

1. Introduction. An overview of the project and an outline of the shared work.

Elo, one of the largest payment brands in Brazil, has built partnerships with merchants in order to offer promotions or discounts to cardholders. However, the questions are: Do these promotions work for either the consumer or the merchant? Do customers enjoy their experience? Do merchants see repeat business? Personalization is key.

In this project, we aggregate merchant.csv with the new_merchant_transactions.csv and historical_transactions.csv tables and then aggregate the concatenated table to the main train table. New features are built by successive grouping on card_id, in order to recover some information. We then developed six algorithms, including linear regression, decision tree, random forest, support vector machine (SVM), K-nearest neighbors (KNN), naive Bayes, k-means clustering, agglomerative nesting (AGNES), and density-based spatial clustering of applications with noise (DBSCAN) to predict the target: customer loyalty, in order to identify and serve the most relevant opportunities to individuals. Our goal is to improve customers' lives and help Elo reduce unwanted campaigns, to create the right experience for customers.

I did pre-processing and feature engineering and produced train_fea_eng.csv for group to train different models. I trained linear regression and decision tree, other models such as random forest, KNN, SVM, etc were trained.

2. Description of your individual work. Provide some background information on the development of the algorithm and include necessary equations and figures.

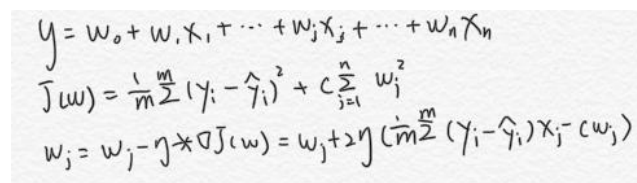
2.1 Cleaning algorithm:

Fillna: Fill NA/NaN values using the specified method.

```
DataFrame.fillna(self, value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, **kwargs)[source]
```

Drop outliers: In most of the cases a threshold of 3 or -3 is used i.e if the Z-score value is greater than or less than 3 or -3 respectively, that data point will be identified as outliers.

2.2 Linear regression: one of the simplest and most commonly used statistical modeling techniques. Makes strong assumptions about the relationship between the predictor variables (x) and the response (y). Only valid for continuous outcome variables (not applicable to binary class)


$$y = w_0 + w_1x_1 + \dots + w_jx_j + \dots + w_nx_n$$
$$J(w) = \frac{1}{m} \sum (\hat{y}_i - y_i)^2 + c \sum_{j=1}^n w_j^2$$
$$w_j = w_j - \eta \frac{\partial J(w)}{\partial w_j} = w_j + 2\eta \left(\frac{1}{m} \sum (\hat{y}_i - y_i) x_{ij} - cw_j \right)$$

Assumption: $y = \beta_0 + \beta_1x + \text{err}$

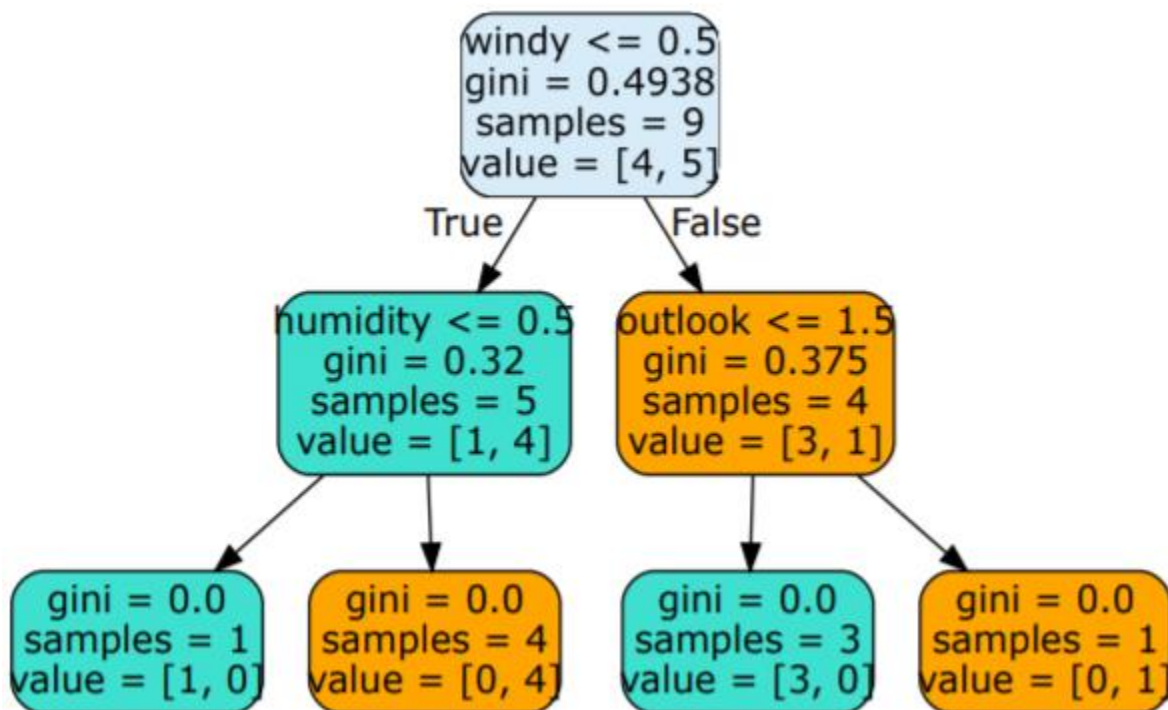
Goal: estimate β_0 and β_1 based on the available data

