Group #02 Avocado Maniac

Avocado data analysis

Jixuan Li 40073785 Dejian Wang 27754396

ABSTRACT

In the project, we tried to classify the avocado type and to make prediction of the avocado price by using classifying models and regression models. Then we decided the best fitting model for the avocado data. We made the conclusion at the end of the file.

1. INTRODUCTION

When you are trying to pick the best avocado from a huge heap in the supermarket, have you ever thought about how we can do an analysis about the avocado data? By this project, we are going to do two main goals: First classifying the avocado into organic or conventional group by its feature. Then we can use the avocado data to train a model to do the price prediction. Thanks to Kaggle we got a bunch of avocado selling stats data of their date, average price, the vending volume of each brand, also the type (organic or normal), and which state this data came from.

2. DATA PREPARATION

The raw data we got from Kaggle is in csv format (Appendix: Table 2-1), there are 14 features in total:

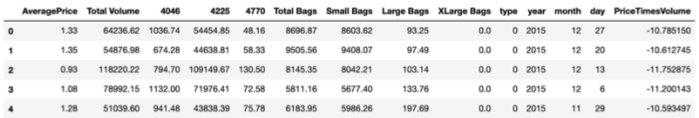
- Titleless index: index starting from 0
- Date: the date of the observation
- AveragePrice: the average price of a single avocado
- Total Volume: total volume of avocados sold
- 4046: total number of avocados with PLU 4046 sold
- 4225: total number of avocados with PLU 4225 sold
- 4770: total number of avocados with PLU 4770 sold
- Total Bags: total # of bags sold in the date
- Small Bags / Large Bags / XLarge Bags: number of bags sold by the size of bag
- Type: whether the avocado data is conventional or organic
- Year: the data recorded year
- Region: the US state of the record

2.1 Feature Preprocessing

In order to elevate the performance of our model, we need to preprocess the data. So, we split the date into year, month, and day. Also, we removed the region column since our analysis is not related to the state difference.

At the same time, as per instructor's requirement, we added a new feature: price/volume. Because on comparison with the volume, price is way too small, we introduced log into the feature to prove underflow.

Table 2-1-2 Engineered Features



	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year	month	day	PriceTimesVolume
0	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	0	2015	12	27	-10.785150
1	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	0	2015	12	20	-10.612745
2	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	0	2015	12	13	-11.752875
3	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	0	2015	12	6	-11.200143
4	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	0	2015	11	29	-10.593497

2.2 Package Importing

In this project, we decided to use the following packages:

·Sklearn
·Pyplot
·Pandas
·Numpy
·Seaborn

3. CLASSIFICATION MODELLING

The main topic of our classification modeling is "given the features, try to determine whether the avocado is Organic or Not.

According to what we learn in this class, we will first use classic models (the best one from Logistic Regression, K neighbors, SVM, Gaussian Naive Bayes and Decision Tree models) to do the classification, then we will try to fit the data with Ensemble Models (the best performance of Ada boost, Gradient Boost, Random Forest, Extra Tree model)

3.1 Data Preparation

Here we need to prepare the data for our classification model training.

- Split parameters and targets from the raw data
- Standardization, scaling target value into interim [-1,1]
- Split the data into train/test sets. (80% training 20% testing)

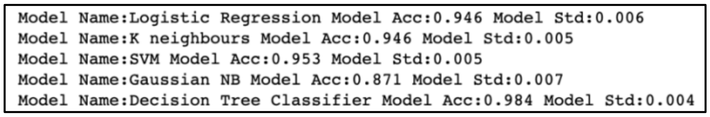
3.2 Classic Classification Modeling

First, let us apply the classic classification models to the data.

We are planning to apply 10-fold cross validation to following classic models showing below, then we pick the one with best performance for further tuning.

- Logistic Regression
- K Neighbors
- Support Vector Machine Classifier
- Gaussian Naive Bayes Classifier
- Decision Tree Classifier

In all the classic models, Decision Tree behaves the best (As shown in Figure 3.2).



