DataCamp

Data Science Certification

Case Study on

[“Coffee Quality Dataset Analysis”]

Capstone Project

Submitted by	Submitted to
Shusant Sapkota	DataCamp
shusant.sapkota@gmail.com	

Submitted on: July 20, 2021

1. Motivation

The coffee is prepared from roasted seed of berries belonging to Coffea species. In 2019/20, more than 166 million bags of coffee are consumed. Finland is the country where an individual consumes 12kg of coffee in average followed by Norway 9.9kg per annum. The quality of coffee is measured/rated in ten areas, they are Aroma, Flavor, Aftertaste, Acidity, Body, Balance, Uniformity, Clean Cup, Cupper Points, and Sweetness.

Aroma is fragrance we get from the brewed coffee, Flavor is the taste of coffee, Body also termed as mouthfeel is tactile impression on the palate when coffee coats tongue. Acidity refers to briny sensation on the tongue tip or tart taste near the back of tongue, Sweetness is the smoothness and mildness of coffee. The Aftertaste is final sensory experience when tasting a coffee. It is highly dependent upon other sensory quality attributes of coffee as mentioned above. Different researches show that, the quality is also impacted by moisture content, and bean defects.

This project aims in finding the major quality coffee attributes that affects the overall Aftertaste of coffee. The Dataset provided by DataCamp is used in analyzing the factors affecting Aftertaste. Finally, machine learning model is prepared to predict the Aftertaste value based on the factors that affect it. The findings of this project can be used to improve Aftertaste of a coffee and machine learning model can be used to predict the Aftertaste from the known attributes.

2. Success Criteria

Dataset

The dataset has been provided by DataCamp. It has 43 input features and a target variable namely Aftertaste. The columns can be categorized into three sub-categories

i. Quality Measures: Aroma, Flavor, Aftertaste, Acidity, Body, Balance, Uniformity, Clean Cup, Sweetness, Moisture, Defects
ii. Bean Metadata: Processing Method, Color, Species
iii. Farm Metadata: Owner, Country of Origin, Farm Name, Lot Number, Mill, Company, Altitude, Region

Out of these features, the quality measures are used in our analysis and model fitting.

Metrics

The metrics is used to evaluate the performance of our trained model. R2-score is used for evaluating the performance of our model. It is calculated using the following formula.

$$R^2 = 1 - \frac{\Sigma(y - \hat{y})^2}{\Sigma\left(y - \bar{y}\right)^2}$$

Where, ȳ – mean

ỹ - predicted value

The higher the value or R2, higher is the accurate performance of model.

3. Analysis Plan

This is a supervised regression problem because the target variable ‘Aftertaste’ is a continuous value. The following steps will be followed for solving this supervision problem:

i. Initially, after loading dataset unnecessary features and columns with most null values are dropped.
ii. Exploratory data analysis (EDA) is performed to discover initial insights of the data.
iii. Make appropriate visualization analysis about features with respect to target variables. From this appropriate feature are selected
iv. The features are used in training several models of different algorithms. The training phase is done in two ways, with more features and less features.
v. The different models are compared against each other in terms of training and testing accuracy. Best model is selected.
vi. Discuss the merits of and improvements of the model.

4. Exploratory Data Analysis

Importing Neccessary Libraries and Dataset

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor
from sklearn.externals import joblib
from sklearn.linear_model import Ridge
df = pd.read_csv('coffee.csv')
df.shape
```

Out[2]:
```
(1311, 44)
```

Dropping the columns which we're not going to use in Analysis

In [3]:
```
df.drop(['Species', 'Unnamed: 0', 'Owner', 'Owner.1', 'Producer', 'In.Country.Par
df.info()
```

