WORKPLACE PROJECT

Avocado Prices and Sales Volume 2015-2023

TABLE OF CONTENTS

1. Importing Packages

2. Data Collection and Description

3. Loading Data

4. Data Cleaning and Filtering

5. Exploratory Data Analysis (EDA)

6. Modeling

7. Evaluation and Validation

8. Final Model

9. Conclusion and Future Work

10. References

IMPORTING PACKAGES

- **Purpose:** Set up the Python environment with necessary libraries and tools.
- **Details:** List and import all the Python packages that will be used throughout the project such as Pandas for data manipulation, Matplotlib/Seaborn for visualization, scikit-learn for modeling, etc.

```
import pandas                as pd
import numpy                 as np
import matplotlib.pyplot     as plt
import plotly.express        as px
import seaborn as sns
import scipy.stats as stats
from sklearn.preprocessing        import MinMaxScaler
from sklearn.preprocessing        import StandardScaler
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.ensemble             import StackingRegressor
from sklearn.metrics              import mean_squared_error, r2_score
from sklearn.preprocessing        import LabelEncoder
import warnings
warnings.filterwarnings("ignore", category=FutureWarning, module="seaborn")
```

DATA DESCRIPTION

- **Purpose:** Data provided here was collected from Hass Avocado Board - Category Data. This is an updated version of dataset already introduced to Kaggle.
- **Details:** The {avocado} package consists of three different datasets that summarize the weekly sales of Hass Avocados at different regional levels.evell).

Dataset Features:

The hass_usa dataset focuses on weekly Hass Avocado sales at the country (i.e., contiguous US) level and consists of the following fields:

- week_ending: The date of the last day of the week in YYYY-MM-DD format
- avg_price_nonorg: The average selling price in US
  $for non-organic Hass avocados. Not adjusted for inflation - plu4046 : The amount of PLU 4046 Hass avocados sold by weight in US pounds. This does not inclu$
  for organic Hass avocados. Not adjusted for inflation
- plu94046: The amount of PLU 94046 Hass avocados sold by weight in US pounds. This does not include avocados sold in pre-packaged quantities. See vignette for more information about Hass PLUs
- plu94225: The amount of PLU 94225 Hass avocados sold by weight in US pounds. This does not include avocados sold in pre-packaged quantities. See vignette for more information about Hass PLUs
- plu94770: The amount of PLU 94770 Hass avocados sold by weight in US pounds. This does not include avocados sold in pre-packaged quantities. See vignette for more information about Hass PLUs
- small_org_bag: The amount of organic Hass avocados (they can be a mix of PLUs) sold in small pre-packaged containers/bags in US pounds. This does not include avocados sold individually. See vignette for more information about pre-packaged avocados
- large_org_bag: The amount of organic Hass avocados (they can be a mix of PLUs) sold in large pre-packaged containers/bags in US pounds. This does not include avocados sold individually. See vignette for more information about pre-packaged avocados
- xlarge_org_bag: The amount of organic Hass avocados (they can be a mix of PLUs) sold in extra-large pre-packaged containers/bags in US pounds. This does not include avocados sold individually. See vignette for more information about pre-packaged avocados.

LOADING DATA

```
df = pd.read_csv("C:/Users/nkabinmf/OneDrive - Vodafone Group/Data Science Course/Avocado_HassAvocadoBoard_20152023v1.0.1.csv") #Load the data into the n
df = pd.DataFrame(df)
df.head() #display the first few rows to give a sense of what the raw data looks like
```

	Date	AveragePrice	TotalVolume	plu4046	plu4225	plu4770	TotalBags	SmallBags	LargeBags	XLargeBags	type	region
0	2015-01-04	1.22	40873.28	2819.50	28287.42	49.90	9716.46	9186.93	529.53	0.0	conventional	Albany
1	2015-01-04	1.79	1373.95	57.42	153.88	0.00	1162.65	1162.65	0.00	0.0	organic	Albany
2	2015-01-04	1.00	435021.49	364302.39	23821.16	82.15	46815.79	16707.15	30108.64	0.0	conventional	Atlanta
3	2015-01-04	1.76	3846.69	1500.15	938.35	0.00	1408.19	1071.35	336.84	0.0	organic	Atlanta
4	2015-01-04	1.08	788025.06	53987.31	552906.04	39995.03	141136.68	137146.07	3990.61	0.0	conventional	BaltimoreWashington

