Machine Learning Nanodegree

Capstone Report

Product Supplier Search Process

Anirban Paul

July 28, 2018

1 Domain Background

The average customer of today owns and uses at least four devices – mobile, tablet, laptop and a desktop on a daily basis through various ecommerce websites like Flipkart, Amazon, Aliexpress etc. At least 50-60% of customers use their smartphones and tablets to make online purchases. Numerous amounts of data are generated every day in various ecommerce websites. Lots of important and valuable information is stored in these data which can be used to discover various patterns and which could give some great insights. The benefits of using machine learning in ecommerce can be found in this link: https://www.loop54.com/how-machine-learning-can-benefit-your-e-commerce-company

With the help of data various customer buying patterns can be discovered which could help retailers in selling their products at a good profit. In other ways these information can also be used to help customers in choosing the best product and recommend them to select the best sender or the top ten vendors for the particular product as in this case is ASUS Laptops. I am going to develop recommender systems which will help the customer to know the top ten vendors for a product. are very powerful tools that are user friendly and provide germane suggestions to users that are quite accurate, and can help create a positive impact.

I would be using three machine learning approaches for building the model for the prediction of the vendors. I will be using AdaBoost, Gradient Boosting and Decision Tree Regressor(if needed) to predict the top ten vendors at for ASUS laptops. Amongst these three models the ADA Boost will serve as a benchmark model and an attempt will be made to build at least one model which outperforms or is at par with the benchmark model, which I am looking forward to achieve with the other two model (Decision Tree Regressor and Gradient Boosting). The dataset that I have used for this project is manually collected from various websites like Flipkart, Amazon, Aliexpress, Snapdeal etc. containing 6 features such as Seller Name, Price, Ratings, Delivery Days, Websites, Shipping Charge and 1 label which is known as Score which is basically a scoring function calculated depending different features. The dataset has 50 different sellers.

Gradient boosting is a machine learning method created by Leo Breiman for regression and classification related problems, which produce mostly decision tree in the form of

an ensemble of weak models. . It builds stage-wise models like other boosting approach and it allows optimization of differentiable loss function.

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gödel Prize for their work. It can be used in combination with many other various types of algorithms to improve performance and get better result. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier.

Some of the previous work of seller selection proces can be found in the below link:

https://www.deltabid.com/vendor-selection-process/

Problem Statement

In this project I will be using three machine learning technique to develop a recommender system that gives top ten vendors for ASUS laptop. There will various sellers and their features depending on those features a scoring function will be calculated which will serve as a target label, more the scores better will be the vendor. The three machine learning techniques are ADA Boost, Decision Tree Classifier and Gradient Boosting. These methods will be used to predict the scores of different vendors on the test set. I will be using AdaBoost, Decision Tree Regressor(if needed) and Gradient Boosting to predict the top ten vendors for ASUS laptops across various market place. Amongst these three models the ADA Boost will serve as a benchmark model and an attempt will be made to build at least one model which outperforms or is at par with the benchmark model, which I am looking forward to achieve with the other two model (Decision Tree Regressor and Gradient Boosting)

The problem is a regression problem, in which the main task is to predict the scores of the vendors and that score will be used to determine the top ten vendors from the test set. There will be various input features like Seller Name, Price, Ratings, Delivery Days, Websites, Shipping Charge and Score.

Metrics

This project will be evaluated with regards to the model's ability to predict the top ten vendors. I expect that the two model that I am going to use other than ADA Boost model which will be my benchmark model among them at least one model amongst them to perform better than the benchmark model. The project should be able to train successfully on random training data and then test the model and produce an accurate score based on features. The Evaluation metrics for this case is Mean Squared Error and also predicted top 10 sellers in test set by the various model will be compared with the original and based on no of wrong vendors present in top 10 will be used to say accuracy.

2 Analysis

Data Exploration

The dataset that I have used for this project is manually collected from various websites like Flipkart, Amazon, Aliexpress, Snapdeal etc. containing 6 features such as Seller Name, Price, Ratings, Delivery Days, Websites, Shipping Charge and 1 label which is known as Score which is basically a scoring function calculated depending different features. The dataset has 50 different sellers. Description of different features is given below:

Features	Description
Seller Name	Seller name on a particular marketplace.
Websites	Online websites formally known as market place names present in the form of a string.
Rating	Ratings of the sellers or vendors by their customers.
Price	Price in rupees on which the seller is selling product.
Delivery Days	No of days taken by vendor to deliver the product.
Shipping Charges	Shipping amount charged by the vendor
Score	The quality of vendor calculated based on above features.

