Zindi Store Sales Forecasting Challenge – 8th Place Solution by Mawero Rodney G.

1.1 Overview

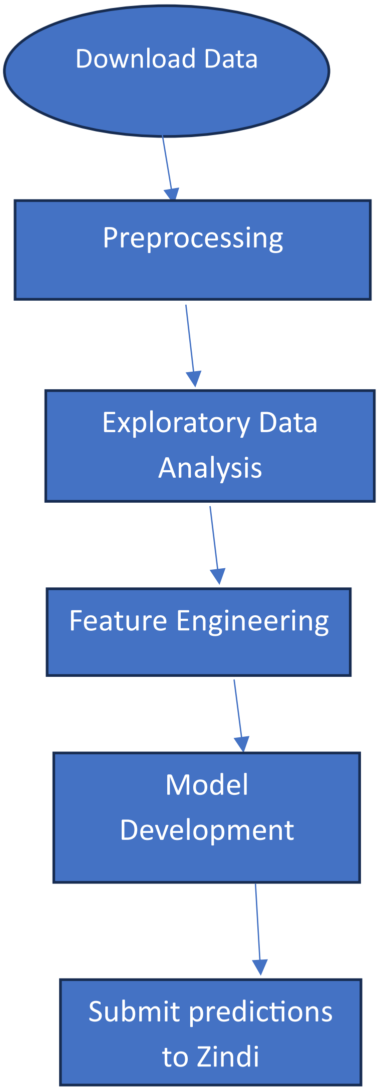
Sales forecasting is the foundation of a business's financial story. Once you have your sales forecast you can create profit and loss statements, cash flow statements and balance sheets, thus helping you set goals for your company. Proper forecasting also ensures you have the right stock at all times and leads to less wasted stock.

1.2 Objectives

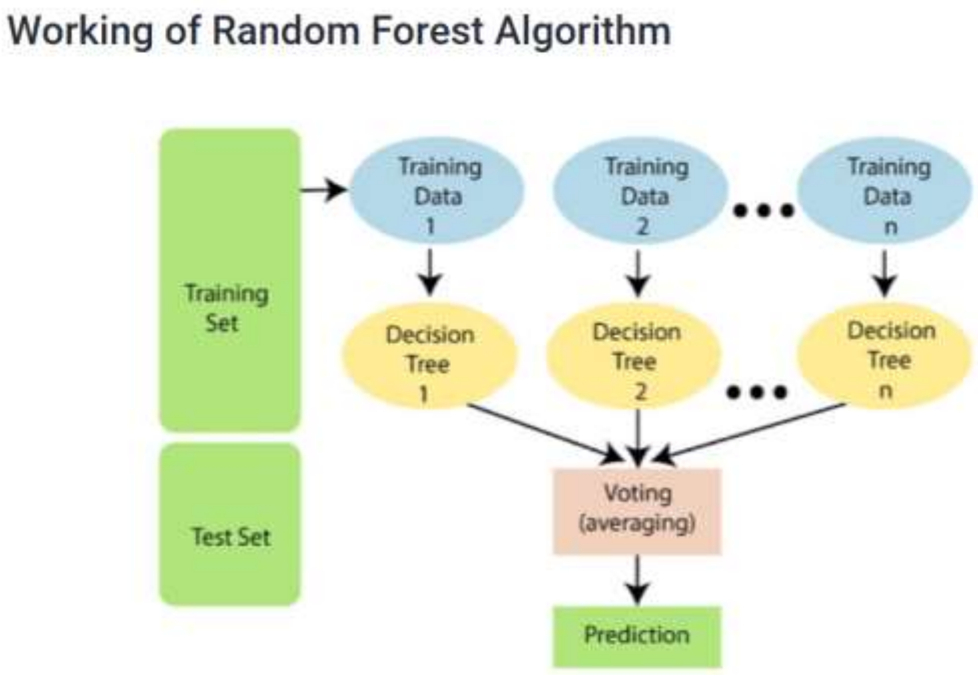
The objective of this challenge is to create a model to forecast the number of products purchased per week per store over the **next eight weeks**, for grocery stores located in different areas in the same country. The solution to this challenge can be used by small chain stores to know how much stock to order per week and per month.

The final model submitted is a **random forest regressor.**

2.1 Machine Learning Pipeline



2.1 Architecture Diagram



Source: Simplilearn https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm

The following steps explain the working Random Forest Algorithm:

Step 1: random samples from a given data or training set are selected

Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision trees.