```
---  ------               --------------  -----
 0   Country.of.Origin    1310 non-null   object
 1   Farm.Name            955 non-null    object
 2   Mill                 1001 non-null   object
 3   Company              1102 non-null   object
 4   Region               1254 non-null   object
 5   Number.of.Bags       1311 non-null   int64
 6   Bag.Weight           1311 non-null   object
 7   Harvest.Year         1264 non-null   object
 8   Aroma                1311 non-null   float64
 9   Flavor               1311 non-null   float64
 10  Aftertaste           1311 non-null   float64
 11  Acidity              1311 non-null   float64
 12  Body                 1311 non-null   float64
 13  Balance              1311 non-null   float64
 14  Uniformity           1311 non-null   float64
 15  Clean.Cup            1311 non-null   float64
 16  Sweetness            1311 non-null   float64
 17  Cupper.Points        1311 non-null   float64
 18  Total.Cup.Points     1311 non-null   float64
```

Dataset Summary

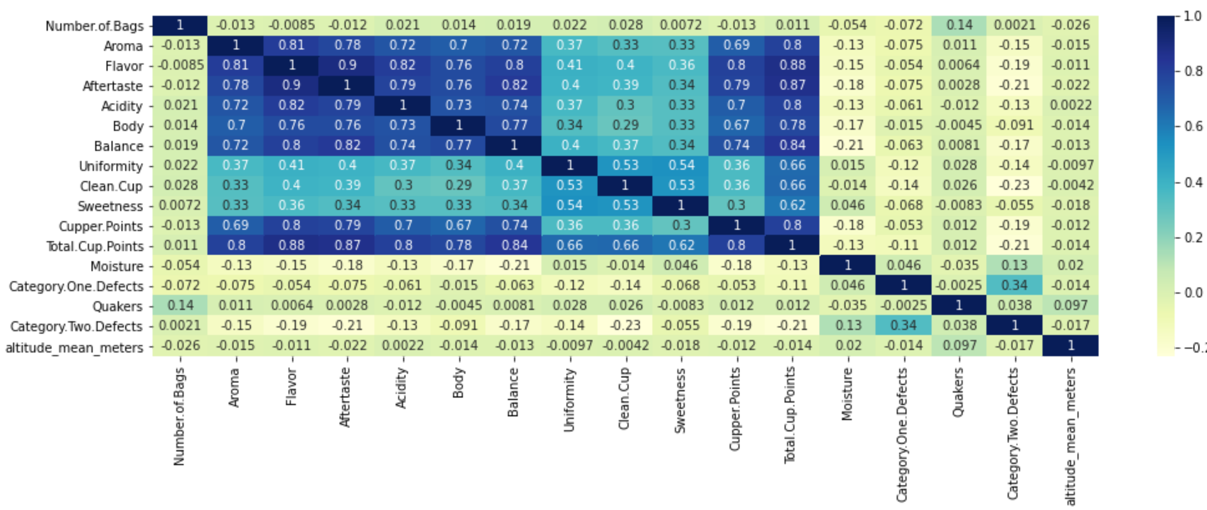
In [4]:
```
df.describe()
```

Out[4]:

	Number.of.Bags	Aroma	Flavor	Aftertaste	Acidity	Body	Balar
count	1311.000000	1311.000000	1311.000000	1311.000000	1311.000000	1311.000000	1311.000(
mean	153.887872	7.563806	7.518070	7.397696	7.533112	7.517727	7.5175
std	129.733734	0.378666	0.399979	0.405119	0.381599	0.359213	0.4063
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000(
25%	14.500000	7.420000	7.330000	7.250000	7.330000	7.330000	7.330(
50%	175.000000	7.580000	7.580000	7.420000	7.500000	7.500000	7.500(
75%	275.000000	7.750000	7.750000	7.580000	7.750000	7.670000	7.750(
max	1062.000000	8.750000	8.830000	8.670000	8.750000	8.580000	8.750(

In [5]:
```
plt.figure(figsize=(17,5))
sns.heatmap(df.corr(), annot=True, cmap="YlGnBu")
```

Out[5]: <AxesSubplot:>



In [6]:
```
bag_weight = []
for item in df['Bag.Weight']:
    if ' kg,lbs' in item:item = int(item.replace(' kg,lbs', ''))
    elif ' kg' in item:item = int(item.replace(' kg', ''))
    elif ' lbs' in item:
        item = int(item.replace(' lbs', ''))
        item = 0.453592*item
    elif ',lbs' in item:
        item = int(item.replace(',lbs', ''))
        item = 0.453592*item
    else:item = int(item)
    bag_weight.append(item)
df['Bag.Weight'] = pd.Series(bag_weight)
```

Feature Extraction:

Extracting the total quantity of coffee produced and droping individual columns

In [7]:
```
df['Total Quantity Produced kg'] = df['Number.of.Bags']*df['Bag.Weight']
df.drop(['Number.of.Bags', 'Bag.Weight'], axis=1, inplace = True)
```

Counting of Countries

In [8]:
```
df_countries = pd.DataFrame(df['Country.of.Origin'].value_counts()).reset_index()
df_countries.columns = ['Country', 'No. of Coffee Farms']
df_countries.head(5)
```

Out[8]:

	Country	No. of Coffee Farms
0	Mexico	236
1	Colombia	183
2	Guatemala	181
3	Brazil	132
4	Taiwan	75

Analyzing Region with Coffee Farms

In [9]:
```
df_region = pd.DataFrame(df['Region'].value_counts()).reset_index()
df_region.columns = ['Region Name', 'No. of Farms']
print(df_region.head(2))
a = list(df_region.head(1)['Region Name'])
a = list(df[df['Region'] == a[0]]['Country.of.Origin'])
print('Huila lies in, '+str(a[0]))
```

