SALES FORECASTING AND OPTIMIZATION SYSTEM

Report

"اللهم علمنا ما ينفعنا و انفعنا بما علمتنا و زدنا بك علما اللهم بارك لنا في اوقاتنا يا الله "

[DATA SCIENCE]

Engineers:
Atef,Mariam,Mohamed Elawady,Mohamed Mousa,Omar Shohieb,Rana Abdelrahman

Project Goal:

To develop a robust sales forecasting system that predicts future product sales across various countries, stores, and product types. The system aims to provide accurate sales predictions to aid in inventory management, resource allocation, and overall business strategy optimization.

Project Scope:

Data Source: Historical daily sales data from January 1, 2015, to December 31, 2018, encompassing sales figures for different products (Mug, Hat, Sticker) across multiple stores (KaggleMart, KaggleRama) in several countries (Finland, Norway, Sweden). The dataset is provided in train.csv.

coding:

1. Data Preprocessing:

- Handle missing values (if any).
- Perform feature engineering, including creating temporal features (year, month, day of the week,day of the month).
- Encode categorical features (country, store, product) using Label Encoding. Saved encoders are in the models directory (e.g., le_country.pkl).
- Identify and handle outliers (e.g., sales < 950 units were filtered as seen in EDA.ipynb).

2. Exploratory Data Analysis (EDA):

- Conduct a thorough analysis of the sales data to identify trends,
- seasonality, patterns, and correlations. This includes visualizing sales over time, by country, store,
- and product. Key EDA work is documented in EDA.ipynb.

3. Model Selection and Training:

- Evaluate multiple time-series forecasting models, including XGBoost, LightGBM, and Prophet, as detailed in Model_Comparison.ipynb.
- Select the best-performing model based on metrics like RMSE, MAE, and R^2 score.
- Train the chosen model (XGBoost was selected) on the preprocessed training data. The training process is implemented in Model_Train.py.
- Save the trained model (e.g., xgboost_model.pkl) and performance metrics (best_model.txt).

4. Deployment:

- Develop a user-friendly web application using Streamlit (app.py) that allows users to: Upload a test dataset (CSV format, similar to test.csv).
- Manually input parameters (date, country, store, product) to get sales predictions.
- View the predicted sales figures.
- See the performance metrics of the deployed model.

Deliverables:

- Jupyter notebooks for EDA and model comparison.
- Python scripts for data preprocessing, model training, and the Streamlit application. Saved model artifacts (trained model, label encoders).
- A requirements.txt file listing all necessary Python packages.
- A README.md file providing an overview of the project and instructions for setup and execution.
- A final report summarizing the project, methodology, results, and conclusions.

Key Technologies and Libraries:

- Python
- Pandas, NumPy for data manipulation.
- Scikit-learn for preprocessing and metrics.
- XGBoost, LightGBM, Prophet for modeling.
- Matplotlib, Seaborn for data visualization.
- Streamlit for web application development.
- Joblib for saving/loading model objects .

Success Metrics:

The primary success of the project will be measured by the accuracy of the sales forecasts (low RMSE/MAE, high R^2) and the usability of the deployed Streamlit application. The model performance metrics (RMSE: ~30.71, MAE: ~19.36, R^2: ~0.99 for XGBoost as per Model_Train.ipynb) indicate a strong predictive capability.