The mathematical equation for scoring function is Score=(ratings*100000)/(delivery days*price*shipping charges). I have multiplied ratings with 100000 because the score was coming very small.

The first five rows of the initial data are given below:

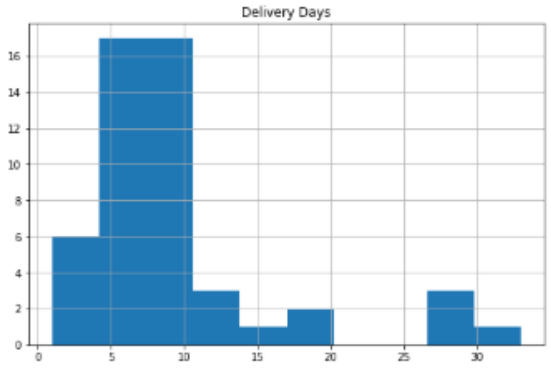
Out[24]:

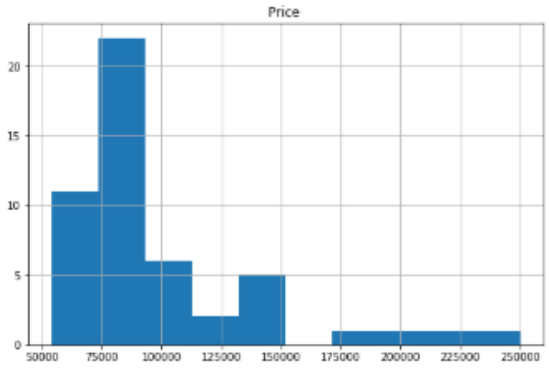
	Seller Name	Price	Website	Ratings	Delivery Days	Shipping Charge	Score
0	AAAEnterprise	69990	www.flipkart.com	3.5	4	1	12.501786
1	Samrat	72790	www.flipkart.com	3.7	6	1	8.471860
2	RetailNet	107990	www.flipkart.com	4.5	5	1	8.334105
3	The Lapstore	85450	www.flipkart.com	3.5	5	1	8.191925
4	Sunstar_Surat	70700	www.amazon.in	4.0	7	1	8.082441

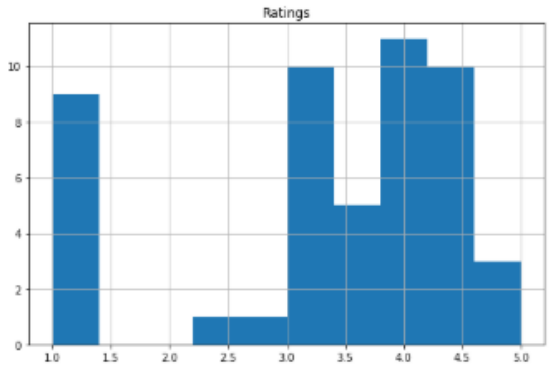
Exploratory Visualization

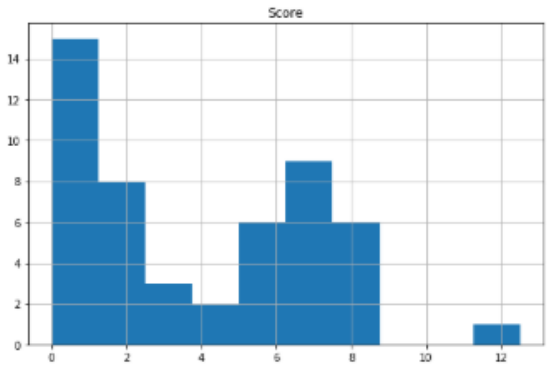
The dataset has 6 features those are Seller Name, Price, Website, Ratings, Delivery Days, Shipping Charge and Score as target label. From the above exploration it is clearly visible that the final column Score is a measure of goodness given to a seller. So the aim of the project will be to use machine learning algorithm to make the algorithm learn from the training set and to