```
  Region Name  No. of Farms
0       huila           112
1     oriente            80
Huila lies in, Colombia
```

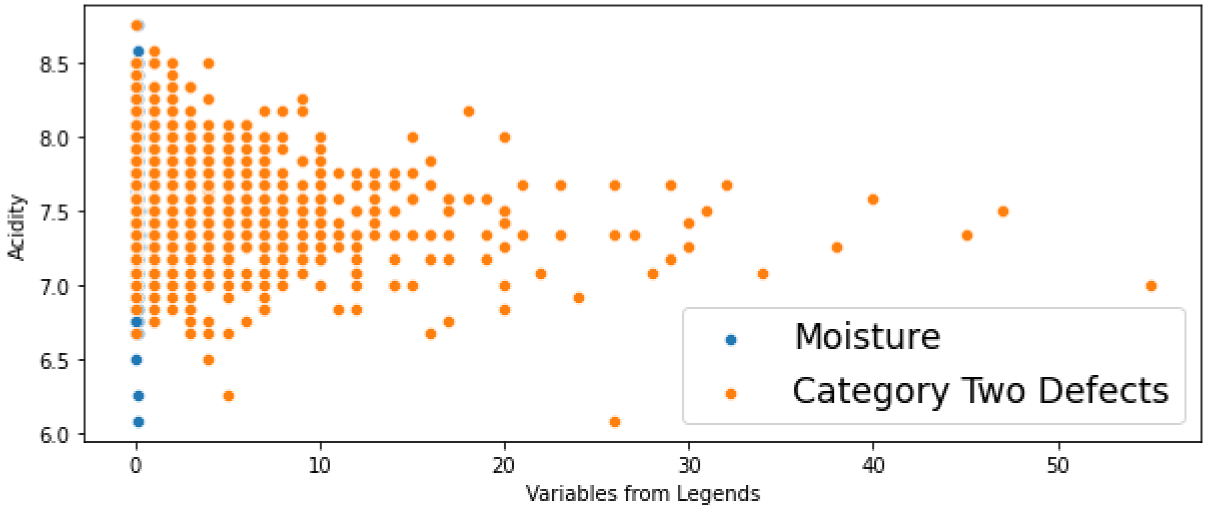
5. Visualization

Removing the outlier which is affecting out model efficieny. The outlier has most of coffee quality features as 0. It seems it is irrelevant data. So, we omit this row. Same for 'Acidity' and 'Body' feature.

In [10]:
```
df = df[df['Flavor']!=0]
df = df[df['Acidity']>5.50]
df = df[df['Body']>5.50]
```

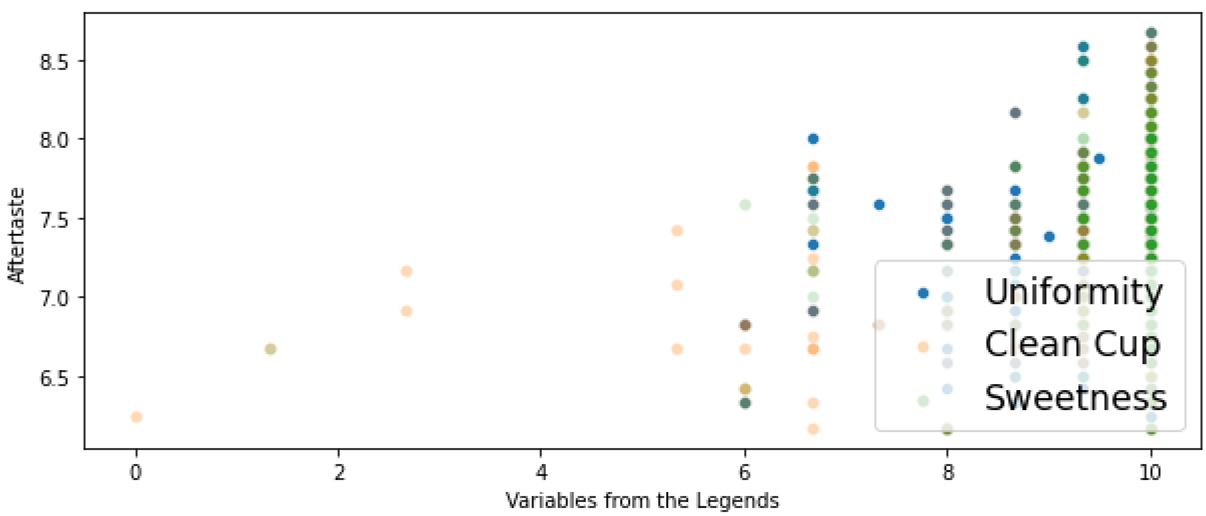
In [11]:

