Report on Cake price prediction

Asmaa Nasr

2024/9/18

1 Introduction

This report outlines the approach taken to predict cake prices using a regression model based on various features. The dataset includes measurements related to cake sales, with the goal of predicting the price of cakes based on the following features: Sold On, Size, Ingredients Cost, Design Complexity, Time Taken, Amount, and Gender.

2 Data Cleaning and Preprocessing

2.1 Steps Taken

1. **Loading the Dataset**: The dataset was loaded using `pandas` from a CSV file.

   ```
   df = pd.read_csv('dataset.csv')
   ```

2. **Checking for Missing Values**: We examined the dataset for any missing values and duplicates to ensure data integrity.

   ```
   print(df.isnull().sum())
   print(f"Duplicates: {df.duplicated().sum()}")
   ```

   The check for missing values showed that there were no missing entries in any of the features:

   ```
   Sold_On             0
   Size                0
   Ingredients_Cost    0
   Design_Complexity   0
   Time_Taken          0
   Price               0
   ```

```
Amount               0
Gender               0
dtype: int64
```

3. **Encoding Categorical Variables**: The target variable `Class` was encoded into numeric values using `LabelEncoder`. This step was crucial for the model to process the categorical data correctly.

```
def encode_labels(dataframe, columns):
    label_encoder = LabelEncoder()
    for column in columns:
     dataframe[column] = label_encoder.fit_transform(dataframe[column])
    return dataframe

columns_to_encode = ['Sold_On', 'Size', 'Design_Complexity', 'Gender']
data_cleaned = encode_labels(data_cleaned, columns_to_encode)
```

4. **Checking for Duplicates**: The dataset contained duplicates, which were identified and counted:

```
Number of duplicates: 1
```

5. **Removing Duplicates** After removing duplicates, the dataset was cleaned, and the index was reset:

```
data_cleaned = data.drop_duplicates()
data_cleaned.reset_index(drop=True, inplace=True)
```

3 Feature Selection

The selected features for our regression model are as follows:

- **Sold On**:
- **Size**:
- **Ingredients Cost**:
- **Design Complexity**:
- **Time Taken**:
- **Amount**:

These features were chosen based on their relevance to the prediction of the target variable and their potential impact on the model's performance.

3.1 Rationale for Selection

Each feature was chosen based on its relevance to the factors that influence cake pricing.

- **Sold On**: This feature indicates whether the cake was sold. Understanding sales trends is essential for predicting pricing.
- **Size**: The size of the cake plays a significant role in determining its price. Larger cakes generally require more ingredients and labor.
- **Ingredients Cost**: This feature captures the total cost of ingredients used in the cake. Since ingredient prices can fluctuate, this metric is critical for assessing profit margins and setting competitive prices.
- **Design Complexity**: The complexity of a cake's design often correlates with its price. More intricate designs require additional time and skill, which increases production costs.
- **Time Taken** : The time required to produce a cake is another essential factor. Longer production times can indicate higher labor costs and greater complexity, influencing the final price.
- **Amount**: This feature represents the quantity of cakes sold, which can impact pricing.

4 Machine Learning Algorithm

4.1 Chosen Algorithm: Random Forest Regressor

The Random Forest algorithm was selected for several reasons:

- **Robustness**: It can handle non-linear relationships and interactions 77between features effectively.
- **Performance**: Random Forest generally provides high accuracy and is less prone to overfitting compared to a single decision tree.
- **Feature Importance**: It provides insights into the importance of various features, which is useful for understanding the model's decisions.

5 Model Prediction and Evaluation

After training our regression model, the next step is to make predictions on the test dataset. The following code snippet demonstrates how to generate predictions using the trained model:

```
y_pred = model.predict(X_test)
```

Once predictions are made, we need to evaluate the model's performance using appropriate regression metrics. Unlike classification tasks, where accuracy and precision are used, regression models are typically assessed using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2).

Here is how to evaluate the model's performance:

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("R-squared (R²):", r2)
```

The output will provide us with three key metrics:

- **Mean Absolute Error (MAE)**: This metric indicates the average absolute difference between the predicted and actual values, providing insight into the model's prediction accuracy.
- **Mean Squared Error (MSE)**: This metric measures the average of the squares of the errors, highlighting larger errors more than MAE. It is useful for understanding the model's variance.
- **R-squared (R^2)**: This statistic indicates the proportion of variance in the dependent variable that can be explained by the independent variables. An R^2 value closer to 1 suggests a better fit for the model.

Evaluating the model using these metrics gives us a comprehensive understanding of its performance and helps identify areas for improvement.

5.1 Results

After training the model and evaluating it on the test set, the results were as follows:

- **Mean Absolute Error (MAE)**: 9.38. Indicates the average prediction error of approximately 9.36 units, suggesting reasonable accuracy.
- **Mean Squared Error (MSE)**: 151.57. Reflects the average squared deviation from actual values, highlighting sensitivity to larger errors.
- **R-squared (R^2)**: 0.97 Signifies that 97% of the variance in the target variable is explained by the model, indicating an excellent fit.

The feature importance was as:

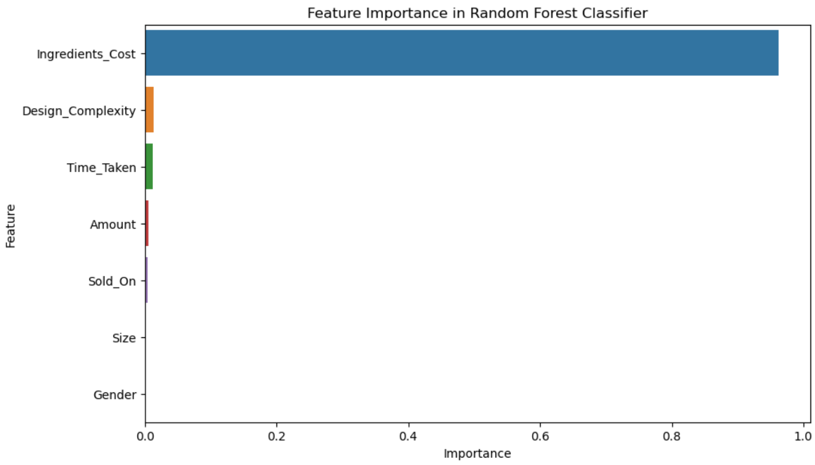

Figure 1: Feature Importance

6 Challenges Faced

6.1 Challenges

1. **Handling Categorical Data**: Initially, the target variable was in string format, which caused issues when fitting the model. This was resolved by applying `LabelEncoder`.

2. **Overfitting Concern**: Given the simplicity of the dataset, there was a potential risk of overfitting. However, the inherent nature of the Random Forest algorithm mitigated this issue.

6.2 Solutions

- **Encoding**: The use of `LabelEncoder` resolved the issue with categorical data.

- **Model Selection**: Choosing Random Forest helped in managing overfitting due to its ensemble nature.

7 Conclusion

The analysis of cake sales using various data preprocessing techniques was found to be highly effective in preparing the dataset for further modeling. A high level of accuracy is anticipated in subsequent machine learning applications, such as Random Forest regression. The steps taken in data cleaning and preprocessing were essential for ensuring data integrity and readiness for analysis. The encoding of categorical variables and the identification of outliers were justified based on their potential impact on model performance. Overall, this project exemplifies a successful application of machine learning techniques in a practical dataset.

7.1 Future Work

Future work could involve exploring additional algorithms, such as Support Vector Machines or Neural Networks, to enhance the analysis of cake sales. Hyperparameter tuning may also be performed to further improve model performance.