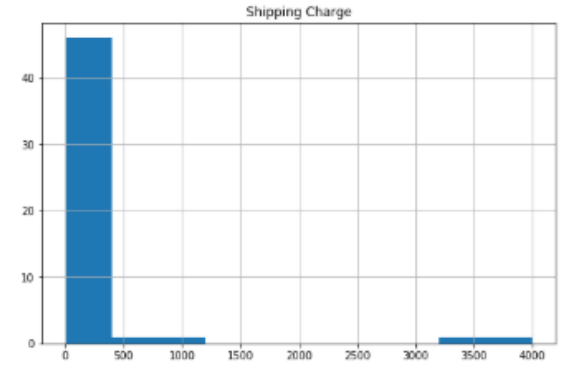
predict the score of the testing set. Histogram is used for the data visualization of features distribution among the sellers. The histogram visualization is given below:





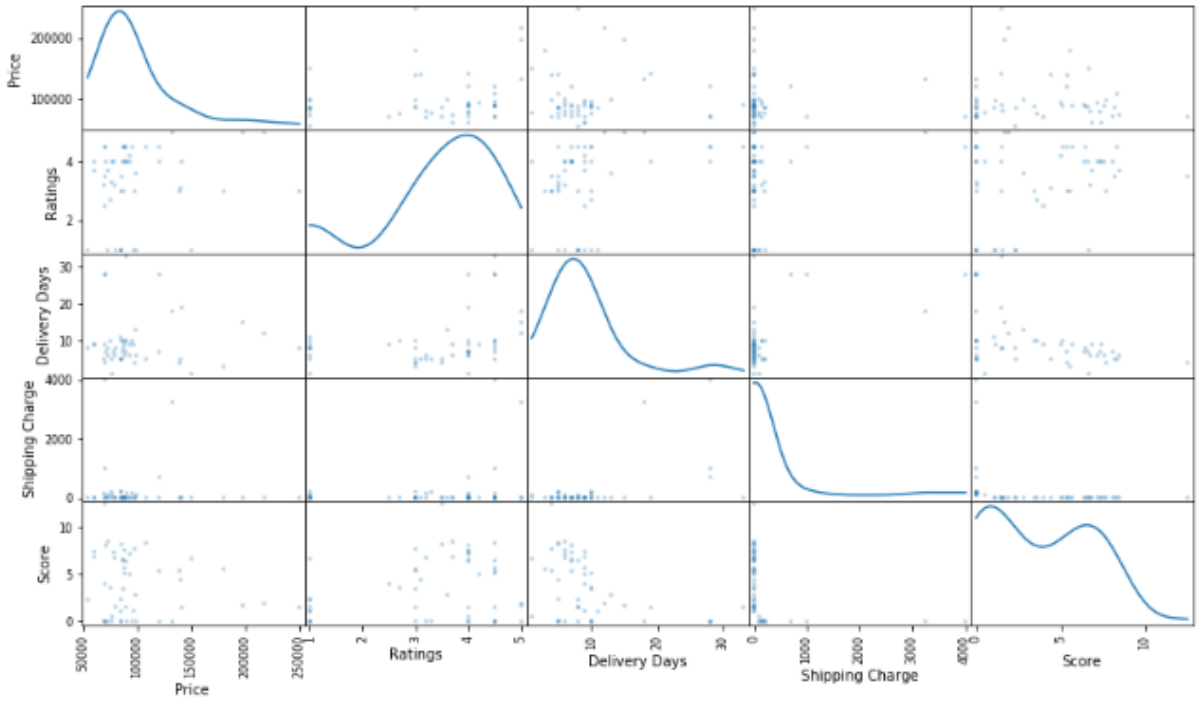






.

To understand the relationship between two variables scatter matrix used is given below:-



From the above visualization it is clear that the variables are linearly independent and all are to be considered while predicting the target label.

