0	3.0	1197158400	2.857143	3.357143	THE VIRGIN OF JUAREZ (2006) directed by K...
4	B003AI2VGA	ATXL536YX71TR	1.0	3.0	1188345600	2.857143	3.903409	Informationally, this SHOWTIME original is ess...

Analysis on numerical columns

Use StandardScaler to normalize data and Use PCA() to check if dimension reduction could be applied

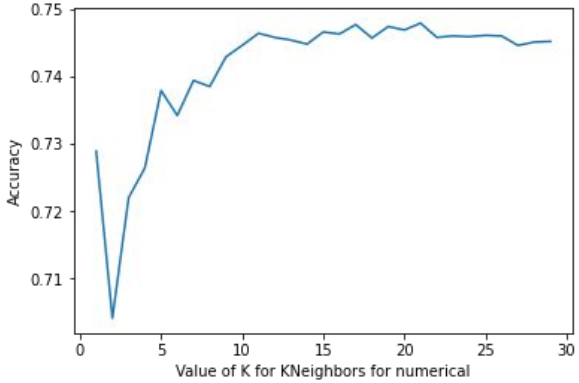


Parameters pruning

KNN Model

Find the best k for KNeighborsClassifier

- For loop



Decision tree Model

Find the best parameters for decision tree model

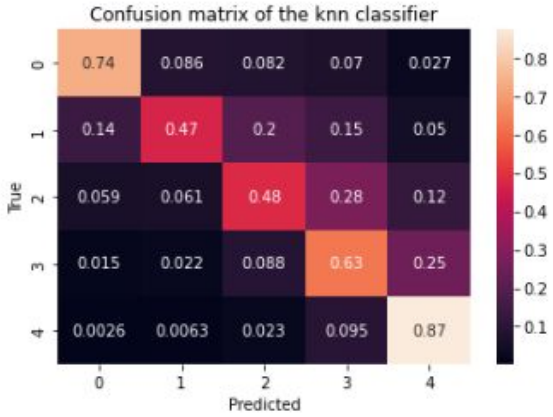
- GridSearchCV

  - Tree best parameters : {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 30, 'min_samples_split': 20, 'min_weight_fraction_leaf': 0.0}
  - Tree best estimator : DecisionTreeClassifier(criterion='entropy', max_depth=5, min_samples_leaf=30,min_samples_split=20)
  - Tree best score : 0.7353000000000001

Predictions on numerical columns

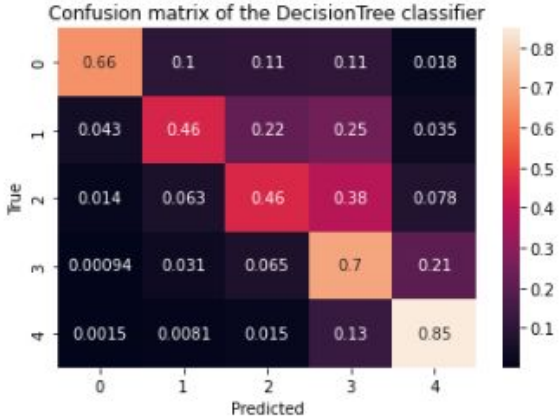
KNN Model

```
Accuracy on testing set =  0.7446
```



