Avocado Price Prediction using Machine Learning

1. Problem Definition

The global avocado market has seen a significant rise due to increasing awareness of its health benefits. The U.S., one of the largest consumers, shows varied price patterns across regions and time. Businesses, especially in agriculture, retail, and distribution, can benefit from price forecasting to optimize their production and marketing strategies.

The objective of this project is to predict avocado prices based on a wide range of features such as sales volume, region, year, type of avocado (organic vs. conventional), and bag sizes. We will employ various machine learning models to predict the average price of avocados, selecting the best one based on performance metrics.

2. Data Analysis

The dataset comprises avocado prices and sales data from multiple regions across the U.S., with data ranging from 2015 to 2018. There are 14 features and 18,000+ observations in the dataset. Key features include:

Date: The date the data was recorded.

AveragePrice: The average price of avocados (target variable).

Total Volume: The total number of avocados sold.

4046, 4225, 4770: Product PLU codes that indicate different avocado varieties.

Total Bags: The total number of bags sold, including small, large, and extra-large.

Type: The avocado type, either organic or conventional.

Year: The year the data corresponds to.

Region: The U.S. region where the sales took place.

3. Exploratory Data Analysis (EDA)

Price Trends Over Time

Plotting the AveragePrice over time revealed clear seasonal trends. There were periodic price spikes, likely due to supply and demand fluctuations. Prices were generally higher in the winter months compared to summer, possibly due to reduced supply during the off-season.

Organic vs. Conventional

Organic avocados consistently commanded a higher price than conventional ones, often by a significant margin. This is likely due to consumer preferences for organic products and the higher production costs associated with organic farming.

Regional Variation

There were significant regional differences in avocado prices. For instance, California, one of the top avocado-producing states, showed more stable but sometimes higher prices, reflecting local supply and demand conditions. On the other hand, regions like the Midwest and Northeast had more price volatility, possibly due to transport costs or varying local demand.

Correlation Analysis

A correlation matrix revealed the relationship between numeric variables. The total sales volume had a weak negative correlation with price, suggesting that larger sales volumes might slightly reduce the price, likely due to supply and demand dynamics. However, other features like bag sizes and PLU codes showed minimal correlation with AveragePrice, indicating that no single feature dominates price prediction and a combination of features is required for better accuracy.

4. Preprocessing Pipeline

Handling Dates

The Date column was split into separate Month and Day columns to capture the seasonal effects more precisely. By doing so, the model can learn the impact of seasonality on pricing trends.

Encoding Categorical Variables

The Type and Region columns were categorical and needed to be transformed. One-hot encoding was applied to both columns. For example, Type was transformed into two binary columns (one for conventional and one for organic), and Region was transformed into multiple binary columns, one for each region.

Feature Scaling

To avoid features with larger ranges dominating model training, standardization (z-score normalization) was applied to features such as Total Volume, Total Bags, and the PLU codes (4046, 4225, 4770). This step ensures that all features have the same scale, helping models like linear regression and gradient boosting perform optimally.

Train-Test Split

The dataset was split into a training set (80%) and a testing set (20%). This split allows us to train models on a majority of the data while evaluating their generalization on unseen data. Cross-validation was used later to assess model performance across multiple folds of the data.

5. Building Machine Learning Models

Several machine learning models were trained and evaluated on this dataset. Below are the models used, along with their performance evaluations:

Linear Regression

Linear regression is a simple model that assumes a linear relationship between input features and the target variable. This model served as the baseline.

Pros: It’s easy to interpret and computationally inexpensive.

Cons: It struggles with non-linear relationships and complex interactions between features.

Performance: Linear Regression resulted in a relatively high Mean Squared Error (MSE), indicating it was not able to capture all the complexities of the data.

Random Forest Regressor

Random Forest is an ensemble learning method that creates multiple decision trees and averages their predictions. It’s effective in capturing non-linear relationships and interactions between features.

Pros: Handles non-linearity, reduces overfitting due to averaging, and works well with large datasets.

Cons: Can be computationally expensive and less interpretable compared to simpler models.

Hyperparameter Tuning: GridSearchCV was used to optimize parameters such as the number of trees (n\_estimators) and maximum depth (max\_depth). After tuning, Random Forest significantly outperformed linear regression.

Performance: Random Forest reduced the MSE and provided better predictions across different regions and time periods.

Gradient Boosting Regressor

Gradient Boosting builds trees sequentially, each one correcting the errors of the previous one. This approach is highly effective for improving model performance.

Pros: It’s very effective at reducing errors in predictions and can capture complex patterns in the data.

Cons: More prone to overfitting, but this can be mitigated with tuning. It is computationally expensive due to sequential learning.

Hyperparameter Tuning: The learning rate and the number of estimators were fine-tuned using GridSearchCV. By adjusting these parameters, the model was able to reduce errors further and generalize better on the test data.

Performance: Gradient Boosting emerged as the best-performing model, achieving the lowest Root Mean Squared Error (RMSE) and outperforming Random Forest by a small margin.

6. Model Evaluation Metrics

To evaluate the models, the following metrics were used:

Mean Absolute Error (MAE): Measures the average absolute difference between the predicted and actual prices.

Mean Squared Error (MSE): Penalizes larger errors by squaring the differences between the predicted and actual prices.

Root Mean Squared Error (RMSE): Provides the error in the same units as the target variable, making it easier to interpret.

7. Hyperparameter Tuning

For both Random Forest and Gradient Boosting, hyperparameter tuning was crucial. Using GridSearchCV, parameters such as the number of trees, tree depth, and learning rate were adjusted to achieve optimal performance. By fine-tuning these models, we were able to significantly reduce the error metrics and improve predictive accuracy.

8. Concluding Remarks

In this project, the Gradient Boosting model performed the best, demonstrating its ability to capture the complex relationships between avocado sales data and price fluctuations. The Random Forest model also performed well, and both models benefited from hyperparameter tuning.

Key takeaways include:

Feature Importance: Sales volume, region, and year were significant predictors of avocado prices. Organic vs. conventional type also played a critical role.

Preprocessing: Proper handling of categorical data and scaling of features greatly improved model performance.

Model Selection: While Linear Regression served as a baseline, Random Forest and Gradient Boosting provided much better predictive accuracy due to their ability to model complex non-linear relationships.

This predictive model can be utilized by avocado producers, retailers, and distributors to forecast future prices and optimize their supply chains accordingly.