Algorithms and Techniques

The three supervised learning algorithms that I will be using to classify the dataset are:

i)Ada Boosting generally works in combination with various types of algorithms or learners. In this case Decision Tree Regressor is boosted with ‘max_depth=4’ which defines the height of the tree and ‘n_estimators=300’ which indicates number trees to be used.

```
In [51]: from sklearn.ensemble import AdaBoostRegressor
         regr_2 = AdaBoostRegressor(DecisionTreeRegressor(max_depth=4),
                                    n_estimators=300, random_state=0)
         pred2=regr_2.fit(features, label)
         pred2= regr_2.predict(test[ftcol].values)
         test['Adaboost Score'] = pred2

         test.head(n=15)
```

ii)In decision trees the target label(Score in this case) can take continuous values are called regression trees. In decision trees the dataset is divided into smaller subsets which are used

to form a tree like structure. The output result is in the leaf nodes of the tree. In decision tree regression, a decision tree is used to make decision also help in decision making. The parameters used in this case are default with ‘random state=0’ because the predicting varies if the random state is natural number.

```
In [40]: from sklearn.tree import DecisionTreeRegressor
         Model2 = DecisionTreeRegressor(random_state=0)
         pred=Model2.fit(features, label)
         pred = pred.predict(test[ftcol].values)
```

iii) Gradient boosting produces mostly decision tree in the form of an ensemble of weak models. . It builds stage-wise models like other boosting approach and it allows optimization of differentiable loss function. In this case the parameters used are ‘n_estimators:1000’ which indicates the number of trees I have increased the numbers of trees in this case to get more better efficiency, max_depth is kept same, with learning rate=0.01. With all these parameters the algorithms produced good predictions with less mean squared error.

```
from sklearn import ensemble
params = {'n_estimators': 1000, 'max_depth': 4, 'min_samples_split': 2,
          'learning_rate': 0.01, 'loss': 'ls', 'random_state': None}
clf = ensemble.GradientBoostingRegressor(**params)

pred1=clf.fit(features, label)
pred1=pred1.predict(test[ftcol].values)
test['Gradient Boost Score'] = pred1

test.head(n=15)
```

These three algorithms will be used to predict the top 10 seller and will be used to compare them with actual top 10 seller in the test set.

Benchmark Model

The Benchmark model for this type of problem is decided as ADA Boost Regressor with the prediction accuracy of 80% of top 10 sellers in test set and other model like Decision Tree Regressor(if needed) and Gradient Boosting will be tried on the test set to obtain better or at par performance with Benchmark Model. The final accuracy of different with the benchmark model will be produced in excel file(CSV) to show the whether top ten vendors produced by the algorithms is present in the original top ten vendors in the test set list or not.

3 Methodology

Data Preprocessing

As it has been mentioned in Data exploratory section that the dataset that I have used for the project is manually collected from various websites like Flipkart, Amazon, Aliexpress, Snapdeal

etc. containing 6 features such as Seller Name, Price, Ratings, Delivery Days, Websites, Shipping Charge and 1 label which is known as Score which is basically a scoring function calculated depending different features. The dataset has 50 different sellers. So since the data is collected manually the dataset is clean. Before running the algorithms the two columns containing the website name and the seller name is dropped as it is in string format and it is not required by the algorithm for predicting the score.

Implementation

This project comprises of three different machine learning algorithms they are Ada Boost, Decision Tree and Gradient Boosting to predict scores of the seller which will be used to predict the top 10 sellers and will also be used to calculate the mean squared error. So before applying the algorithm the dataset is split randomly using train test split into 70% training data and 30% testing data.

Train Test Split

```
In [29]: from sklearn.model_selection import train_test_split
         train,test = train_test_split(new_data,test_size=0.3,random_state=42)
         len(train)
```

After the split Adaboost is implemented on the dataset and the result is given below

```
In [51]: from sklearn.ensemble import AdaBoostRegressor
         regr_2 = AdaBoostRegressor(DecisionTreeRegressor(max_depth=4),
                                    n_estimators=300, random_state=0)
         pred2=regr_2.fit(features, label)
         pred2= regr_2.predict(test[ftcol].values)
         test['Adaboost Score'] = pred2

         test.head(n=15)
```

```
In [52]: from sklearn.metrics import mean_squared_error
         mean_squared_error(test['Score'].values, pred2)

Out[52]: 5.676856895950545
```

The top ten predicted by Ada Boost is

Actual top 10 by the in test set	Top 10 predicted by Ada boost
The Lapstore	**Cart2India SLP** **X**
Sunstar_Surat	The Lapstore
Sigmait_slc	Sunstar_Surat
Weppas	Sigmait_slc
eClinic	OnlineKings
Pagr4114	eClinic
OnlineKings	Pagr4114
Mobile And Laptop World	**Sunstar_hazira** **X**
Nt Traders	Nt Traders
Web Fanatics	Weppas

So it can be seen from the table that the top 10 predicted by ADA boost has 80% accuracy with two wrongly predicted sellers marked in red and the Mean Squared error is = **5.676856895950545** so this is the benchmark model.