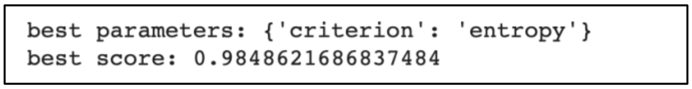
```
Model Name:Logistic Regression Model Acc:0.946 Model Std:0.006
Model Name:K neighbours Model Acc:0.946 Model Std:0.005
Model Name:SVM Model Acc:0.953 Model Std:0.005
Model Name:Gaussian NB Model Acc:0.871 Model Std:0.007
Model Name:Decision Tree Classifier Model Acc:0.984 Model Std:0.004
```

Figure 3.2 Result of 10-Fold Cross Validation - Classic Classification Models

3.2.1 Decision Tree Parameter Refining

As we can see from the 10-fold cross validation, the Decision Tree Model scores the most, so we continue refining the DT classifier:

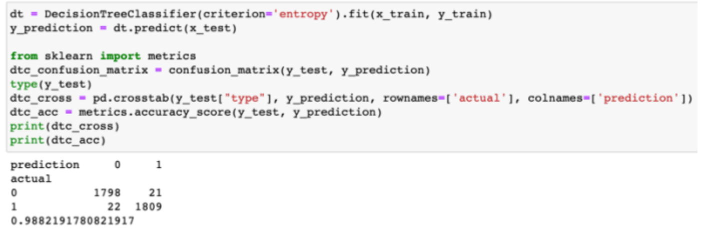
We use grid search, let sklearn help us to search the best hyperparameter.



```
best parameters: {'criterion': 'entropy'}
best score: 0.9848621686837484
```

Figure 3.2.1a Decision Tree Modeling Grid Search Result

So, we built a DT model with entropy.



```
dt = DecisionTreeClassifier(criterion='entropy').fit(x_train, y_train)
y_prediction = dt.predict(x_test)

from sklearn import metrics
dtc_confusion_matrix = confusion_matrix(y_test, y_prediction)
type(y_test)
dtc_cross = pd.crosstab(y_test["type"], y_prediction, rownames=['actual'], colnames=['prediction'])
dtc_acc = metrics.accuracy_score(y_test, y_prediction)
print(dtc_cross)
print(dtc_acc)
```

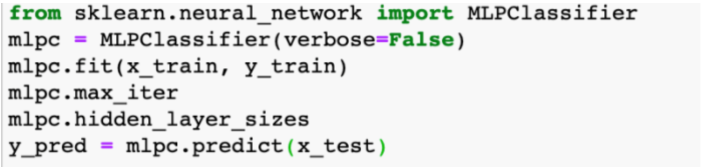
```
prediction     0     1
actual
0           1798    21
1             22  1809
0.9882191780821917
```

Figure 3.2.1b Decision Tree Model Training and Confusion Matrix

3. 3 Ensemble Models

Now we can apply the showing ensemble models to the raw data.

- Ada Boost
- Gradient Boosting
- Random Forest
- Extra Trees

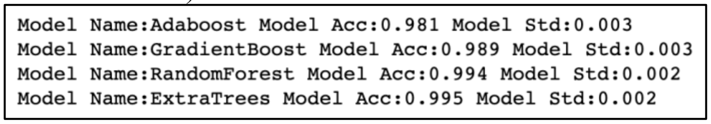


```
from sklearn.neural_network import MLPClassifier
mlpc = MLPClassifier(verbose=False)
mlpc.fit(x_train, y_train)
mlpc.max_iter
mlpc.hidden_layer_sizes
y_pred = mlpc.predict(x_test)
```

Figure 3.4a Neural Network Model Training

Similar as before, we use 10-fold validation to verify the models.

As we can see from the cross-validation score, the Extra Tree Classifier behaves the best (with the score of 0.995).