Final Model $\hat{y} = \beta^0 + \beta^1 x + \text{err}$

β^0 and β^1 are model parameters

Objective: minimize the error, the difference between our observations and the predictions made by our linear model

2.3 Decision tree: a hierarchical technique that means a series of decisions are made base on some metrics. Decision trees are Nonparametric, there are no assumptions on concerning hyperparameters or distributions. Decision trees are based on graph-based models. They are the type of Acyclic Graphs. Decision tree graphs are based on nodes and edges. These nodes and edges defined by decision rules applied to the input features.

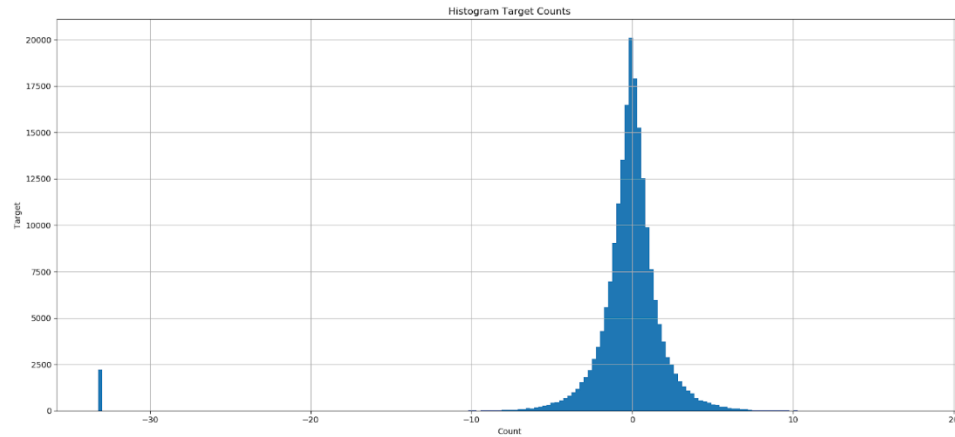


3. Describe the portion of the work that you did on the project in detail. It can be figures, codes, explanation, pre-processing, training, etc.

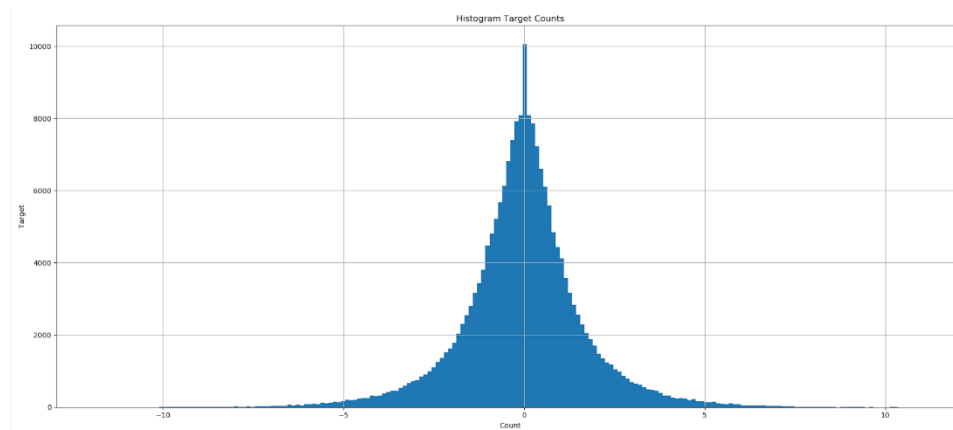
3.1 Preprocessing and feature engineering

- 1) Use function of 'missing_values_table' to check missing values of all the tables;
- 2) Check unique values of 'category_2', 'category_3' and 'merchant_id' in historical_transactions and new_merchant_transactions and fill missing values with values different from unique values existed in those columns.
- 3) Calculated z-score of target and drop outliers whose Z-score value is greater than or less than 3 or -3 respectively.