DATA CLEANING AND FEATURE ENGINEERING

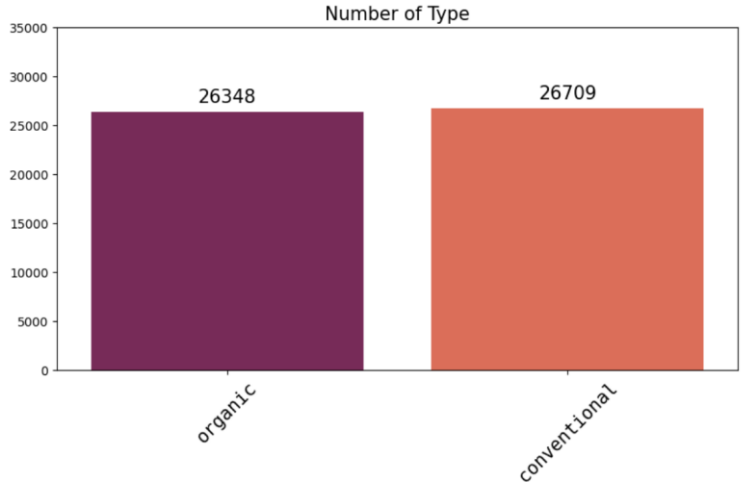
Data Cleaning

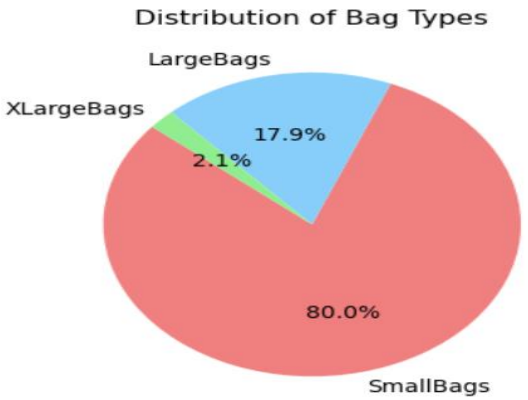
Activity	Results
df.isnull().sum()	SmallBags 12390 LargeBags 12390 XLargeBags 12390
print(f"Number of duplicated rows: {duplicate_count}")	Number of duplicated rows: 0
print(f"Number of outliers in AveragePrice: {outliers_price.sum()}")	Number of outliers in AveragePrice: 358
df = df[~outliers_price]	Identified outliers and removed them and now we are left with 53057 rows.

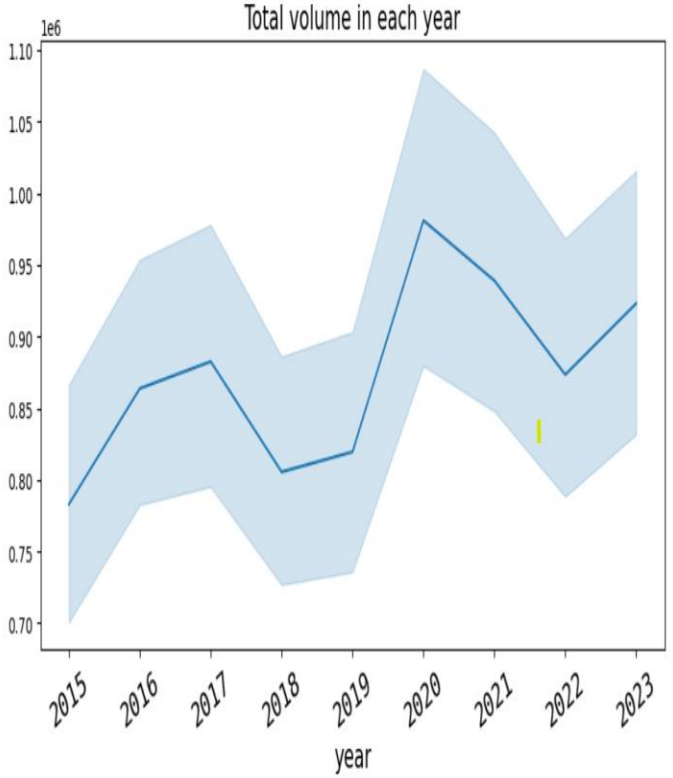
Feature Engineering

Handling Missing Values	
df_filled = df.fillna(0)	Number of missing values: 0
converting Date column to datetime format in order to create "Year" variable	Date to Year
Converting categorical data	type to type_encoded Regions to Regions_Count