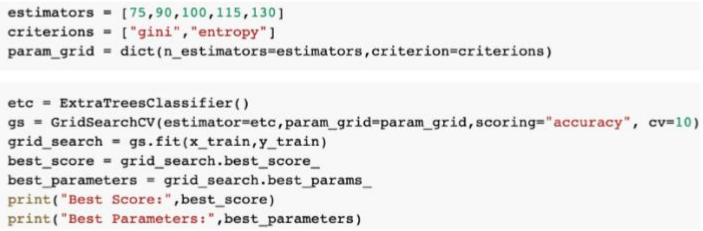


```
Model Name:Adaboost Model Acc:0.981 Model Std:0.003
Model Name:GradientBoost Model Acc:0.989 Model Std:0.003
Model Name:RandomForest Model Acc:0.994 Model Std:0.002
Model Name:ExtraTrees Model Acc:0.995 Model Std:0.002
```

Figure 3.3 Result of 10-Fold Cross Validation - Ensemble Classification Models

3.3.1 Extra Decision Tree Classifier Refining

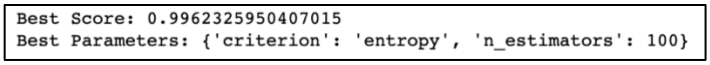
We can use grid search to get the hyperparameter set of the extra tree classifier.



```
estimators = [75,90,100,115,130]
criterions = ["gini","entropy"]
param_grid = dict(n_estimators=estimators,criterion=criterions)
```

```
etc = ExtraTreesClassifier()
gs = GridSearchCV(estimator=etc,param_grid=param_grid,scoring="accuracy", cv=10)
grid_search = gs.fit(x_train,y_train)
best_score = grid_search.best_score_
best_parameters = grid_search.best_params_
print("Best Score:",best_score)
print("Best Parameters:",best_parameters)
```

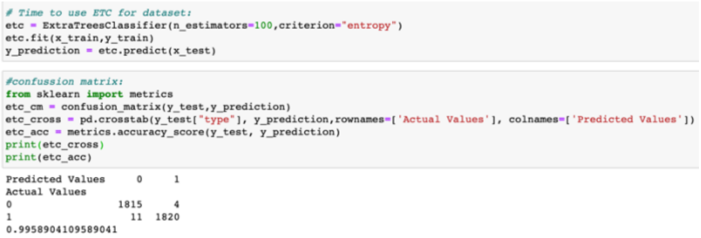
Figure 3.3.1a Extra Tree Classifier Grid Search Code



```
Best Score: 0.9962325950407015
Best Parameters: {'criterion': 'entropy', 'n_estimators': 100}
```

Figure 3.3.1b Extra Tree Classifier Grid Search Result

We can build an **extra tree classifier** with **entropy** criterion, and **n_estomators = 100**.



```
# Time to use ETC for dataset:
etc = ExtraTreesClassifier(n_estimators=100,criterion="entropy")
etc.fit(x_train,y_train)
y_prediction = etc.predict(x_test)
```

```
#confussion matrix:
from sklearn import metrics
etc_cm = confusion_matrix(y_test,y_prediction)
etc_cross = pd.crosstab(y_test["type"], y_prediction,rownames=['Actual Values'], colnames=['Predicted Values'])
etc_acc = metrics.accuracy_score(y_test, y_prediction)
print(etc_cross)
print(etc_acc)
```

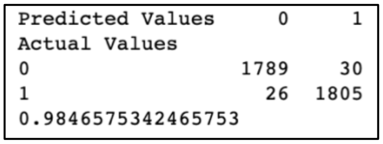
```
Predicted Values     0     1
Actual Values
0                 1815     4
1                   11  1820
0.9958904109589041
```

Figure 3.3.1c Extra Tree Classifier Building

As we can see from the result of ensemble models, the accuracy score is 0.995, in comparison with the classic model accuracy score: 0.988, it works better.

