Final Project

K.N.Toosi University of Technology
Introduction to Data Mining

Fall 2024

Part I
Sales Prediction

Dataset

The dataset is available for download on the course website.

Dataset Description:
Here's a description of the columns based on their names:

- Store_id: A unique identifier for each store in the dataset.
- RetailType: The category or type of retail store (e.g., grocery, clothing, electronics).
- Stock variety: Refers to the range or variety of products offered by the store (e.g., basic, extended, premium).
- DistanceToRivalStore: The distance from this store to its nearest rival store.
- RivalOpeningMonth: The month when a rival store opened in proximity to the store.
- RivalEntryYear: The year when a rival store entered the market or opened near the store.
- ContinuousBogo: Likely indicates whether a "Buy One Get One" (BOGO) offer is currently active (possibly a binary or numeric flag).
- ContinuousBogoSinceWeek: The number of weeks since the continuous BOGO offer was initiated.
- ContinuousBogoSinceYear: The year when the continuous BOGO offer started.
- ContinuousBogoMonths: The total duration (in months) for which the continuous BOGO offer has been active
- DayOfWeek: The day of the week (e.g., Monday, Tuesday, etc.) when the sales data was recorded.
- Date: The specific date on which the sales data was collected.
- Sales: The total sales made by the store on the given day.
- NumberOfCustomers: The number of customers who visited the store on the given day.
- Is_Open: A binary indicator (e.g., 1 for open, 0 for closed) that specifies whether the store was open on that day.

- BOGO: A flag indicating whether a "Buy One Get One" (BOGO) offer was active on the given day (e.g., 1 for active, 0 for not active).
- Holiday: A binary indicator (e.g., 1 for holiday, 0 for non-holiday) to indicate whether the day was a recognized holiday.

Task

1. **Load Dataset:**
   - Read the training data from CSV file, selecting relevant columns. Consider that some columns may not be useful for your analysis and can be omitted.
2. **Load and Merge Store Data:**
   - Read the stores data from another CSV file to get additional information about each store.
   - Combine the training and store data based on the **store_id** column to create a comprehensive dataset.
3. **Train & test Data:**
   - Please divide 70% of the training examples into the training set and use the remaining 30% as the test set. Select the first 70% of the examples in chronological order, as we aim to evaluate our models on their ability to extrapolate to dates beyond the training range.
4. **Preprocess Data:**
   - Replace the missing values in the 'DistanceToRivalStore' column with the median of the existing values, and set the remaining missing values to zero. Feel free to modify this for better approaches if you prefer.
   - Extract 'Year', 'Month', 'Day', and 'WeekOfYear' from the 'Date' column, then remove the 'Date' column. Utilize the `pd.to_datetime` function and its attributes for easy handling.
   - Remove the 'Customers' column since it is not available during testing.
   - Standardize the features using `StandardScaler` to ensure all features have a similar scale.
5. **Prepare Data for Modeling:**
   - Separate the features (`X`) and the target variable (`y`), which is **total_sales**.
6. **Train and Evaluate Model:**

- Train two models: Linear Regression and Random Forest Regressor and evaluate their performances on the test data.

7. **Feature Selection & Importance:**

   - Utilize `feature importance` from the Random Forest model. Which features were identified as the most important by this model?

Part II
Sentiment Analysis

Dataset

The dataset is available for download on the course website.

Task

1. **Load and preprocess data:**
    - Load the dataset and preprocess the reviews (e.g., lowercasing, removing stop words and punctuation)
2. **Vectorize reviews:**
    - TF-IDF:
        - `from sklearn.feature_extraction.text import TfidfVectorizer`
    - Word2Vec:
        - `from gensim.models import Word2Vec`
    - BERT:
        - `from sentence_transformers import SentenceTransformer`
3. **Hyperparameter tuning for classification Models:**[1]
    - Conduct a grid search on each model to identify the optimal hyperparameters, utilizing the F1 score for evaluation. ensure the F1 score exceeds 0.8.
        - Logistic Regression
        - Random Forests
        - K-Nearest Neighbors

Note

Any attempt to use AI tools for generating the code is strictly prohibited. Students will be asked to present and explain their code during a class session.

[1] Due to the potential data imbalance caused by splitting reviews at a threshold of 3, we compare three different algorithms to ensure robustness.