Decision tree Model

```
Accuracy on testing set =  0.7371
```



Data clean on text

- Clean text
  - Tokenize the text
  - remove all the punctuations
  - remove all the stopwords

```
['Synopsis', 'On', 'daily', 'trek', 'Juarez', 'Mexico', 'El', 'Paso', 'Texas', 'increasing', 'number', 'female', 'workers', 'fo
und', 'raped', 'murdered', 'surrounding', 'desert', 'Investigative', 'reporter', 'Karina', 'Danes', 'Minnie', 'Driver', 'arrive
s', 'Los', 'Angeles', 'pursue', 'story', 'angers', 'local', 'police', 'factory', 'owners', 'employee', 'undocumented', 'alien
s', 'pointed', 'questions', 'relentless', 'quest', 'truth.', 'br', 'br', 'Her', 'story', 'goes', 'nationwide', 'young', 'girl',
'named', 'Mariela', 'Ana', 'Claudia', 'Talancon', 'survives', 'vicious', 'attack', 'walks', 'desert', 'crediting', 'Blessed',
'Virgin', 'rescue', 'Her', 'story', 'enhanced', '``', 'Wounds', 'Christ', "''", 'stigmata', 'appear', 'palms', 'She', 'claims',
'received', 'message', 'hope', 'Virgin', 'Mary', 'soon', 'fanatical', 'movement', 'forms', 'fight', 'evil', 'holds', 'strangleh
old', 'area.', 'br', 'br', 'Critique', 'Possessing', 'lifelong', 'fascination', 'esoteric', 'matters', 'Catholic', 'mysticism',
'miracles', 'mysterious', 'appearance', 'stigmata', 'I', 'immediately', 'attracted', "'05", 'DVD', 'release', 'Virgin', 'Juare
z', 'The', 'film', 'offers', 'unique', 'storyline', 'blending', 'current', 'socio-political', 'concerns', 'constant', 'flow',
'Mexican', 'migrant', 'workers', 'forth', 'U.S./Mexican', 'border', 'traditional', 'Catholic', 'beliefs', 'Hispanic', 'populati
on', 'I', 'I', 'surprised', 'unexpected', 'route', 'taken', 'plot', 'means', 'methods', 'heavenly', 'message', 'unfolds.', 'b
r', 'br', 'Virgin', 'Juarez', 'film', 'care', 'watch', 'interesting', 'merit', 'viewing', 'Minnie', 'Driver', 'delivers', 'soli
d', 'performance', 'Ana', 'Claudia', 'Talancon', 'perfect', 'fragile', 'innocent', 'visionary', 'Mariela', 'Also', 'starring',
'Esai', 'Morales', 'Angus', 'Macfadyen', 'Braveheart', '``', 'There', 'Is', 'So', 'Much', 'Darkness', 'Now', 'Come', 'For', 'Th
e', 'Miracle', "''"]
```

```python
from nltk.stem.snowball import SnowballStemmer
from nltk.tokenize import word_tokenize, sent_tokenize
from spacy.lang.en.stop_words import STOP_WORDS
import string


punct = string.punctuation      #list of puntuations
stopwords = list(STOP_WORDS)    # list of stopwords
snowball = SnowballStemmer(language='english')
def Cleaned_data(df):
    data=df
    message=sent_tokenize(data)
    word_tokens=[]
    for word in message:
        word_tokens+=word_tokenize(word)
    cleaned_tokens=[]
    for token in word_tokens:
        #remove all the punctuations
        if token not in punct and token not in stopwords:
            cleaned_tokens.append(token)
    return cleaned_tokens
```

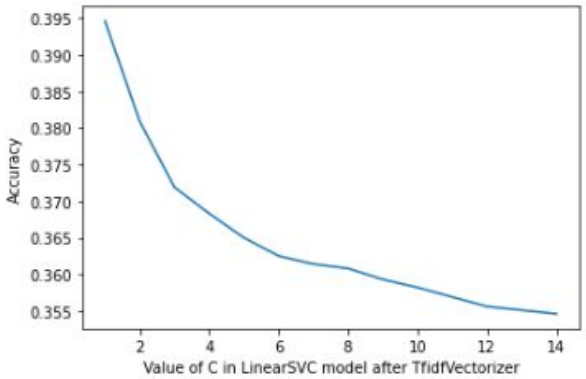
- Apply Tfidf methods

```python
Tfid=TfidfVectorizer(tokenizer=Cleaned_data,max_features=20000, ngram_range=(1,5))
```

Parameter pruning

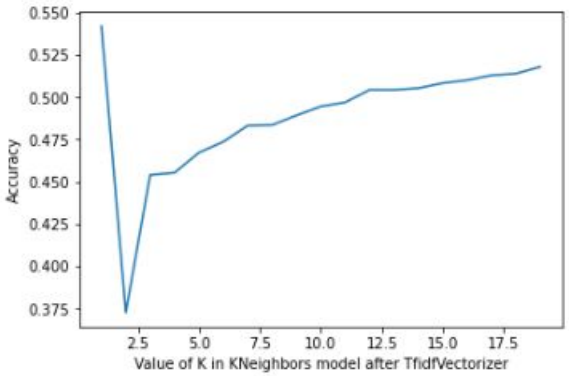
LinearSVC model

- For loop



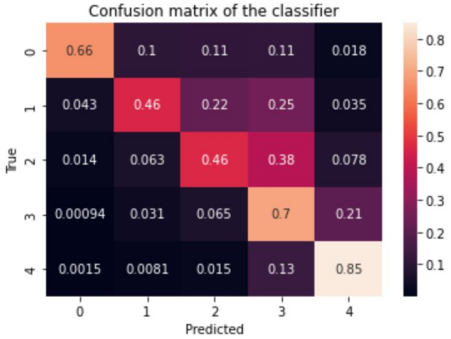
KNN model

- For loop



MultinomialNB model

```
Accuracy on testing set =  0.7371
```



How to combine numerical data and text data?

