Abalone dataset

Problem description:

The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

Attribute Information

Given is the attribute name, attribute type, the measurement unit and a brief description. The number of rings is the value to predict.

Name / Data Type / Measurement Unit / Description

Sex / nominal / -- / M, F, and I (infant)
Length / continuous / mm / Longest shell measurement
Diameter / continuous / mm / perpendicular to length
Height / continuous / mm / with meat in shell
Whole weight / continuous / grams / whole abalone
Shucked weight / continuous / grams / weight of meat
Viscera weight / continuous / grams / gut weight (after bleeding)
Shell weight / continuous / grams / after being dried
Rings / integer / -- / +1.5 gives the age in years.

You have to predict the rings of each abalone which will lead us to the age of that abalone.

Abalone Case Study Project:

```
]:
import warnings
warnings.simplefilter("ignore")
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from math import sqrt
import scipy.stats as stats
from scipy.stats import zscore

from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import SGDRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingRegressor

from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

import pickle
```

I am importing the all library which I required for EDA, visualization, prediction and finding all matrices. The reason of doing this is that it become easier to use all the import statement at one go and we do not require to import the statement again at each point. We could find all the importing statement at one place without finding it on whole notebook and can update also.

Loading Data Set into variable:

```
ds = pd.read_csv("https://raw.githubusercontent.com/roni96007/ProjectsDatatrained/main/Practice_Projects/Project%204/Abalone.csv")
```

```
ds
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	7
...	...	...	...	...	...	...	...	...	...
4172	F	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11
4173	M	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10
4174	M	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9
4175	F	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10
4176	M	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12

4177 rows × 9 columns

Here I am loading the data set into a variable i.e. “df” and processing the first 5 rows.As in this data set most of the column are float in nature and type and sex is of categorical value.

Data Analysis:

```
[76]: ds.columns

[76]: Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
       'Viscera weight', 'Shell weight', 'Rings'],
      dtype='object')

[4]: ds.shape

[4]: (4177, 9)

[5]: ds.isnull().sum()

