BigMart Sales Prediction Project

Objective

To predict the sales of products at BigMart outlets based on historical sales data and product/store information using machine learning techniques.

Steps for the Project

1. Problem Statement

Develop a predictive model to estimate the sales of products for BigMart. This will help in:

- Stock management
- Revenue estimation
- Enhancing marketing strategies

2. Data Collection

Use publicly available datasets or provided datasets with features like:

- Store Information: Store type, location, size, etc.
- Product Information: Product type, category, visibility, price, etc.
- Sales Information: Historical sales data.

Popular datasets:

- BigMart Sales dataset: https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/

3. Exploratory Data Analysis (EDA)

- Understand the dataset: Inspect the columns, data types, and missing values.
- Visualizations: Analyze relationships between sales and variables (store size, product visibility, etc.).

- Correlations: Use correlation heatmaps to identify key predictors.

4. Data Preprocessing

- Handle Missing Values: Impute missing data for critical features.

- Feature Engineering: Create new features such as yearly sales trends or interaction terms.

- Encoding Categorical Variables: Convert categorical data using one-hot or label encoding.

- Normalization/Scaling: Standardize numerical features for better model performance.

5. Model Development

- Split Data: Divide data into training and testing sets (e.g., 80:20 ratio).

- Model Selection: Experiment with models like Linear Regression, Random Forest, Gradient Boosting, Neural Networks.

- Hyperparameter Tuning: Optimize model parameters using Grid Search or Random Search.

6. Evaluation Metrics

Evaluate the models using:

- Root Mean Squared Error (RMSE)

- Mean Absolute Error (MAE)

- R-squared (R^2)

7. Deployment

Deploy the model using:

- Flask/Django for a web application.

- Streamlit for a simple, interactive dashboard.

Provide inputs for store and product details to predict sales.

Code Outline

a. Import Libraries

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

b. Load Data

```python
data = pd.read_csv('bigmart_data.csv')
```

c. Preprocess Data

```python
# Handle missing values
data['Item_Weight'].fillna(data['Item_Weight'].mean(), inplace=True)
# Encode categorical variables
data = pd.get_dummies(data, columns=['Item_Type', 'Outlet_Type'], drop_first=True)
```

d. Split Data

```python
X = data.drop(['Item_Outlet_Sales'], axis=1)
y = data['Item_Outlet_Sales']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

`` ``` ``

e. Train Model

```python
model = RandomForestRegressor(random_state=42)

model.fit(X_train, y_train)
```

f. Evaluate Model

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

rmse = np.sqrt(mean_squared_error(y_test, y_pred))

r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse}, R²: {r2}")
```

g. Deployment

Use Flask or Streamlit for the user interface where users input store and product details to get predictions.

Expected Deliverables

1. Exploratory Analysis: Visualizations and insights.

2. Predictive Model: Well-tuned and evaluated ML model.

3. Dashboard/Interface: A user-friendly tool for predictions.