WALMART STORES SALES FORECASTING

BY

UGOCHUKWU DARLINGTON IGBOKWE

Overview

The Walmart Store Sales Forecasting project aims to predict weekly sales for various store locations using historical sales data and external factors. This project uses Python and data science techniques to analyze trends, classify customers, and recommend inventory levels. The analysis incorporates promotions, holidays, and external factors like CPI and unemployment rates to provide actionable insights.

For this project, we have used the dataset available from 'Walmart Store Sales Forecasting' project that was available on Kaggle. In this dataset, we have weekly sales data for 45 stores and 99 departments for a period of 3 years. In addition, we had store and geography specific information such as store size, unemployment rate, temperature, promotional markdowns etc. Using these factors, we needed to develop a regression model that can forecast the sales and is also computationally efficient and scalable. The key issues that we faced in this analysis is the large dataset that resulted into several computational challenges because of which we had to modify our approach in addressing the problem. We also faced significant challenges in identifying the right variables on which the analysis could be conducted

Project Objectives

1. **Predict Weekly Sales**: Use historical data to forecast future sales for different stores.
2. **Classify Customers**: Group customers based on purchasing behavior using clustering techniques.
3. **Inventory Recommendations**: Estimate optimal inventory levels based on demand forecasts.
4. **Evaluate Promotions**: Analyze the impact of markdown promotions on sales performance.
5. **Incorporate External Factors**: Factor in holidays, CPI, and unemployment rates for improved prediction accuracy.
6. **Identify Trends**: Perform time series analysis to uncover seasonal and regional trends.

Data Description

The project utilizes three primary datasets:

1. **features.csv**:
    - Contains store-level features like Temperature, Fuel_Price, CPI, Unemployment, and markdown promotions.
    - Includes IsHoliday to indicate holiday weeks.
2. **stores.csv**:
    - Metadata about stores, including Store, Type, and Size.
3. **train.csv/test.csv**:
    - Historical data containing Store, Dept, Date, and Weekly_Sales (in train.csv).
    - Test data includes Store, Dept, and Date for prediction.

Tools and Libraries

- **Libraries**: pandas, numpy, matplotlib, seaborn, scikit-learn.
- **Techniques**:
    - Regression models (e.g., Random Forest Regressor, Decision Tree Regression)

Workflow

1. Data Loading and Preprocessing

- Import datasets using pandas.
- Handle missing values:
    - Fill markdown columns with 0.
    - Use forward fill for CPI and Unemployment.
- Convert Date columns to datetime format.
- Merge datasets on relevant keys (Store, Date).

2. Feature Engineering

- Add time-related features: Year, Month, Week, Day, IsWeekend.
- Encode categorical variables like Type using one-hot encoding.
- Aggregate data to analyze trends at the store and department levels.

3. Exploratory Data Analysis (EDA)

- **Sales Trends**: Visualize sales trends over time.
- **Holiday Effects**: Compare sales during holiday vs non-holiday weeks.
- **Promotions**: Assess the impact of markdowns on sales.
- **External Factors**: Analyze relationships between sales, CPI, and unemployment.
- **Regional Trends**: Compare sales by store type and size.

4. Modeling

4.1 Time Series Forecasting

- Use Prophet to predict weekly sales based on historical data.
- Evaluate forecasts visually and compute RMSE for accuracy.

4.2 Demand Forecasting

- Train a Random Forest Regressor using features like CPI, unemployment, and markdowns.
- Evaluate model performance using RMSE and feature importance.

4.3 Customer Classification

- Perform clustering using KMeans on customer-related features.
- Visualize clusters to identify purchasing behavior patterns.

Key Code Snippets

```
import numpy as np
import pandas as pd
from pandas.plotting import autocorrelation_plot as auto_corr

import plotly.express as px
import plotly graph_objects as go
from plotly.ofline import iplot
import matplotlib.pyplot as plt
import matplotlib.patches as patches
%matplotlib inline
import matplotlib as mpl
```

```
import seaborn as sns
```

```
df_train = pd.read_csv("train.csv")
df_features = pd.read_csv("features.csv")
df_stores = pd.read_csv("stores.csv")
df_test = pd.read_csv("test.csv")
```

Results and Insights

1. **Trends**:
    - Sales exhibit seasonal patterns, with peaks during holidays.
    - Promotions significantly boost sales during markdown periods.
2. **Forecasting**:
    - Time series models effectively predict weekly sales with RMSE of 500-1000 (example).
3. **Customer Clusters**:
    - Three distinct customer segments identified based on purchasing behavior.
4. **Inventory Recommendations**:
    - Suggested optimal stock levels to meet demand during peak weeks.

Conclusion and Recommendations

- Walmart can use this model to optimize inventory levels, target promotions, and anticipate peak sales periods.
- Incorporating real-time data (e.g., weather conditions) may further improve accuracy.
- Expand analysis to include department-level trends for deeper insights.