“Voting system”

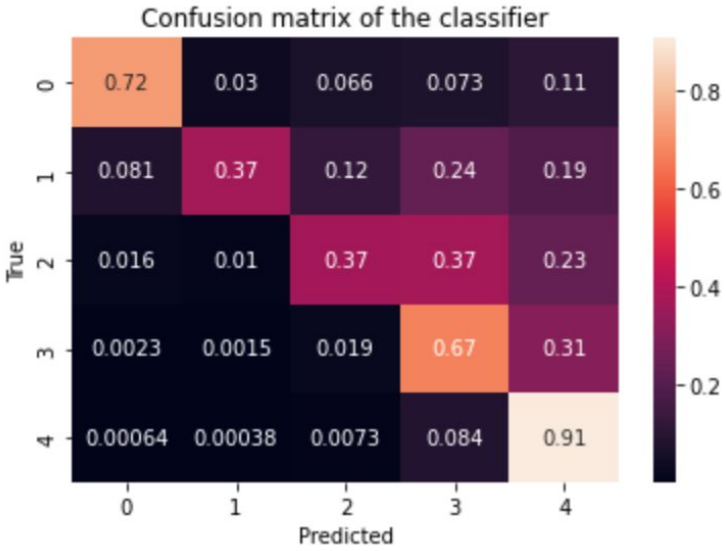
- Make three sets of predictions based on three models
- The score predicted by the most models (more than half) is the final prediction
- If predictions are all different, use the prediction of MultinomialNB

```
model1 = Pipeline(steps=[('Tfidf', TfidfVectorizer(tokenizer=Cleaned_data,max_features=20000, ngram_range=(1,5))),
                         ('classifier', MultinomialNB())]).fit(X_train_processed_cat, Y_train1)
model2 = KNeighborsClassifier(n_neighbors=10).fit(X_train_processed_num, Y_train1)
model3 =DecisionTreeClassifier(max_depth=5, min_samples_leaf=20,
                               min_samples_split=20).fit(X_train_processed_num, Y_train1)
```

```
def most_frequent(List):
    occurence_count = Counter(List)
    return occurence_count.most_common(1)[0][0]
Ypred_final=[]
for x in range(len(Ypred_1)):
    result=[]
    result.append(Ypred_1[x])
    result.append(Ypred_2[x])
    result.append(Ypred_3[x])
    Ypred_final.append(most_frequent(result))
```

More accurate :) !!!

```
Accuracy on testing set =  0.75673
```



KNN on num: 0.7446

Decision tree on num: 0.7371

MultinomialNB on text: 0.7371

Representation of numerical data in text

- Express numerical data in a sentence
- Add it to the beginning of the text

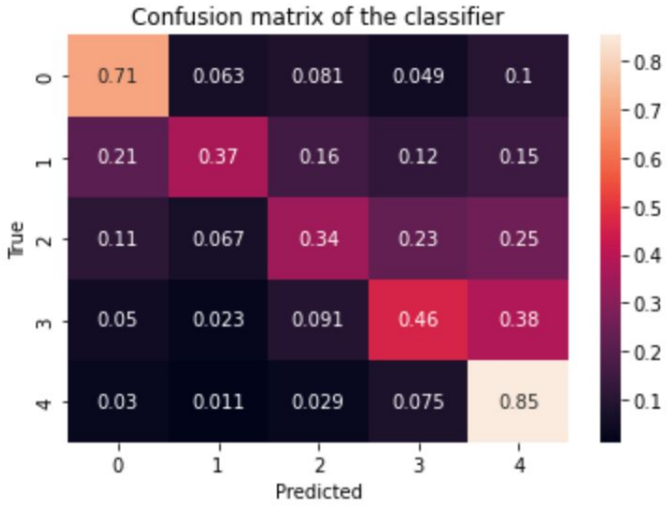
```
combined += "The helpfulness for this product is {:}, and the average score is {:} \
             and the user has average score of {:}."
            .format(row["Helpfulness"],row["Average_product_score"],row["Average_User_score"])
```

Data after merging:

	Score	reviews
0	3.0	The helpfulness for this product is 1.0, and t...
1	3.0	The helpfulness for this product is 1.0, and t...
2	5.0	The helpfulness for this product is 0.8, and t...
3	3.0	The helpfulness for this product is 1.0, and t...
4	3.0	The helpfulness for this product is 1.0, and t...