3.4 Neural Network

Similarly, we can train a Neural Network Classification Model to do the classification project.



```
Predicted Values     0     1
Actual Values
0                 1789    30
1                   26  1805
0.9846575342465753
```

Figure 3.4b Neural Network Model Performance

3.5 Model Comparison

As the results shown above, we can get the accuracy scores easily:

Decision Tree Classifier: 0.988

Extra Tree Classifier: 0.995

Neural Network Classifier: 0.984

As the accuracy score shown, extra tree classifier performs the best.

4.REGRESSION MODELING

Since classification modeling did a great job on classifying the tag between conventional and organic avocados, we are now curious about the performance of machine learning on avocado price prediction.

4.1 Ordinary Least Squares (OLS) Estimation

This part is to test whether the quality of the dataset affects our future modelling, the high R-squared indicates that our model explains a lot of the response variability. And the small P values indicate that we can reject the null hypothesis that quantity has no effect on Price (Figure 4.1 in appendix).

4.2 Regression Predict (Classical Models)

Before getting into the regression modelling, the data set has been split into 80% training set and 20% testing set for future testing. There are 6 classical regression modeling which includes linearly Regression, lasso, elastic net, K-neighbors Regressor, decision tree regressor and SVR. The target is to compare this regression model in order to find which model best fits the avocado original dataset. And choose the "best fitting model" for further comparison.

As we can see from Figure 4.2, the score of both SVR (support vector regressor) and linear regressor reach the highest and beat the rest models, but we can't simply conclude the champions will be the best regressors to predict the price of avocado. Instead, we try to use ensemble models to give it another try. In order to compare my classical decision tree regressor (DTR) to ensemble models in the next step, I need to tune my hyper parameter that is used in DTR. But the performance does not match my expectation, which only rise from 0.647 to 0.660.

4.3 Ensemble Models

The following ensemble models will be the challengers of classical models above, which includes ada-boosting regressor, gradient-boosting regressor, random forest regressor and extra-tree regressor. Since there are both boosting and bagging ways of modelling chosen in this part, the compared result will be more persuasive and effective (Figure 4.3).

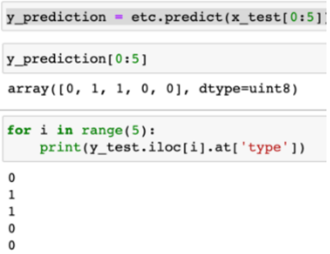
4.3.1 Compare Ensemble Models to Classical Models

Before comparing classical and ensemble models, other two popular models (Neural Network and Xgboost) were added to increase impartiality of the tender result (Figure 4.4).

5.SUMMARY

5.1 Classification and Regression Models Testing Result

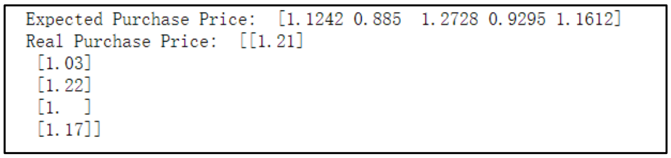
Since we can see that the extremely random tree model did great predictions at both classification and Regression section, we planned to use 5 test data sets to verify the accuracy of these models (ETR and ETC) again.



```
y_prediction = etc.predict(x_test[0:5]

y_prediction[0:5]

array([0, 1, 1, 0, 0], dtype=uint8)

for i in range(5):
    print(y_test.iloc[i].at['type'])

0
1
1
0
0
```

Figure 5.1.a Testing Extremely Random Tree Classifier.

The 5-test data sets are 100% match the real price, which means that ETC did perfectly in classifying avocado between conventional and organic in this case.



```
Expected Purchase Price:  [1.1242 0.885  1.2728 0.9295 1.1612]
Real Purchase Price:  [[1.21]
 [1.03]
 [1.22]
 [1.  ]
 [1.17]]
```

Figure 5.1.b Testing Extremely Random Tree Regressor

The result of using ETR to predict the price of avocado by using 5 test data sets is very close to the real price, therefore our regression model has good accuracy.