```
plt.figure(figsize=(10,4))
ax = sns.scatterplot(x = 'Moisture', y = 'Acidity', data = df)
sns.scatterplot(x = 'Category.Two.Defects', y = 'Acidity', data = df, markers='s'
plt.legend(['Moisture', 'Category Two Defects'], loc='lower right', fontsize='xx-
plt.xlabel('Variables from Legends')
plt.show()
```

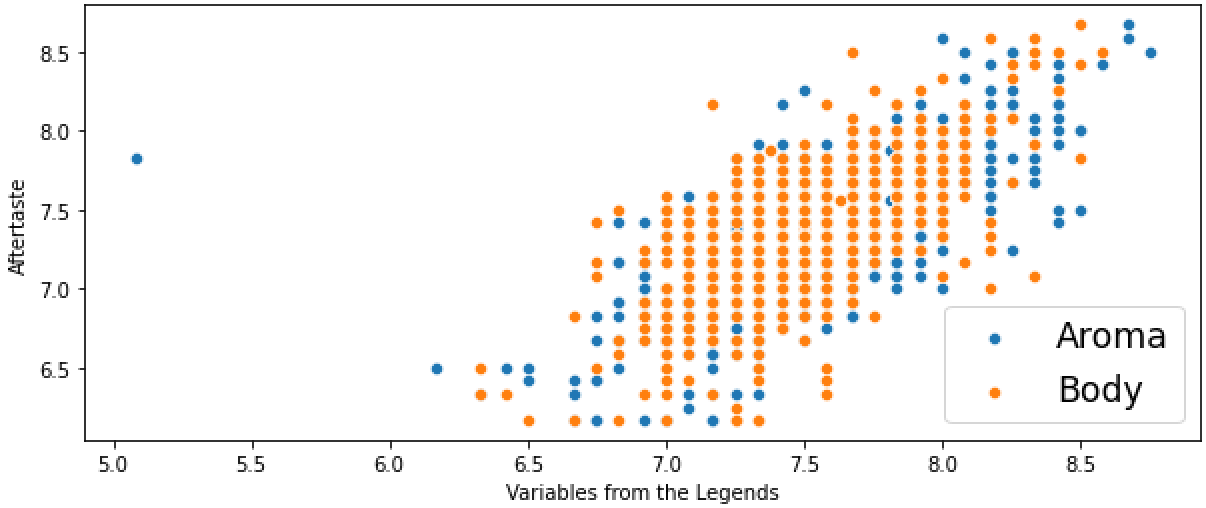


In [12]:

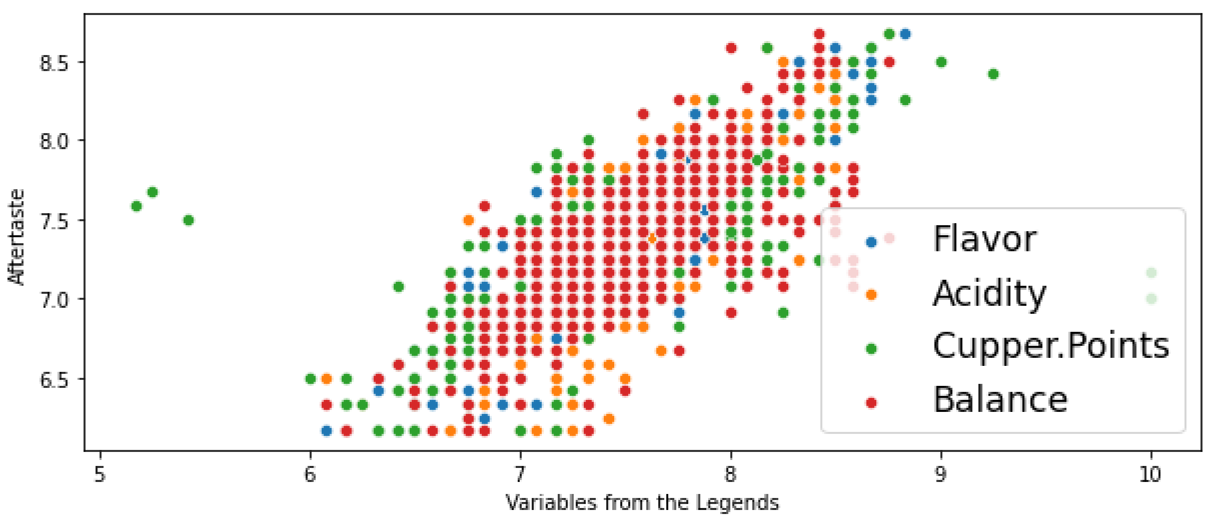
```
plt.figure(figsize=(10,4))
ax5 = sns.scatterplot(x = 'Uniformity' , y = 'Aftertaste', data = df)
ax6 = sns.scatterplot(x = 'Clean.Cup' , y = 'Aftertaste', data = df, alpha=0.3)
ax7 = sns.scatterplot(x = 'Sweetness' , y = 'Aftertaste', data = df, alpha=0.2)
plt.legend(['Uniformity', 'Clean Cup', 'Sweetness'], loc='lower right', fontsize=
plt.xlabel('Variables from the Legends')
plt.show()
```



In [13]:
```
plt.figure(figsize=(10,4))
ax = sns.scatterplot(x = 'Aroma', y = 'Aftertaste', data =df)
sns.scatterplot(x = 'Body', y = 'Aftertaste', data = df)
plt.legend(['Aroma', 'Body'], loc='lower right', fontsize='xx-large')
plt.xlabel('Variables from the Legends')
plt.show()
```



In [14]:
```
plt.figure(figsize=(10,4))
ax1 = sns.scatterplot(x = 'Flavor', y= 'Aftertaste', data=df)
ax2 = sns.scatterplot(x = 'Acidity' , y = 'Aftertaste', data = df)
ax3 = sns.scatterplot(x = 'Cupper.Points' , y = 'Aftertaste', data = df)
ax4 = sns.scatterplot(x = 'Balance' , y = 'Aftertaste', data = df)
plt.legend(['Flavor', 'Acidity', 'Cupper.Points', 'Balance'], loc='lower right',
plt.xlabel('Variables from the Legends')
plt.show()
```



In [15]:
```
final_df = df[['Aroma', 'Flavor', 'Aftertaste', 'Body', 'Balance', 'Uniformity','
final_df.to_csv('df_final.csv',index=False)
```

6. Discussion