Decrease the accuracy :(

```
Accuracy on testing set =  0.67902
```



KNN on num: 0.7446

Decisiontree on num: 0.7371

MultinomialNB on text: 0.7371

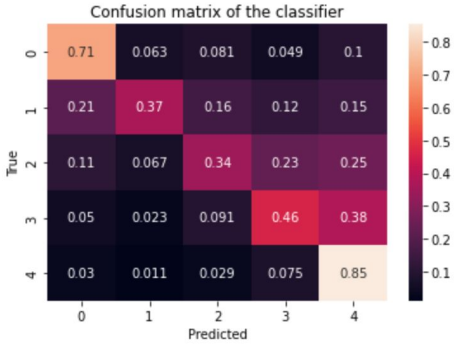
Voting system: 0.7567

Limitations and Challenges

- We only use a proportion of data(approximately 1 million), because the running time is too long
- Limited number of models
- Better method to clean text ?
- Better predictions on Score 2 and 3?

```
['Synopsis', 'On', 'daily', 'trek', 'Juarez', 'Mexico', 'El', 'Paso', 'Texas', 'increasing', 'number', 'female', 'wor
kers', 'found', 'raped', 'murdered', 'surrounding', 'desert', 'Investigative', 'reporter', 'Karina', 'Danes', 'Minni
e', 'Driver', 'arrives', 'Los', 'Angeles', 'pursue', 'story', 'angers', 'local', 'police', 'factory', 'owners', 'empl
oyee', 'undocumented', 'aliens', 'pointed', 'questions', 'relentless', 'quest', 'truth.', 'br', 'br', 'Her', 'story',
'goes', 'nationwide', 'young', 'girl', 'named', 'Mariela', 'Ana', 'Claudia', 'Talancon', 'survives', 'vicious', 'atta
ck', 'walks', 'desert', 'crediting', 'Blessed', 'Virgin', 'rescue', 'Her', 'story', 'enhanced', '```', 'Wounds', 'Chri
st', "'''", 'stigmata', 'appear', 'palms', 'She', 'claims', 'received', 'message', 'hope', 'Virgin', 'Mary', 'soon',
'fanatical', 'movement', 'forms', 'fight', 'evil', 'holds', 'stranglehold', 'area.', 'br', 'br', 'Critique', 'Possess
ing', 'lifelong', 'fascination', 'esoteric', 'matters', 'Catholic', 'mysticism', 'miracles', 'mysterious', 'appearanc
e', 'stigmata', 'I', 'immediately', 'attracted', "'05", 'DVD', 'release', 'Virgin', 'Juarez', 'The', 'film', 'offer
s', 'unique', 'storyline', 'blending', 'current', 'socio-political', 'concerns', 'constant', 'flow', 'Mexican', 'migr
ant', 'workers', 'forth', 'U.S./Mexican', 'border', 'traditional', 'Catholic', 'beliefs', 'Hispanic', 'population',
'I', 'I', 'surprised', 'unexpected', 'route', 'taken', 'plot', 'means', 'methods', 'heavenly', 'message', 'unfolds.',
'br', 'br', 'Virgin', 'Juarez', 'film', 'care', 'watch', 'interesting', 'merit', 'viewing', 'Minnie', 'Driver', 'deli
vers', 'solid', 'performance', 'Ana', 'Claudia', 'Talancon', 'perfect', 'fragile', 'innocent', 'visionary', 'Mariel
a', 'Also', 'starring', 'Esai', 'Morales', 'Angus', 'Macfadyen', 'Braveheart', '```', 'There', 'Is', 'So', 'Much', 'Da
rkness', 'Now', 'Come', 'For', 'The', 'Miracle', '''']
```

```
Accuracy on testing set =  0.67902
```



Thank you for listening :)