5.2 Something Interesting

During the testing of current models, we find that if we add the new feature “AvgPrice/TotalVolume” to the training data set, most regressor performance will increase remarkably. The linear regressor even reach around 0.96 accuracy (Figure 5.2) and each esemble model R2 score increase around 0.12 (Figure 5.3). most important of all, in final comparison, the R2 score range of all four most powerful regression models increased from 0.65-0.85 to 0.8-0.96(Figure 5.4).

5.3 Conclusion

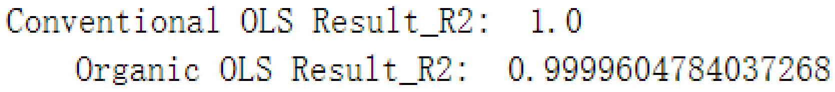
As we can see from the previous result, for both classification and regression model, the Extra Decision Tree Model behaves the best. So, for the avocado data, we can say EDT is the best solution.

During the preparation of this project, we realized that for different data sets, we need to use different ML models to fit it. There’s no best algorithm in the machine learning area, only the most suitable algorithm for the specific data. So, when we try to solve a brand-new ML project, we need to try different ML models first, and then pick the one with the best behavior for the prediction making. Although the process may take a long time, it is of vital importance to get a better machine learning prediction result.

APPENDIX

	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLargeBags	type	year	region
0	2015/12/27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0	conventional	2015	Albany
1	2015/12/20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0	conventional	2015	Albany
2	2015/12/13	0.93	118220.22	794.7	109149.67	130.5	8145.35	8042.21	103.14	0	conventional	2015	Albany
3	2015/12/6	1.08	78992.15	1132	71976.41	72.58	5811.16	5677.4	133.76	0	conventional	2015	Albany
4	2015/11/29	1.28	51039.6	941.48	43838.39	75.78	6183.95	5986.26	197.69	0	conventional	2015	Albany
5	2015/11/22	1.26	55979.78	1184.27	48067.99	43.61	6683.91	6556.47	127.44	0	conventional	2015	Albany

Table 2-1 Sample Raw Data



```
Conventional OLS Result_R2:  1.0
    Organic OLS Result_R2:  0.9999604784037268
```