Since, the analysis part was divided into two major parts; Exploratory Data Analysis and Visual Analysis (Visualization). The individual part has respective findings.Let's discuss the first part of our analysis. The initial raw dataset has 44 columns of different data types and 1311 observations. Most of the columns have NULL values. The columns which are not necessary for further processes are dropped including those with many NULL values. There are highest number of coffee farms in Mexico followed by Columbia, Guatemala, and Brazil. The region Huila has highest number of Coffee Farms which is 112 and it is located. Almost 61% of Columbian coffee farms are located in Huila region.

From EDA heatmap, we found that the 'Aftertaste' attribute for determining the coffee quality is majorly correlated with Aroma, Flavor, Acidity, Body, Balance, Uniformity, CleanCup, Sweetness, CupperPoints. The other non-quality features Moisture, Category Two Defects are also visualized against Aftertaste but found to be very slightly correlated. When above quality features are visualized against Aftertaste in scatter plot, we found that Aroma, Acidity, CupperPoints Flavor, Balance, and Body are most correlated features. So, we considered these features for Model Fitting.

7. Model Fitting

a. The models of different algorithms are fitted in two manner. Firstly, more features to fit our model

In [16]:
```
df = pd.read_csv('df_final.csv')
X = df[['Aroma', 'Acidity', 'Cupper.Points', 'Flavor','Balance', 'Body']]
y = df['Aftertaste']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
algorithms = ['LinearRegression', 'Decision Tree', 'Random Forest', 'KNNeighbors'
training_acc = []
testing_acc = []
```

In [17]:
```
# Linear Regression
model_lr = LinearRegression(normalize=True)
model_lr.fit(X_train, y_train)
training_acc.append(model_lr.score(X_train, y_train))
testing_acc.append(model_lr.score(X_test, y_test))
#Decision Tree
model_dt = DecisionTreeRegressor()
model_dt.fit(X_train, y_train)
training_acc.append(model_dt.score(X_train, y_train))
testing_acc.append(model_dt.score(X_test, y_test))
```

In [18]:
```
#Random Forest
model_rf = RandomForestRegressor(max_depth=5)
model_rf.fit(X_train, y_train)