Histogram of target with outliers



Histogram of target without outliers



- 4) Get dummies of 'feature_1', 'feature_2' and 'feature_3' in historical_transactions and new_merchant_transactions, train and test.
- 5) Define functions that aggregate the info contained in tables. The first function aggregates the function by grouping on card_id; the second function first aggregates on the two variables card_id and month_lag. Then a second grouping is performed to aggregate over time.
- 6) Fill missing values after feature engineering with 0
- 7) Merge all the dataframes and then write the merged df to a .csv file

3.2 Models

3.2.1 Linear Regression

- 1) Split the data into features X and target y
- 2) Using train_test_split from sklearn.model_selection, divide the data into training and testing (with test_size=0.3 and random_state = 0)

- 3) Use a self-defined linear regression to fit a linear model.
- 4) Use GridSearchCV from sklearn.model_selection to tune hyperparameters such as learning rate and constant penalty on coefficients, with scoring='neg_mean_squared_error', n_jobs=-1, iid=False, cv=KFold(n_splits=10, random_state=0), return_train_score=True.

3.2.2 Decision Tree

- 1) As the target in the dataset is continuous, we converted the target to categories using:
`bins = np.arange(-12.5, 12.5, 1)`
`names = np.arange(-12, 12, 1)`
`data['new_target'] = pd.cut(data['target'], bins, labels=names)`
- 2) Using LabelEncoder() to transfer the target.
- 3) Using StandardScaler() to standardize X_train and X_test
- 4) Using train_test_split from sklearn.model_selection, divide the data into training and testing (with test_size=0.3 and random_state = 0)
- 5) Use GridSearchCV from sklearn.model_selection to tune hyperparameters such as max_depth, min_samples_leaf and min_samples_split
- 6) Train a decision tree with criterion as gini and best hyperparameter selected using X_train and y_train.
- 7) Make predictions y_pred using the trained model and X_test
- 8) Calculate accuracy and mean_square_error using y_pred and y_test.
- 9) Get confusion matrix
- 10) Display decision tree.

4. Results. Describe the results of your experiments, using figures and tables wherever possible. Include all results (including all figures and tables) in the main body of the report, not in appendices. Provide an explanation of each figure and table that you include. Your discussions in this section will be the most important part of the report.

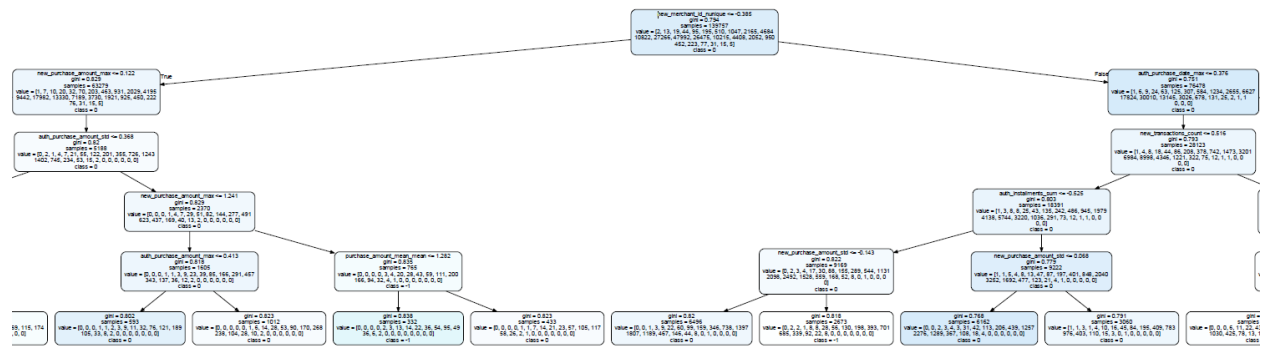
4.1 Linear Regression

Best_score: -0.8982704791215965; best_params: {'estimator__eta': 0.1};
Mean_squared_error: 2.65; r2_score: 0.09

