RATINGS PREDICTION PROJECT

Submitted by:

AJITESH KUMAR

ACKNOWLEDGMENT

I would like to thank FlipRobo Technologies for giving me the opportunity to work on this project. I am very grateful to DataTrained team for providing me the knowledge which helped me a lot to work on this project. Reference sources are:

1. Google

2. YouTube

3. TowardsDataScience

4. Stackoverflow

5. DataTrained Notes

INTRODUCTION

- ## Business Problem Framing

  This problem is related to one such clients of FlipRobo Technologies who has a website where people write different reviews for technical products. Now they are adding a new feature to their website i.e. the reviewer will have to add stars(rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don't have a rating. So, we have to build an application which can predict the rating by seeing the review.

- ## Conceptual Background of the Domain Problem

  Internet has revolutionizing the way of shopping. Now we can do shopping seating in our homes by few clicks. The e-commerce industry is growing rapidly by extending it's reach to almost every corner of the world. So, there are plenty of online marketing websites are available and lot's of product are available. This leads to confusion in our mind that from where we can get the best product.

- ## Review of Literature

  Reviews & Ratings plays very significant role in deciding the correct product. Now a days, buyer can get the real idea about the product by reading the product reviews and ratings posted by the other buyers on the e-commerce website.

Analytical Problem Framing

- Data Sources and their formats

Loading the dataset:

```
df = pd.read_csv('final_data.csv')

df.head(10)
```

	Unnamed: 0	Ratings	Reviews
0	0	5	This laptop is very light weight thus you can ...
1	1	5	Great for the price range, I was sceptical abo...
2	2	5	Good Product, I am satisfied all of the featur...
3	3	5	Bought for 36k with SBI credit card discount.F...
4	4	5	I just bought this laptop after exchanging my ...
5	5	5	I was confused between asus and lenovo s145 la...
6	6	5	Good product in this price ...
7	7	5	I compare various leptop in i7 -11th gen with ...
8	8	5	After using one week I fell fell the laptop is...
9	9	4	Good laptop with best processor. It was delive...

- Data Preprocessing Done

```
#Replacing email address with 'email'

df['Reviews']=df['Reviews'].str.replace(r'^.+@[^\.].*\[a-z]{2,}$','emailaddress')

#Replacing URLs with 'webaddress'
df['Reviews']=df['Reviews'].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\s*)?$','webaddress')

#Replacing money symbol with 'moneysymb'(£ can type with ALT key+156)
df['Reviews']=df['Reviews'].str.replace(r'£|\$','dollers')

#Replacing 10 digit phone number(format include paranthesis, space, no spaces,dashes) with 'phone number'
df['Reviews']=df['Reviews'].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$','phonenumber')

#Replacing whitespace between terms with a single space
df['Reviews']=df['Reviews'].str.replace(r'\s+',' ')

#Replacing number with 'numbr'
df['Reviews']=df['Reviews'].str.replace(r'^\d+(\.\d+)?','numbr')

#Removing punctuation
df['Reviews']=df['Reviews'].str.replace(r'[^\w\d\s]',' ')

#Removing leading and trailing whitespace
df['Reviews']=df['Reviews'].str.replace(r'^\s+|\s+?$',' ')

df.head()
```

	Ratings	Reviews	length
0	5	this laptop is very light weight thus you can ...	507
1	5	great for the price range i was sceptical abo...	508

