Table of context

**1. Problem Definition**

**2. Data Analysis/EDA**

**3. Pre-processing Pipeline**

**4. Building Machine Learning Models**

**5. Concluding Remarks**

1. Problem Definition

The model is to find the average price of avocadoes.

This data was downloaded from the Hass Avocado Board website in May of 2018 & compiled into a single CSV.

The table below represents weekly 2018 retail scan data for National retail volume (units) and price. Retail scan data comes directly from retailers’ cash registers based on actual retail sales of Hass avocados.

Starting in 2013, the table below reflects an expanded, multi-outlet retail data set. Multi-outlet reporting includes an aggregation of the following channels: grocery, mass, club, drug, dollar and military. The Average Price (of avocados) in the table reflects a per unit (per avocado) cost, even when multiple units (avocados) are sold in bags.

The Product Lookup codes (PLU’s) in the table are only for Hass avocados. Other varieties of avocados (e.g. greenskins) are not included in this table.

Some relevant columns in the dataset:

Date - The date of the observation

AveragePrice - the average price of a single avocado

type - conventional or organic

year - the year

Region - the city or region of the observation

Total Volume - Total number 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

**This is a linear regression problem.**

2. Data Analysis/EDA

#linear algebra

import numpy as np

# data processing

import pandas as pd

# data visualization

import seaborn as sns

import matplotlib.pyplot as plt

# Algorithms

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import cross\_val\_score

from sklearn.linear\_model import Lasso

from sklearn.ensemble import RandomForestRegressor

from sklearn import metrics

from sklearn.tree import DecisionTreeRegressor

import joblib

import warnings

warnings.filterwarnings("ignore")

There are total of 16468 rows and 14 columns

After the analysis of data there are only 1517 rows apart from that all the values are Nan values so deleted those rows.

Dropped the Unnamed: 0 column as it has representation of indexes

Date column has been removed and month and day column are added new to the dataset.

Dropped the type column as it has only one unique value as conventional.

**df1.info()**

**<class 'pandas.core.frame.DataFrame'>**

**RangeIndex: 1517 entries, 0 to 1516**

**Data columns (total 15 columns):**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

**0 Unnamed: 0 1517 non-null float64**

**1 AveragePrice 1517 non-null float64**

**2 Total Volume 1517 non-null float64**

**3 4046 1517 non-null float64**

**4 4225 1517 non-null float64**

**5 4770 1517 non-null float64**

**6 Total Bags 1517 non-null float64**

**7 Small Bags 1517 non-null float64**

**8 Large Bags 1517 non-null float64**

**9 XLarge Bags 1517 non-null float64**

**10 type 1517 non-null object**

**11 year 1517 non-null float64**

**12 region 1517 non-null object**

**13 Month 1517 non-null int64**

**14 Day 1517 non-null int64**

**dtypes: float64(11), int64(2), object(2)**

**memory usage: 177.9+ KB**

There was value as "0.00" present in XLarge Bags column and Large Bags column with huge amount which has to be replaced either by mean or median. Basically both the columns are to be replaced by median as the graph is not normally distributed. But the value XLarge Bags column was replaced by mean because the XLarge Bags column median value as coming zero.

3. Pre-processing Pipeline

**Skeweness removal**

There is skewness present in 'Total Volume', '4046', '4225', '4770', 'Total Bags','Small Bags', 'Large Bags', 'XLarge Bags', 'year'.

**df1.skew()**

**AveragePrice -0.109444**

**Total Volume 6.200138**

**4046 6.051830**

**4225 6.394926**

**4770 5.405164**

**Total Bags 5.366378**

**Small Bags 5.355185**

**Large Bags 5.154399**

**XLarge Bags 5.958090**

**year 1.828332**

**region 0.288146**

**Month 0.101439**

**Day 0.041303**

**dtype: float64**

After removing the skewness through log transforation and cube root transformation.

**df1.skew()**

**AveragePrice -0.109444**

**Total Volume 0.667461**

**4046 -0.160523**

**4225 0.184359**

**4770 1.526943**

**Total Bags 0.695445**

**Small Bags 0.713786**

**Large Bags -0.546940**

**XLarge Bags -1.217555**

**year 1.828332**

**region 0.288146**

**Month 0.101439**

**Day 0.041303**

**dtype: float64**

**Outliers Removal**

First found the outliers present in which columns through boxplot. Then found that outliers present in "Total Volume","4046","4225","4770","Total Bags","Small Bags","Large Bags","XLarge Bags".

Tried to remove outliers through zscore and IQR:

--> 2% of data loss is there while removing outliers with zscore

--> 26% data loss is there while removing outliers with IQR

So we will go with less data loss i.e. ZSCORE method

Data was scalled through StandardScaler

4.      Building Machine Learning Models

4 models are tried to find the best accuracy. Linear regression, lasso(with grid search cv and found the best parameters), Random forest regressor(with grid search cv and found the best parameters) and decision tree regressor.

Found the best model Random forest regressor with below details:

**rf=RandomForestRegressor(criterion="mse",max\_features="log2")**

**rf.fit(x\_train,y\_train)**

**rf.score(x\_train,y\_train)**

**pred\_decision=rf.predict(x\_test)**

**rfs=r2\_score(y\_test,pred\_decision)**

**print("R2 score:",rfs\*100)**

**print('MAE:', metrics.mean\_absolute\_error(y\_test, pred\_decision))**

**print('MSE:', metrics.mean\_squared\_error(y\_test, pred\_decision))**

**print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, pred\_decision)))**

**r2score as 79.59569911112285**

**MAE: 0.059232542372881385**

**MSE: 0.0069104144406779726**

**RMSE: 0.08312890255908574**

The final object was made as Avacado\_Final.obj

5. Concluding Remarks

Random Forest

What is Random Forest ?

Random Forest is a supervised learning algorithm. Like you can already see from it’s name, it creates a forest and makes it somehow random. The forest it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. With a few exceptions a random-forest classifier has all the hyperparameters of a decision-tree classifier and also all the hyperparameters of a bagging classifier, to control the ensemble itself.

The random-forest algorithm brings extra randomness into the model, when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better model. Therefore when you are growing a tree in random forest, only a random subset of the features is considered for splitting a node. You can even make trees more random, by using random thresholds on top of it, for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

The best model is Random Forest Regressor with criterion = "mse" and max\_features = "log2"