Figure 4.1 OLS Estimation Output

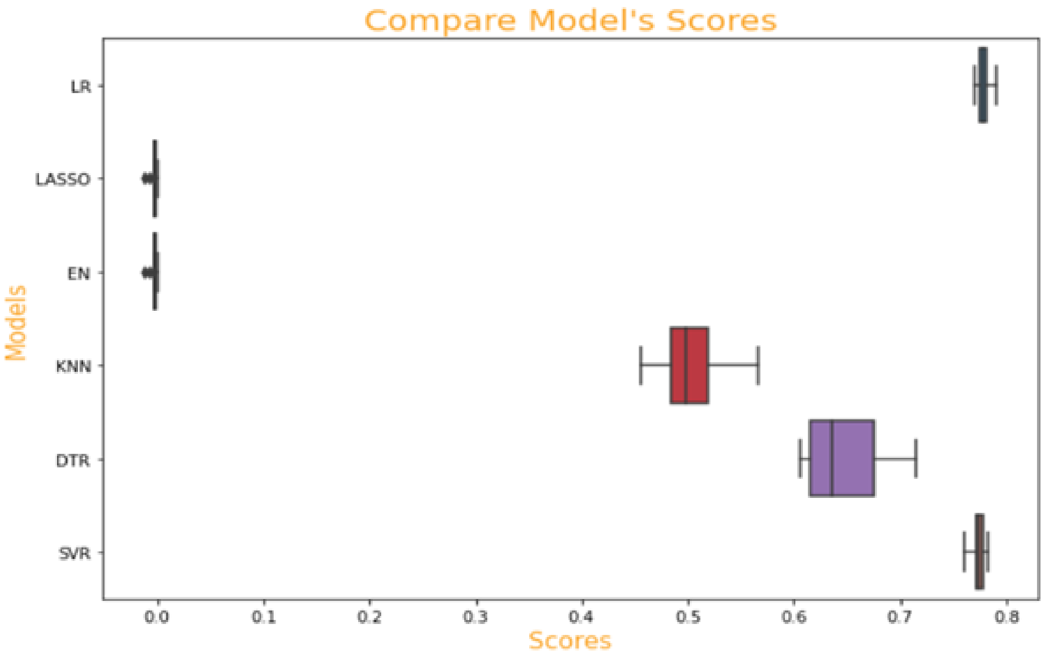


Figure 4.2 Classical Regression Models Comparison

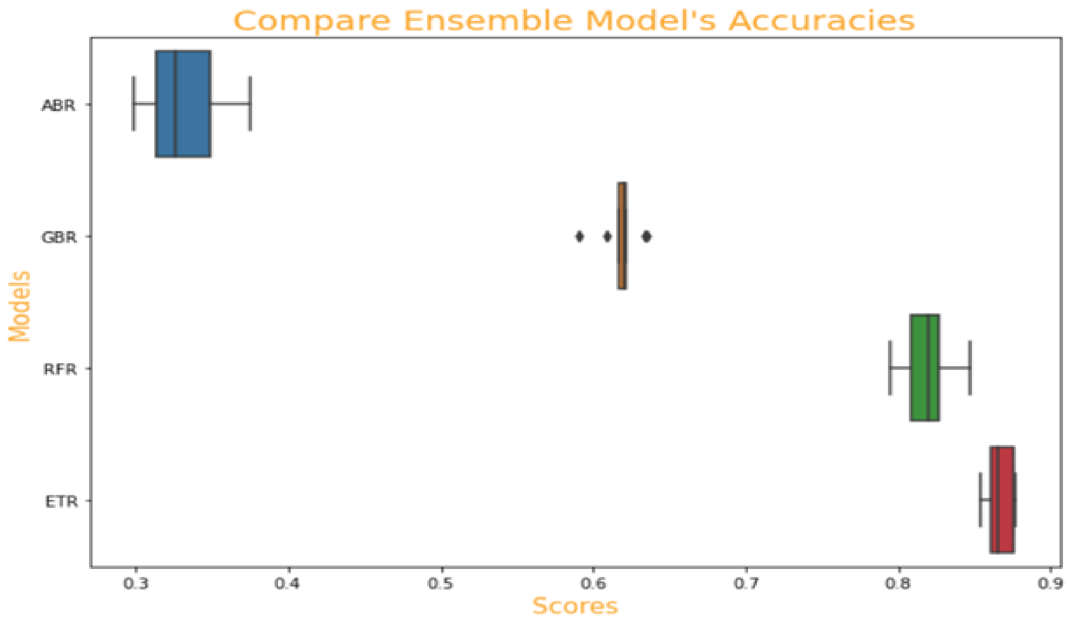


Figure 4.3 Ensemble Regression Models Comparison