Result for decision tree algorithm implementation

```
In [40]: from sklearn.tree import DecisionTreeRegressor
         Model2 = DecisionTreeRegressor(random_state=0)
         pred=Model2.fit(features, label)
         pred = pred.predict(test[ftcol].values)
```

```
In [44]: from sklearn.metrics import mean_squared_error
         mean_squared_error(test['Score'].values, pred)

Out[44]: 4.156632960967857
```

The top ten predicted by Decision Tree Regressor is

Actual top 10 by the in test set	Top 10 predicted by Decision Tree Regressor
The Lapstore	Sunstar_Surat
Sunstar_Surat	The Lapstore
Sigmait_slc	Sigmait_slc
Weppas	OnlineKings
eClinic	eClinic
Pagr4114	**Sunstar_hazira** **X**
OnlineKings	Mobile And Laptop World
Mobile And Laptop World	Nt Traders
Nt Traders	Weppas
Web Fanatics	Web Fanatics

So it can be seen from the table that the top 10 predicted by Decision Tree Regressor has 90% accuracy w ith one wrong predicted sellers marked in red and the Mean Squared error is = **4.156632960967857.** So thi s shows improvement in both accuracy and Mean Squared Error.

Result for Gradient Boosting algorithm implementation

```
from sklearn import ensemble
params = {'n_estimators': 1000, 'max_depth': 4, 'min_samples_split': 2,
          'learning_rate': 0.01, 'loss': 'ls', 'random_state': None}
clf = ensemble.GradientBoostingRegressor(**params)

pred1=clf.fit(features, label)
pred1=pred1.predict(test[ftcol].values)
test['Gradient Boost Score'] = pred1

test.head(n=15)
```

```
from sklearn.metrics import mean_squared_error
mean_squared_error(test['Score'].values, pred1)
```

```
1.827484043606553
```

The top ten predicted by Gradient Boosting Algorithm is

Actual top 10 by the in test set	**Top 10 predicted by Gradient Boosting**
The Lapstore	Sunstar_Surat
Sunstar_Surat	The Lapstore
Sigmait_slc	Sigmait_slc
Weppas	eClinic
eClinic	OnlineKings
Pagr4114	Pagr4114
OnlineKings	**Sunstar_hazira** **X**
Mobile And Laptop World	Nt Traders
Nt Traders	Mobile And Laptop World
Web Fanatics	Weppas

So it can be seen from the table that the top 10 predicted by Gradient boosting has 90% accuracy (more th an the benchmark model) with one wrong predicted sellers marked in red and the Mean Squared error is = **1.827484043606553,** this shows that though the accuracy didn't change but the Mean Squared Error have reduced at considerable amount. So this is the final model.

Refinement

After running the project first with the benchmark Ada Boosting Regression model with 'n_estimators'=300, I got accuracy 80% and 5.676856895950545 of mean squared error. Then I applied normal decision tree regressor and got a accuracy of 90% and 4.156632960967857 of mean squared error which is higher than the benchmark model. Then I implemented with Gradient Boosting with 'n_estimators'=1000 and got 90% of accuracy and 1.827484043606553 of mean square error which is better than the benchmark model and decision tree. So the final model for this system is Gradient Boosting Algorithm.