training_acc.append(model_rf.score(X_train, y_train))
testing_acc.append(model_rf.score(X_test, y_test))
#KNN
model_knn = KNeighborsRegressor(n_neighbors=7)
model_knn.fit(X_train, y_train)
training_acc.append(model_knn.score(X_train, y_train))
testing_acc.append(model_knn.score(X_test, y_test))
#SVM
model_svr = SVR()
model_svr.fit(X_train, y_train)
training_acc.append(model_svr.score(X_train, y_train))
testing_acc.append(model_svr.score(X_test, y_test))
#XGBoost
model_x = XGBRegressor(n_estimators = 25, max_depth=2)
model_x.fit(X_train, y_train)
training_acc.append(model_x.score(X_train, y_train))
testing_acc.append(model_x.score(X_test, y_test))
```

b. The features are reduced to four, and fitted in the respective models

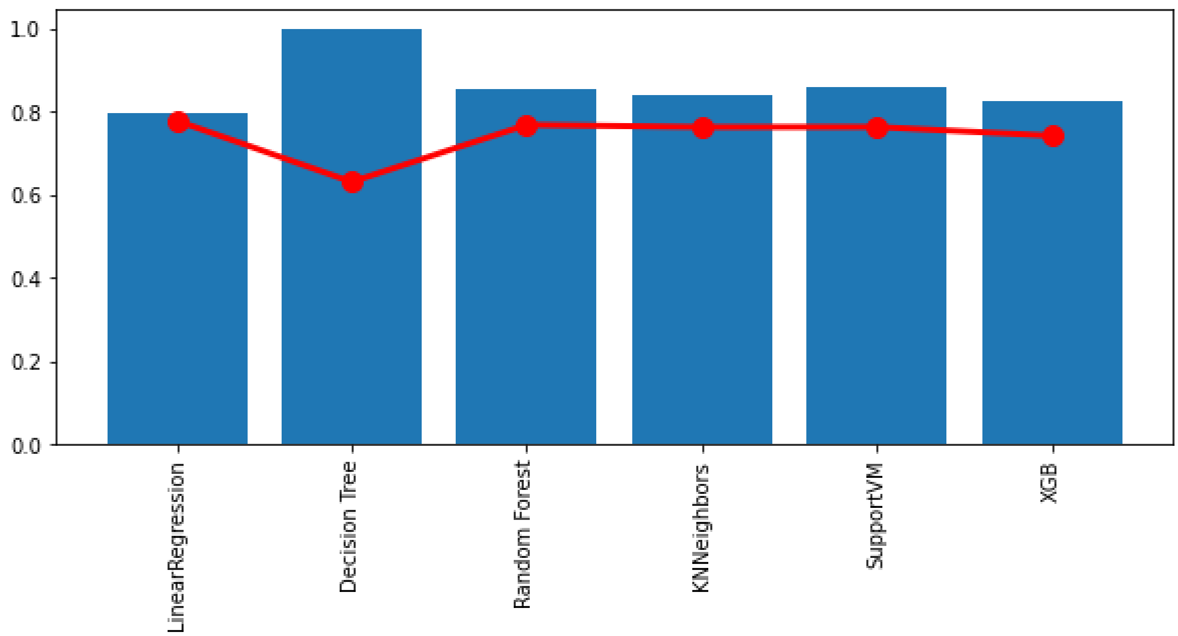
In [19]:
```
df = pd.read_csv('df_final.csv')
X = df[['Flavor','Balance', 'Acidity', 'Cupper.Points']]
y = df['Aftertaste']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
algorithms = ['LinearRegression', 'Decision Tree', 'Random Forest', 'KNNeighbors'
training_accl = []
testing_accl= []
# Linear Regression
model_lr = LinearRegression(normalize=True)
model_lr.fit(X_train, y_train)
training_accl.append(model_lr.score(X_train, y_train))
testing_accl.append(model_lr.score(X_test, y_test))
#Decision Tree
model_dt = DecisionTreeRegressor()
model_dt.fit(X_train, y_train)
training_accl.append(model_dt.score(X_train, y_train))
testing_accl.append(model_dt.score(X_test, y_test))
#Random Forest
model_rf = RandomForestRegressor(max_depth=5)
model_rf.fit(X_train, y_train)
training_accl.append(model_rf.score(X_train, y_train))
testing_accl.append(model_rf.score(X_test, y_test))
#KNN
model_knn = KNeighborsRegressor(n_neighbors=7)
model_knn.fit(X_train, y_train)
training_accl.append(model_knn.score(X_train, y_train))
testing_accl.append(model_knn.score(X_test, y_test))
#SVM
model_svr = SVR()
model_svr.fit(X_train, y_train)
training_accl.append(model_svr.score(X_train, y_train))