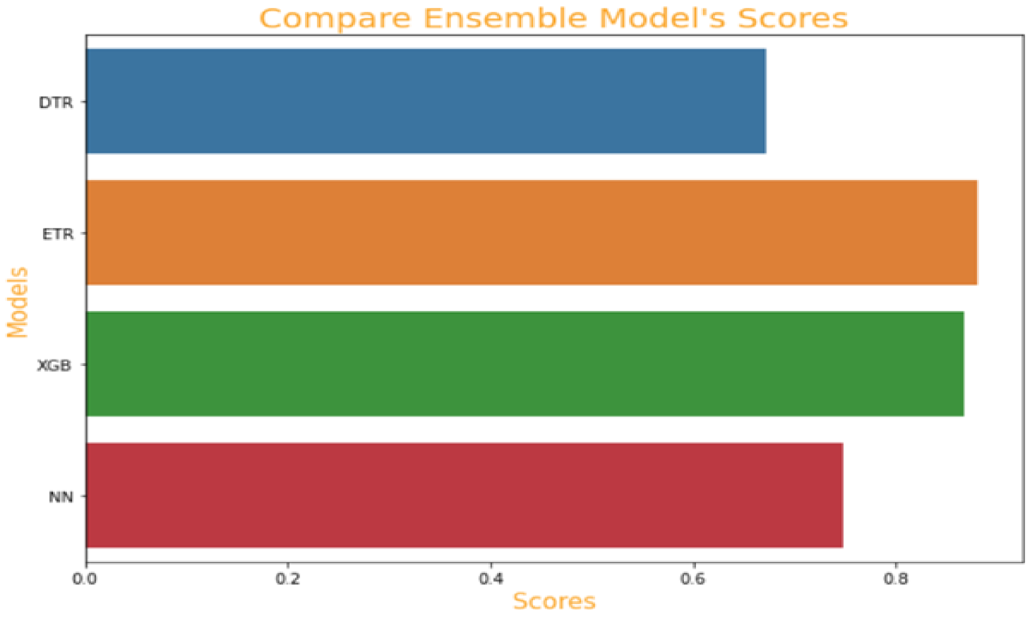


Figure 4.4 Compare Ensemble Regressor Score with Classical Regressor

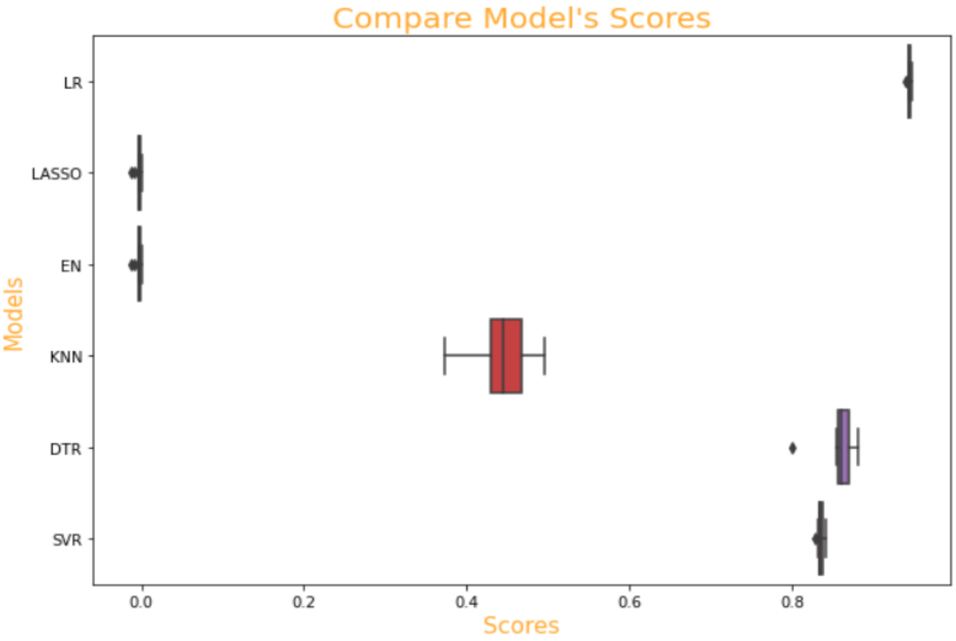


Figure 5.2 Classical Regression Models Comparison(new)