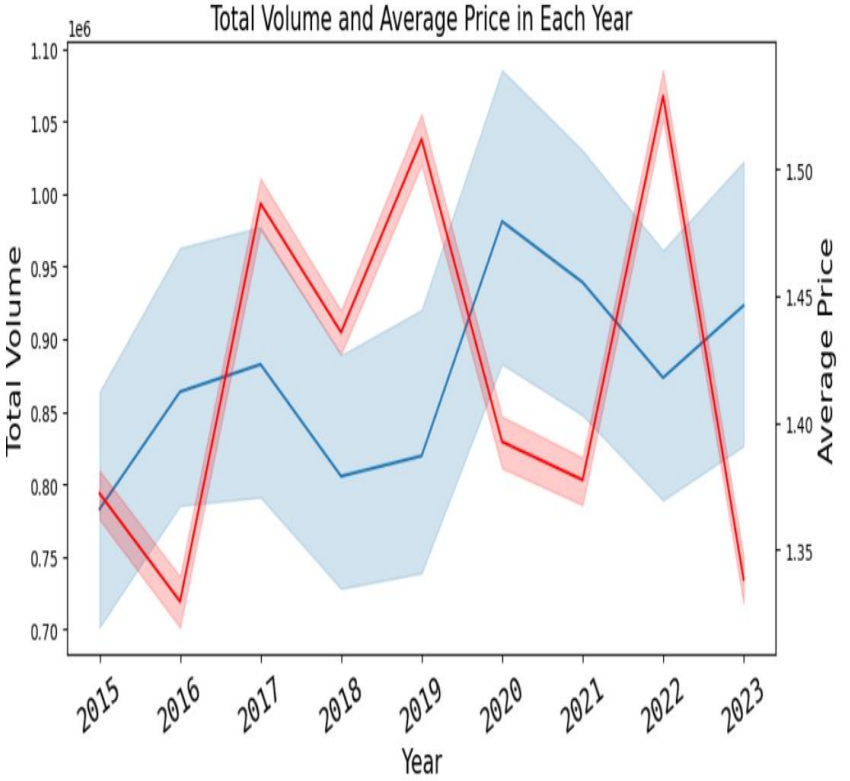
EXPLORATORY DATA ANALYSIS (EDA)







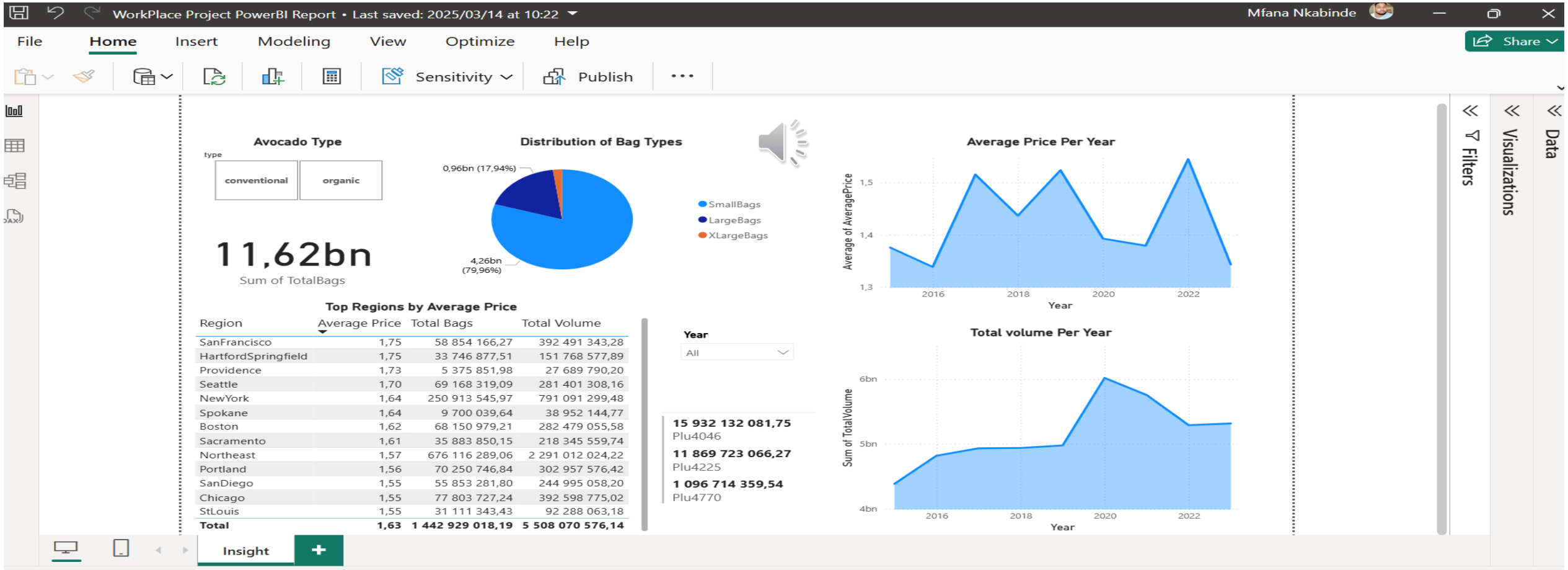
`TotalVolume` per `Year` shows a high increase in avocado volume around `Year` `2020` which creates an Hyphothesis of Low `TotalVolume` innitiates also High `AveragePrice` for the same `Year`.