testing_accl.append(model_svr.score(X_test, y_test))
```

In [21]:
```
#XGBoost
model_x = XGBRegressor(n_estimators = 25, max_depth=2)
model_x.fit(X_train, y_train)
training_accl.append(model_x.score(X_train, y_train))
testing_accl.append(model_x.score(X_test, y_test))
```

8. Model Evaluation

Evaluation of Models trained with more Features and then less Features

In [22]:
```
plt.figure(figsize=(10,4))
ax = plt.bar(algorithms,training_acc)
plt.plot(algorithms, testing_acc, color='r', lw=3, marker='o', ms=10)
plt.xticks(rotation='vertical')
plt.show()
```

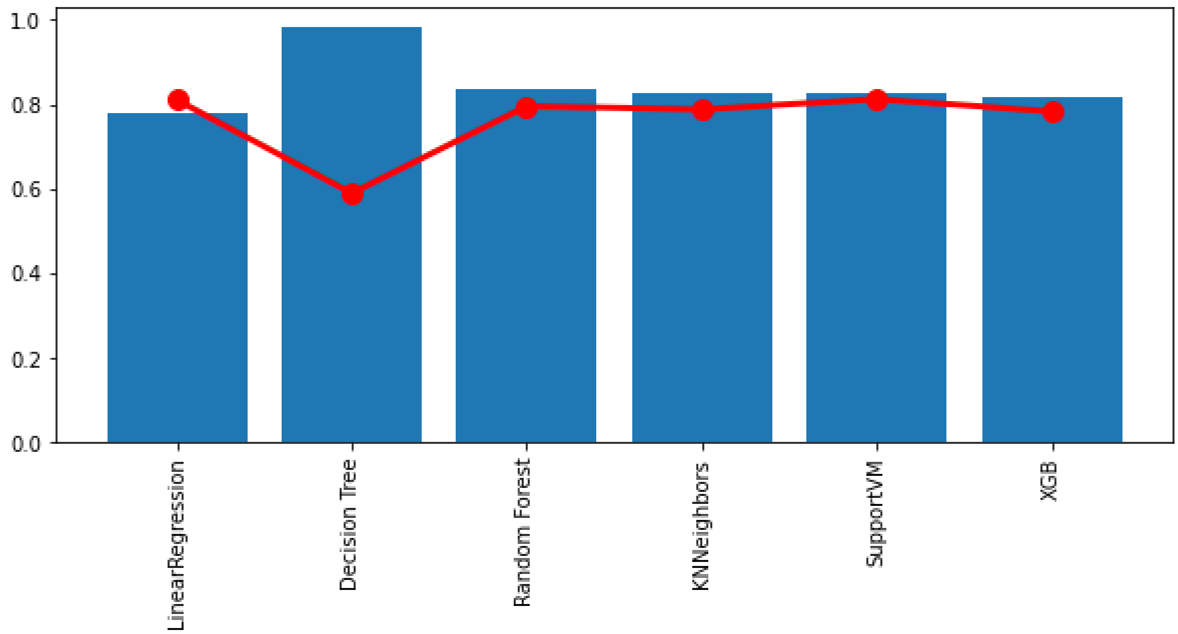


In [23]:
```
model_evaluation = pd.DataFrame(data = {'Model': algorithms, 'Training Accuracy':
model_evaluation.head(10)
```

Out[23]:

	Model	Training Accuracy	Testing Accuracy
0	LinearRegression	0.795596	0.778554
1	Decision Tree	0.999715	0.630847
2	Random Forest	0.852889	0.768431
3	KNNeighbors	0.839714	0.763077
4	SupportVM	0.858587	0.763344
5	XGB	0.825549	0.742055

In [24]:
```
plt.figure(figsize=(10,4))
ax = plt.bar(algorithms,training_accl)
plt.plot(algorithms, testing_accl, color='r', lw=3, marker='o', ms=10)
plt.xticks(rotation='vertical')
plt.show()
```



In [25]:
```
model_evaluation = pd.DataFrame(data = {'Model': algorithms, 'Training Accuracy':
model_evaluation.head(10)
```

Out[25]:

	Model	Training Accuracy	Testing Accuracy
0	LinearRegression	0.779179	0.810678
1	Decision Tree	0.983414	0.589301
2	Random Forest	0.836610	0.795524
3	KNNeighbors	0.827339	0.788099
4	SupportVM	0.828582	0.812459
5	XGB	0.816722	0.783056

Saving Best Model

In [26]:
```
joblib.dump(model_rf, 'model_svm.pkl')
```

Out[26]: ['model_svm.pkl']

9. Results

From the Exploratory analysis, Aftertaste of coffee is largely dependent upon these attributes Flavor, Acidity, CupperPoints, Balance, Aroma, Body, Uniformity, Clean Cup, Sweetness, Moisture, and Category Two Defects. When these attributes are visually analyzed, Moisture Category Two Defects, Uniformity, Clean Cup, and Sweetness have minimal impact in Aftertaste. Hence, Flavor, Acidity, CupperPoints, Balance, Body, and Aroma affects the Aftertaste of coffee largely.