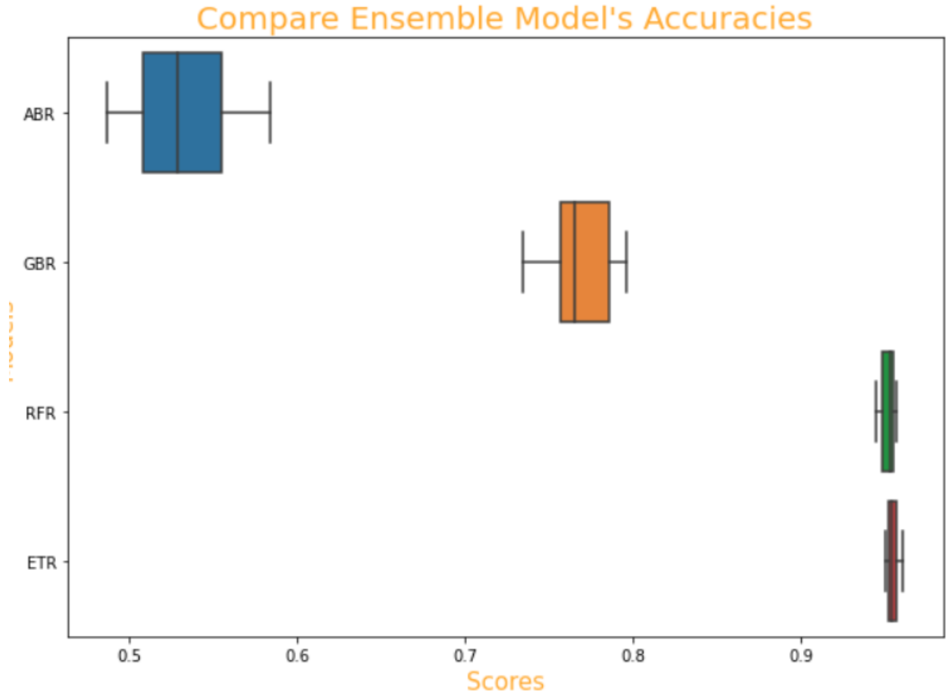


Figure 5.3 Ensemble Regression Models Comparison(new)

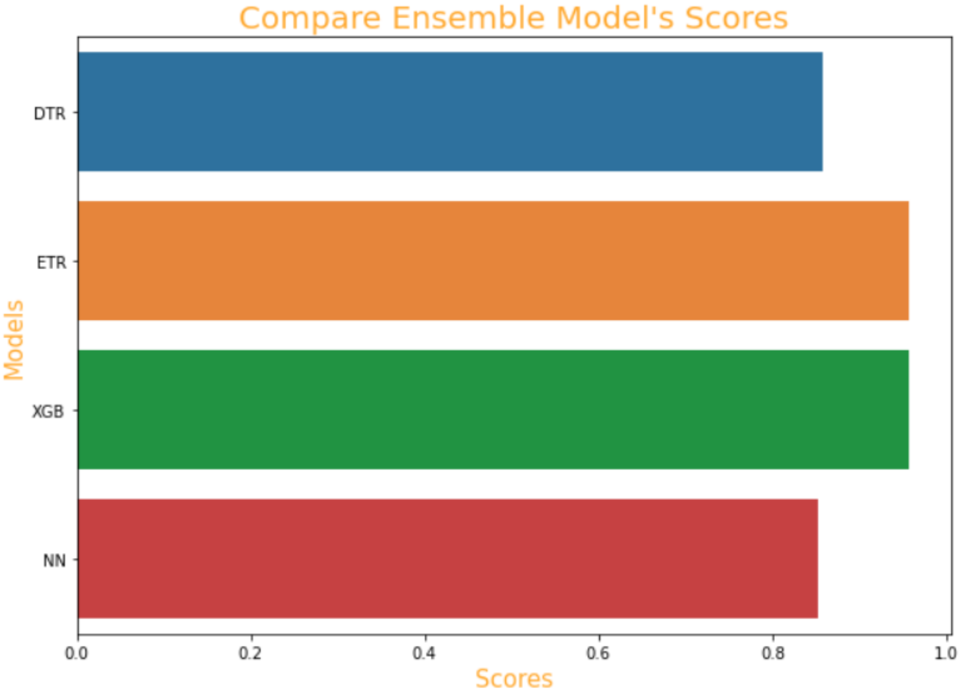


Figure 5.4 Compare Ensemble Regressor Score with Classical Regressor(new)