4 Results and discussion

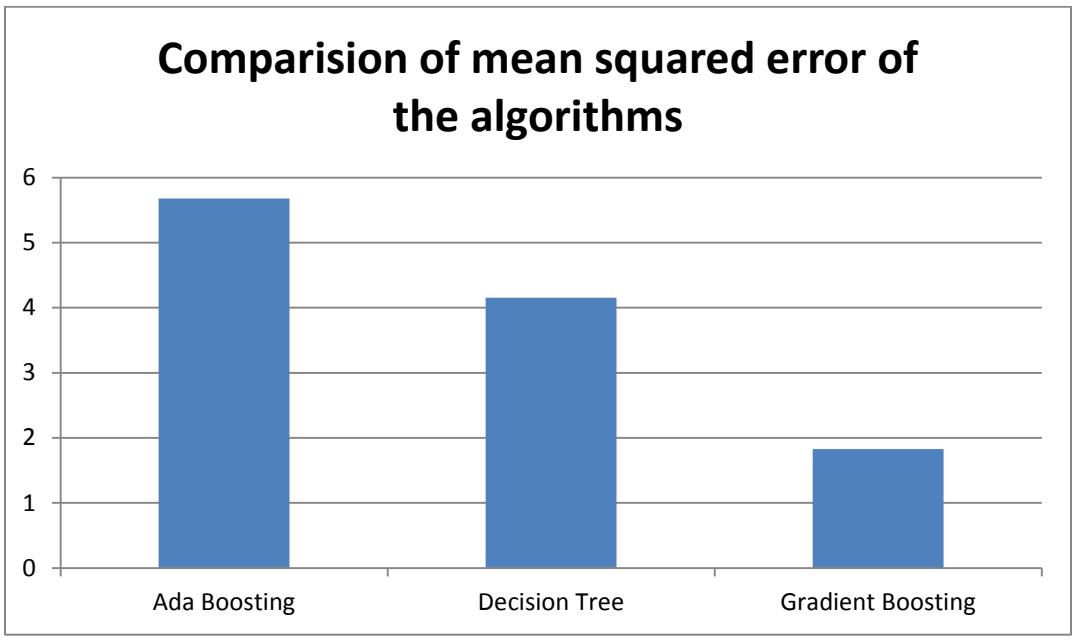
Model Evaluation and Validation

I have evaluated each regression model based on their mean squared error and accuracy to predict top ten sellers. From the beginning of the project my aim of the project was to make a recommender systems which will help customer to know the top ten sellers of ASUS laptop at a particular place will have less mean squared error and higher accuracy than the benchmark model(Ada Boost), so that being the important part of the project. On the basis of their mean squared error I have provided a bar chart. Mean squared error results for all three algorithms are as follows:

a) Ada Boosting: 5.676856895950545

b) Decision Tree: 4.156632960967857

c) Gradient Boosting: 1.827484043606553

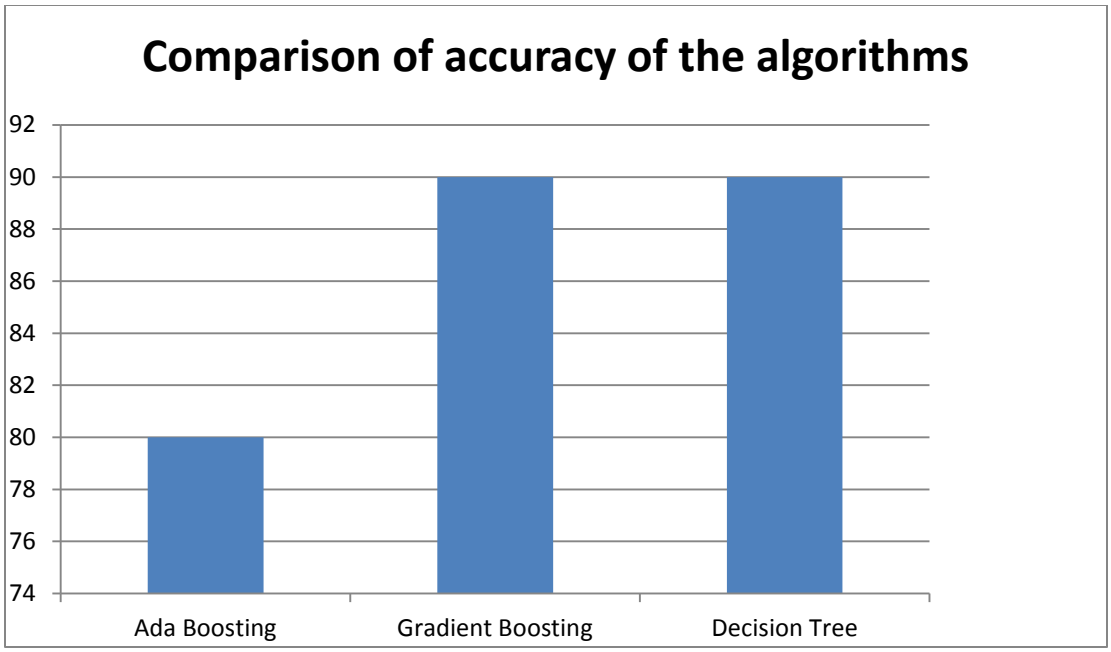


Accuracy results for all three algorithms are as follows:

a) Gradient Boosting: 90%

b) Neural Network: 90%

c) Ada Boosting: 80%



Justification

I would like to justify my project implementation results by referring to the implementation section of the project. In the beginning the benchmark model Ada Boosting was applied on the dataset and the accuracy was 80% and the mean squared error was 5.676856895950545 which was very high. So the target has been to get better accuracy than the benchmark model and less mean squared. So to achieve the aim different algorithms like Decision Tree and Gradient Boosting was tried. So from the implementation section it is clear that the Gradient Boosting was the final model with accuracy 90% while predicting the top 10 sellers and mean squared error = 1.827484043606553 which was great improvement the from the benchmark model(mean squared error=5.676856895950545), hence it justifies the project.

5 Conclusion

Free Form Visualization

Out[51]:

	Price	Ratings	Delivery Days	Shipping Charge	Score	Decision Tree Score	Gradient Boost Score	Adaboost Score
13	86480	4.0	7	1	6.607638	6.385390	6.837647	6.896399
39	75990	3.3	5	200	0.043427	0.035420	0.088545	0.061836
30	197683	5.0	15	1	1.686201	1.915797	1.912903	1.915797
45	84999	1.0	10	215	0.005472	0.017052	0.015019	0.035420
17	179999	3.0	3	1	5.555586	1.500060	4.690405	5.395683
48	132401	5.0	18	3244	0.000647	0.031546	0.039830	0.031546
26	98388	3.6	13	1	2.814602	5.020674	3.664752	5.020674
25	86990	3.0	10	1	3.448672	5.114218	3.181973	3.889028
32	89399	4.5	33	1	1.525337	5.681883	4.368951	5.067446
19	119990	4.5	7	1	5.357589	6.385390	6.128359	6.993682

Since the dataset is randomly divided into 30% of test sets and 70% training data we can see from the visualization that after the split the value predicted by Gradient Boost and Decision Tree quite close with the actual value of score, still the Gradient Boosting is a better algorithm because it was 90% accurate in predicting the sellers(second last column in the table) and the

score predicted by Gradient Boosting had least mean squared error of 1.827484043606553 while others had mean squared error more than 4, which indicates quite good performance.

Reflection

For this project, I first referred to various websites, market place like Flipkart, amazon, Aliexpress, Snapdeal to get information about the dealers of ASUS laptop. I then identified my aim. This was followed by research on which algorithm or regression algorithm would be best suited for my project among which them I selected one as a benchmark model. The dataset that was to be used was identified and studied in depth. The dataset was subjected to preprocessing which mostly involved dropping the string formatted column. The regression models I thought to be appropriate and have used are Gradient Boosting, Decision Tree and Ada Boositing(Benchmark model). I have discussed why I have chosen these three regression algorithms and how they can help in fulfilling the objective of the project. I have then evaluated each regression model based on their mean squared error and their ability to predict the top sellers of ASUS laptops. The entire framework of the project was also discussed.

From the mean squared error results, Gradient boosting algorithm was found to be the most accurate algorithm. Decision Tree Algorithm was found to have higher mean squared error than Gradient Boosting but less than the benchmark model Ada Boositing algorithm. One can infer from these observations that ensemble techniques are very powerful and cannot be ignored in the prediction of important data. A recommender system has been developed and the objective of the project has been achieved.

Improvement

I have used a simple Ada Boosting and compared it with algorithms like Decision Tree and Gradient boosting. The decision tree and gradient boosting techniques proved to be more efficient than Ada Boosting. From these observations it can be seen that ensemble techniques are very powerful and cannot be ignored in the prediction of important data.

Obviously there is little scope to make improvement to achieve more accuracy, like using various powerful tools like pruning and principal component analysis or pipelining and grid search to improve the accuracy value and the mean squared value can be improved. Even complex neural networks can also be used to make a recommender with more accuracy with the real world.