- Hardware and Software Requirements and Tools Used

  We have used the following Software & Libraries

  1. Jupyter Notebook
  2. Python
  3. Pandas
  4. Numpy
  5. Matplotlib
  6. Seaborn
  7. NLTK
  8. SkLearn

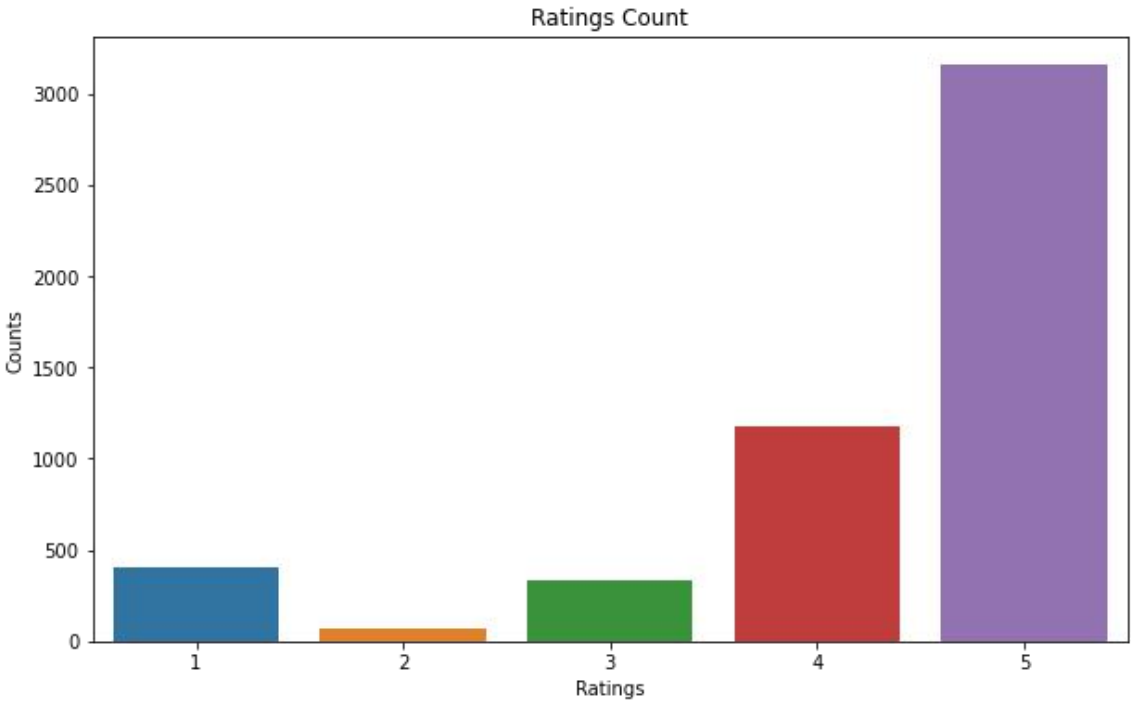
Model/s Development and Evaluation

- Key Metrics for success in solving problem under consideration

	Model	Accuracy_score	Cross_val_score	Difference
0	LogisticRegression	75.058275	61.324521	13.733754
1	KNeighborsClassifier	70.396270	57.108966	13.287304
2	DecisionTreeClassifier	74.592075	53.515488	21.076587
3	RandomForestClassifier	77.622378	64.160416	13.461962

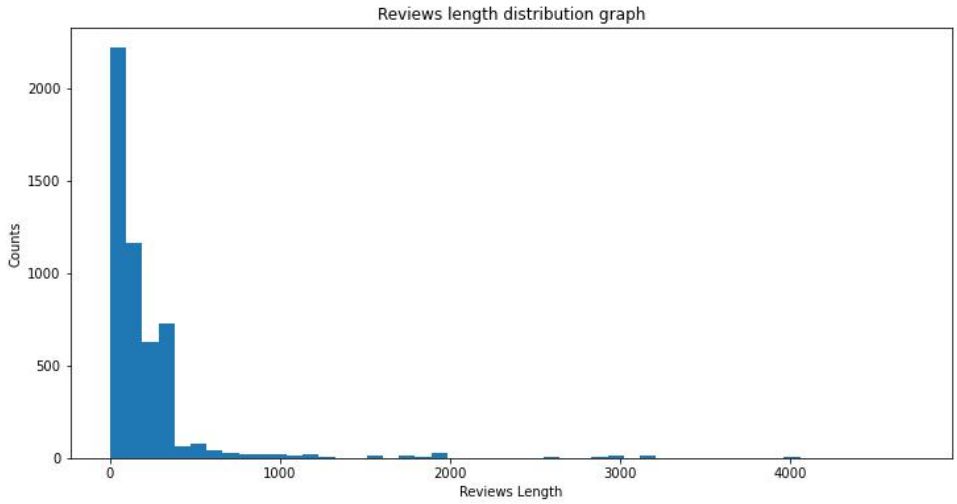
- From the above table, we found that the minimum difference between the accuracy score and cross validation score is for KNeighborsClassifier. **So, the best fit model for our project is KNeighborsClassifier.**

- Visualizations




```
# Create and print a Reviews length distribution graph
review_length_distribution = pd.DataFrame(df["Reviews"].str.len())
review_length_distribution = review_length_distribution[review_length_distribution.Reviews < 5000]
review_length_distribution.groupby(["Reviews"])
review_length_distribution = review_length_distribution.plot(kind='hist', legend=None, bins=50, figsize=[12, 6])
review_length_distribution.set_xlabel("Reviews Length")
review_length_distribution.set_ylabel("Counts")
review_length_distribution.set_title("Reviews length distribution graph")
```

```
Text(0.5, 1.0, 'Reviews length distribution graph')
```



- Interpretation of the Results

1. Logistic Regression model:

```
LR=LogisticRegression()
Model.append('LogisticRegression')
LR.fit(x_train,y_train)
print(LR)
pre=LR.predict(x_test)
print('\n')
AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)
score.append(AS*100)
sc=cross_val_score(LR,x,y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc,'\n')
cv_score.append(sc*100)
```

```
LogisticRegression()


Accuracy_score=  0.7505827505827506
Cross_val_score= 0.6132452093181239
```

---

2. KNeighborsClassifier:

```
KNN=KNeighborsClassifier(n_neighbors=6)

Model.append('KNeighborsClassifier')
KNN.fit(x_train,y_train)
print(KNN)

pre=KNN.predict(x_test)
AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)

score.append(AS*100)
sc=cross_val_score(KNN,x,y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc)
cv_score.append(sc*100)
```

```
KNeighborsClassifier(n_neighbors=6)
Accuracy_score=  0.703962703962704
Cross_val_score= 0.5710896619396719
```

3. DecisionTreeClassifier:

```
DT=DecisionTreeClassifier()
Model.append('DecisionTreeClassifier')
DT.fit(x_train,y_train)
print(DT)
pre=DT.predict(x_test)
print('\n')
AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)
score.append(AS*100)
sc=cross_val_score(DT,x,y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc,'\n')
cv_score.append(sc*100)
```

```
DecisionTreeClassifier()


Accuracy_score=  0.745920745920746
Cross_val_score= 0.5351548774849746
```

4. RandomForestClassifier:

```
RF = RandomForestClassifier()
Model.append('RandomForestClassifier')
RF.fit(x_train,y_train)
print(RF)
pre=RF.predict(x_test)
print('\n')

AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)
score.append(AS*100)
sc=cross_val_score(RF,x,y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc,'\n')

cv_score.append(sc*100)
```

```
RandomForestClassifier()


Accuracy_score=  0.7762237762237763
Cross_val_score= 0.6416041590006321
```

CONCLUSION

- Key Findings and Conclusions of the Study

  After comparing all the models we found that the best fit model for our problem is KNeighbours Classifier.