Zooming in on `AveragePrice` and `TotalVolume` we see exactly what we have hyphothesised, Low Volume of Avocados leads to High Price Vise Versa.

C:

POWER BI REPORT



MODELLING

Model	Mean Absolute Error (mae)	Root Mean Squared Error (RMSE)
Model 1 - Decision Tree	0.2576	0.5075
Model 2 - Random Forest	0.1998	0.4470
Model 3 - Linear Regression	0.2199	0.5075

Final Model -Random Forest Regressor:

- **Reason for Selection:**

The Random Forest Regressor outperformed both the Decision Tree and Linear Regression models.

It demonstrated the lowest RMSE (0.4471) and MSE (0.1999), indicating better generalization to unseen data.

The Random Forest Regressor was selected due to its superior predictive accuracy and ability to generalize well across unseen data.

CONCLUSION AND FUTURE WORK

- **Summary of Results:** : Through rigorous evaluation, the Random Forest Regressor emerged as the best-performing model, demonstrating superior generalization and predictive accuracy compared to the Decision Tree and Linear Regression models. With the lowest RMSE (0.4471) and MSE (0.1999), it is well-suited for addressing the given regression task effectively.

- **Key Insights Gained:** :

EDA: Avocado's average price is driven by availability of avocados, less production means higher price

Performance Metrics: Random Forest's ensemble approach proves to be a powerful method for reducing prediction errors and improving model robustness.

- **Future Directions:** :

Projects: Use the similar method in my future projects as it has proven to yield results and clear insights.

Model Deployment: Integrating the Random Forest model into a real-world system for continuous evaluation and feedback.

Advanced Techniques: Experimenting with other algorithms such as Gradient Boosting or Neural Networks could further improve predictive performance for complex datasets.

REFERENCES

Data Sources:

[Avocado Prices and Sales Volume 2015-2023] - Data provided here was collected from Hass Avocado Board - Category Data. This is an updated version of dataset already introduced to Kaggle.

URL: https://www.kaggle.com/datasets/vakhariapujan/avocado-prices-and-sales-volume-2015-2023

Libraries and Tools:

NumPy: Harris, C.R., et al., "Array programming with NumPy," Nature, 2020. URL: https://numpy.org/

Pandas: McKinney, W. "Data structures for statistical computing in Python," Proceedings of the 9th Python in Science Conference, 2010. URL: https://pandas.pydata.org/

Matplotlib: Hunter, J.D., "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, 2007. URL: https://matplotlib.org/

Scikit-learn: Pedregosa, F., et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2011. URL: https://scikit-learn.org/

Documentation and Tutorials:

Data Science: https://athena.explore.ai/

THE END

Thank you.