[5]: Sex               0
     Length            0
     Diameter          0
     Height            0
     Whole weight      0
     Shucked weight    0
     Viscera weight    0
     Shell weight      0
     Rings             0
     dtype: int64
```

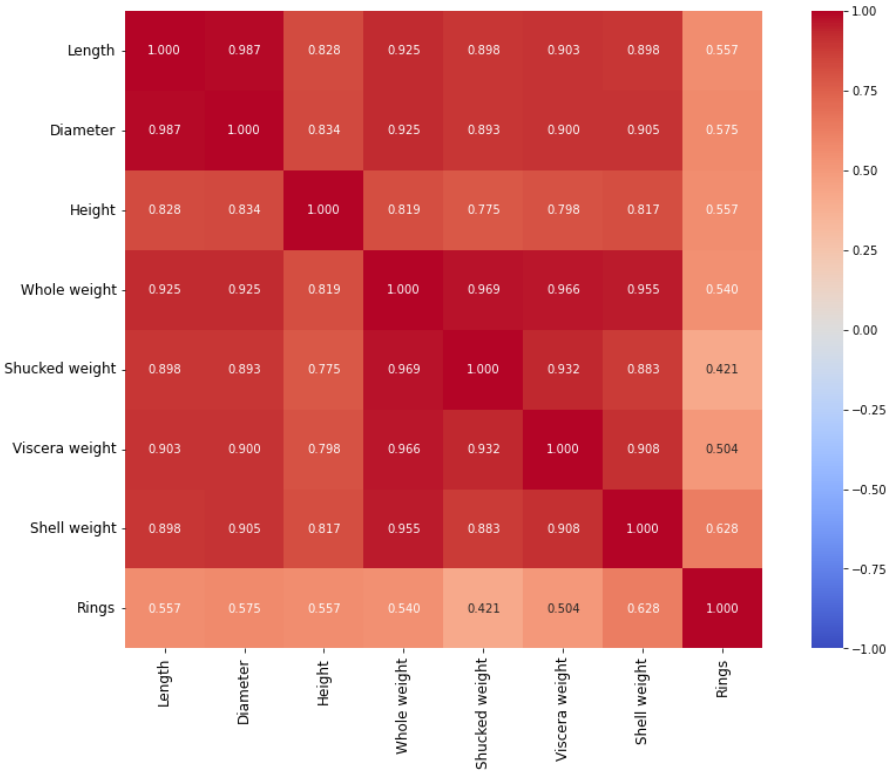
- As seen in data set there is one ‘Unnamed:0’ column which does not play any important role for prediction in the price of avocado, so I am dropping that column.

- Also, I am checking the shape of the data set as there are 1517 rows and 13 column s after deleting the index column.
- As we there are no null vales present in the dataset.
- Also, most of the column are of same data type that is float type ,region and date a re of object data type & year is of int type.

In above, I am finding that year 2017 is aggressive year where avocado price is higher as compared to other year and 2015 is at second number.

Also, I am finding that at each year present in the data set, which type of avocado is has total count, so both type of avocado is present almost in same amount in the data set.

Stastical summary



Above statistics data show that their multiple outliers mostly in XLargeBags There is also difference between mean and 50% value in some of the columns which used to get fix for better prediction

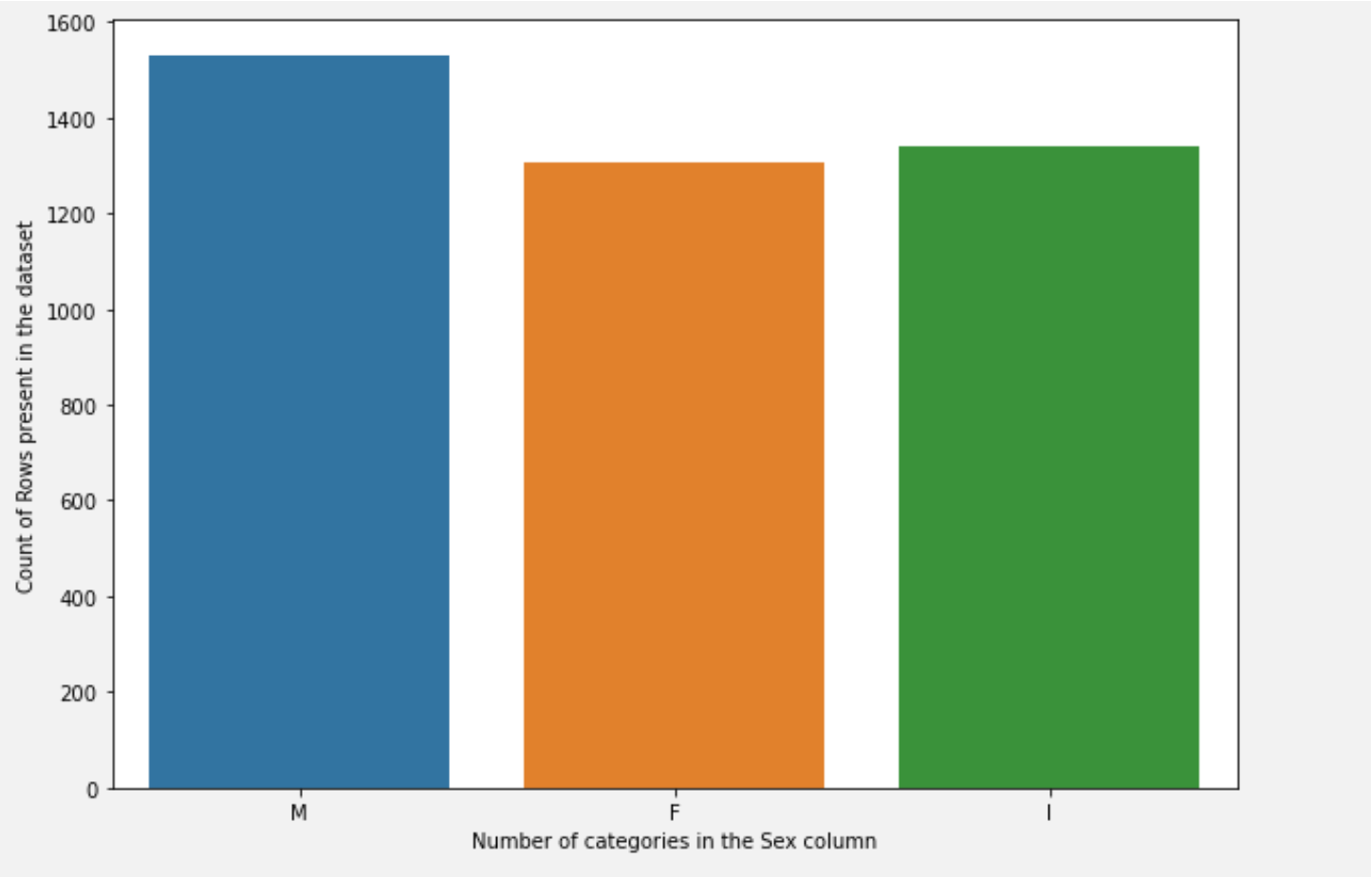
- Also, number of rows in each column are same, means there are no null values in the data set.
- Also, the mean and 50%value of most of the column are same and the STD and mean are very close to each other.
- Most of the column statistics data are near to 0 values.
- By checking the difference between the 75% and max value there are outliers in some of the column, I will check it soon.

Exploratory Data Analysis:

In this portion we can plot different graph using different columns and try to visualize the data using matplotlib and seaborn library.

We use different graph include:

- Bar plot
- Count plot
- Line plot
- Histogram and Pair plot



From above we came to know that:

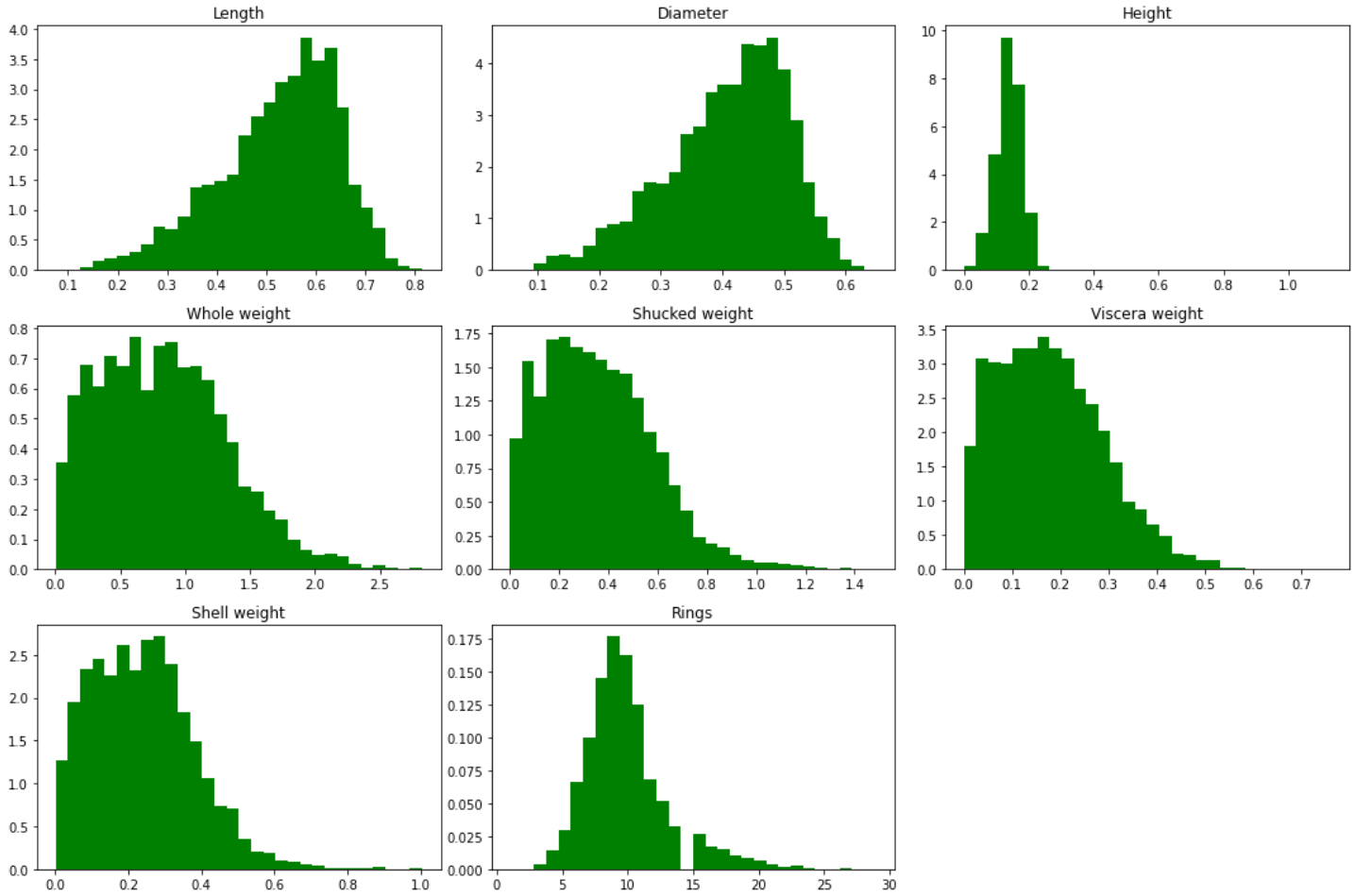
- Year 2017 is that year where the price is maximum as compared to other year, and there is less difference among rest of the year.
- September and October are the month where max no of average price is there, but the thing is almost for whole year the price is almost same for the avocado this prove that there is so much craze of avocado rather than india.

Plotting Histogram:

- A **histogram** shows the frequency on the vertical axis and the horizontal axis is another dimension. Usually it has bins, where every bin has a minimum and maximum value. Each bin also has a frequency between x and infinite
- So, in this we can also check whether the graph is right skewed, left skew or the graph is normally distributed graph.

From plotting this histogram, I used the bin size as 30, we can take any bin size (suited as per as data).

- Average price column is normally distributing over the histogram.
- Rest of the data are not much varying in term of numbers, so they are almost left skewed data
- To make the column as normal distributed we can use different methods, but I am using numPy log to make the skew values as normal distributed.



- ## Correlation Matrix:

- **Correlation** Matrix is basically a covariance matrix. A summary measure called the **correlation** describes the strength of the linear association. **Correlation** summarizes the strength and direction of the linear (straight-line) association between two quantitative variables. Denoted by r, it takes values between -1 and +1.

- Now I am finding the correlation value of each column, this value is categorized into mainly 2 parts that are:
- - Positive correlated value
- - Negative correlated value
- The most the value is positive means that column is much co related and vice versa.
- I am using seaborn heatmap to plot the correlated matrix and plot the corr value in the heatmap graph

Checking outliers:

From above we can say that many outliers are present in cloumns.

An **outlier** is a data point in a data set that is distant from all other observations. A data point that lies outside the overall distribution of the data set

Now that we know outliers can either be a mistake or just variance, how would you decide if they are important or not. Well, it is simple if they are the result of a mistake, then we can ignore them, but if it is just a variance in the data, we would need think a bit further.

From above image we can clear see that there are number of black dots in most of the column which are referring to the outliers, so it means most of the data are outside the distribution.

So now we detect the outliers now the second step is to remove the outliers, there are different way to remove the outliers that are find the IQR, zscore values.

I am using both zscore value then I again check if there are some of the outliers then I will remove it by replacing the outliers with the mean value of that column.