Step 4: Finally, selection of the most voted prediction result as the final prediction result. This combination of multiple models is called Ensemble. The ensemble method used in random forest is **bagging**

Bagging: Creating **a different training subset** from sample training data **with replacement is called Bagging.** The final output is based **on majority voting**.

3.0 ETL – Extraction Transformation and Loading of Data

Data

Description	**Files**
Contains the target. This is the dataset used to train the model	- **train.csv**
Resembles Train.csv but without the target-related columns. This is the dataset used to apply the trained model.	- **test.csv**
Information about holidays	- **holidays.csv**
Information about the different stores such as their locations	- **stores.csv**
Information about the time periods with their associated date features e.g. day of the week, day of the year,	- **dates.csv**
Shows the submission format for the challenge	**- SampleSubmission.csv**

- Data was downloaded to local directory.
- train.csv and test.csv were merged with the 3 supplementary datasets – holidays, stores and dates
- columns after merging
  'date', 'store_id', 'category_id', 'target', 'onpromotion',
  'nbr_of_transactions', 'year', 'month', 'dayofmonth', 'dayofweek',
  'dayofyear', 'weekofyear', 'quarter', 'is_month_start',
  'is_month_end',
  'is_quarter_start', 'is_quarter_end', 'is_year_start', 'is_year_end',
  'year_weekofyear', 'is_holiday', 'holiday_type', 'city', 'stores_type',
  'cluster'
- test data has all the above columns with 2 exceptions after merging: **target** and **nbr_transactions**
- On checking for null values, **holiday_type** had null values in both train and test merged datasets. They were filled with -0.1 since 0 was a value in holiday_type

3.1 EDA

- Facet grid plots revealed **store_52** was only opened in year 4.
- **store_22** and **store_21** opened from the second year hence lack data from the first year
- Most stores were open throughout from year 1 to year 4
- **End of year** had a noticeable peak in sales, followed by a trough at the beginning of the following year. This coincides with Christmas + New year festivities. Sales pick up thereafter
- Venn diagram plots revealed that test data is a subset of train data. The test data has fewer holiday types. Train has all the holiday types in test data plus those absent in test data.

3.2 Feature Engineering

- Since the objective is **to forecast weekly sales data** for the **next 8 weeks** in test data, the merged train and test data were aggregated by the target sum and onpromotion-(total items sold on promo) in a week + year+year_weekofyear + store_id + category_id. The original datasets had daily sales data.
- holiday_count column was created to count all unique holiday types > =0 since NANs in holiday type were filled with -0.1. Taking NANs as 0, and holiday counts > 0 did not yield significant performance gains.
- is_holiday was redefined as a binary column where holiday counts per week were > 0 =1 else 0 . is_holiday in the original data was binary for daily data
- a time-based split was done on merged train data using year_weekofyear. The last 8 weeks were used for validation
- log transformation was done in 2 different ways:
  - on **y_train** and **y_val,** then leaving predictions **as is**
  - on **X_test** after training on **raw** y_train and y_val

4.0 Model Development

- The best 2 models were Random Forest Models
  - Model 1 params
    - random_state = 42
    - max_depth =12
  - Model 2 params
    - random_state = 42 only
    - hence max_depth = default (None)

Features Selected :

- 'year', "year_weekofyear", 'weekofyear', 'month', 'onpromotion', 'city', "holiday_count",'cluster',

Features Dropped:

- nbr_of_transactions = ABSENT FROM TEST DATA
- Time features less than week were discarded
- Is_holiday is already captured in holiday count hence dropped
- holiday_type was dropped as it increased rmse , although the test data contained a subset of training data holiday types
- quarter increased rmse, dropping it improved performance
- Sparse binary time features such as is_month_start, is_year_start were dropped as they worsened model performance
- Lag features were in earlier iterations of model development. RMSE was stuck between 2-4 RMSE

Models Used in Experiments

-Random forest

- Catboost – LightGBM

Random Forest outperformed the other 2.

5. Inference

- Training + inference for each of the models was done in the same jupyter notebook

6. Run Time

- **Less than 10 minutes**

7. Performance metrics

- **Model 1 - 1.330325751 – Public Leaderboard**

  1.425069361 – Private Leaderboard

- **Model 2 - 1.570639167 – Public Leaderboard**
  1.573959068 – Private Leaderboard

8. Final Remarks

- **Onpromotion was the feature with the highest feature importance.**
- **Customers love a good deal. So MORE PROMOS = MORE SALES**

A vote of thanks to:

Young Ai Leaders - Kampala Hub

Tufuna Technologies Cyber Camp

And
THE ZINDI TEAM – not forgetting Erick Jagwara – AMB.