A dataframe with these final features are selected along with target variable. This frame is divided into training and testing set. The training set is used in training six different machine learning algorithms. Each model are hyper tuned manually. These trained models are evaluated by both training set and testing set using R-square approach. Most of the model gets overfitted and this problem is minimal in Linear Regression with training accuracy 79.6% and testing accuracy 77.9%.

In second case, we take only four features dropping Aroma and Body. Again, the models are trained and evaluated similarly. The problem of overfitting is minimal in most of the models except Decision Tree. The Support Vector Machine performs better than other. It's training and testing accuracy of is 82.9% and 81.2% respectively and is saved for future purposes.

10. Future Work

The dataset has 44 columns, we've used mainly the quality features to study the factors affecting the Aftertaste. There are many other features of Bean Metadata and Farm metadata. Future enhancements of this project are listed below:

i. These features can be used to identify the Country, Farm producing the utmost quality Coffee.
ii. The mill processing good quality coffee can identified.
iii. The quality of coffee based upon its color, and processing method can be found.

These future findings might be beneficial for coffee exporters, sellers, and even countries where people consume maximum coffee in average.

11. References

API reference. API reference - seaborn 0.11.1 documentation. (n.d.). https://seaborn.pydata.org/api.html.

User Guide. User Guide - pandas 1.3.0 documentation. (n.d.). https://pandas.pydata.org/docs/user_guide/index.html.

User Guide. scikit. (n.d.). https://scikit-learn.org/stable/user_guide.html.