```
# Z score

z=np.abs(zscore(ds))
threshold=3
np.where(z>3)

ds=ds[(z<3).all(axis=1)]
ds
```

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Sex_F	Sex_I	Sex_M
0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	15	0	0	1
1	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7	0	0	1
2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9	1	0	0
3	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10	0	0	1
4	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	7	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
4172	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11	1	0	0
4173	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10	0	0	1
4174	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9	0	0	1
4175	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10	1	0	0
4176	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12	0	0	1

4027 rows × 11 columns

So, I first find the zscore value and then I decide to make one threshold value as 3 which is standard of industry recommend value and then I remove all the outliers which zscore value is greater than 3.

Pre Processing Pipeline:

Separating data variable and independent variable i.e. Average price

Drop and Standard Scaler:

Here I am making two variable x and y where x is having all column except Average Price and Date, we can also drop the Date column, but I kept for EDA purpose and y is having only Rings column.

Also, I am using the standard scaling method on x variable

Building Machine Learning Models:

Machine Learning Model for Regression and Evaluation Metrics

```
# Regression Model Function

def reg(model, X, Y):
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=21)

    # Training the model
    model.fit(X_train, Y_train)

    # Predicting Y_test
    pred = model.predict(X_test)

    # RMSE - a lower RMSE score is better than a higher one
    rmse = mean_squared_error(Y_test, pred, squared=False)
    print("RMSE Score is:", rmse)

    # R2 score
    r2 = r2_score(Y_test, pred, multioutput='variance_weighted')*100
    print("R2 Score is:", r2)

    # Cross Validation Score
    cv_score = (cross_val_score(model, X, Y, cv=5).mean())*100
    print("Cross Validation Score:", cv_score)

    # Result of r2 score minus cv score
    result = r2 - cv_score
    print("R2 Score - Cross Validation Score is", result)
```

Above I am using the for loop which help me to provide the R2 score at each random state and for the best state where R2 score is maximum is come as output value.