4.2 Decision Tree

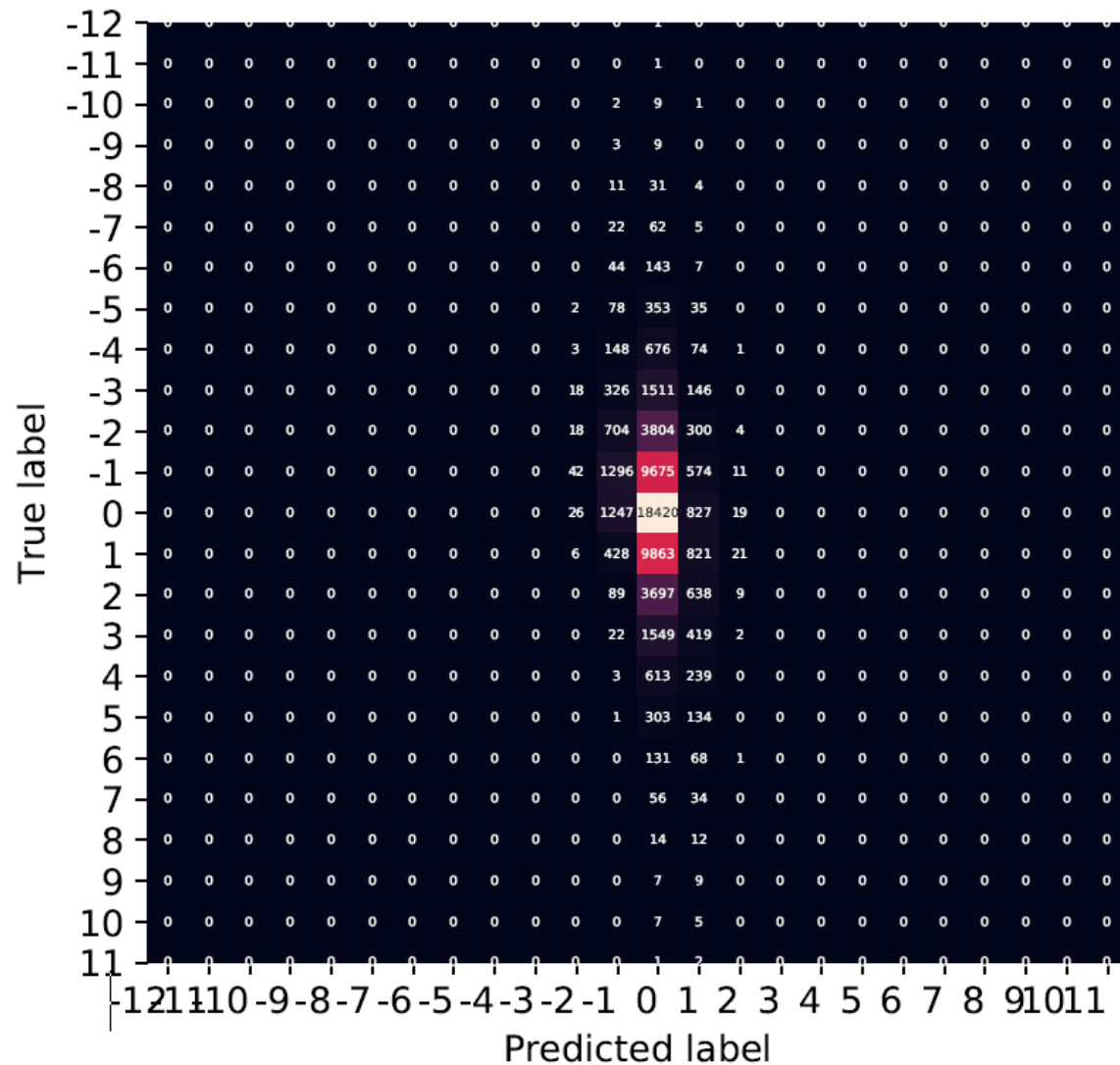
Best hyperparameters are 'max_depth': 6, 'min_samples_leaf': 1, 'min_samples_split': 2

Part of the decision tree:



Accuracy: 34.33 and Mean_square_error: 2.87

Confusion matrix:



5. Summary and conclusions. Summarize the results you obtained, explain what you have learned, and suggest improvements that could be made in the future.

For continuous target, linear regression is the one model to be used for prediction. However, decision tree fitted using converted categorical target performs well, as the MSE is 2.87.

As for improvements could be done for this project, we could try LGBost model which uses a novel technique of Gradient-based One-Side Sampling (GOSS) to filter out the data instances for finding a split value.

6. Calculate the percentage of the code that you found or copied from the internet.

Feature Engineer: $((38-11)+(49-42)+(80-53)+(146-127)+(268-164))/273 = 67\%$

Linear Regression (183): 0%

DT_GS_Xinyu (137): 0%

GUI (907): 0%

Total: $183/(273 + 183 + 137 + 907) = 12.2\%$

7. References.

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html>

<https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba>

Elo Merchant Category Recommendation Help understand customer loyalty. Elo. (March, 2019). Retrieved from <https://www.kaggle.com/c/elo-merchant-category-recommendation>