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

As one of the largest payment brands in Brazil, Elo offers promotions or discounts to its cardholders in partnership with merchants. To evaluate the effectiveness of those promotions, Elo needs to know if the promotions are able attract consumers and merchants and to what extent. Do customers enjoy their experience? Do merchants see repeat businesses? Personalization is the key.

Shared work

1. Feature engineering
2. How to clean and merge data
3. Conclusion and summary

In this project, merchant.csv was aggregated with the new_merchant_transactions.csv and historical_transactions.csv tables. The concatenated table was then aggregated into the main train table. New features were built by successive grouping on card_id in order to recover information. Random forest was utilized to select top 25 import features in the name of feature_select.csv. The csv was then utilized to show KNN and naïve Bayes model. Six algorithms including linear regression, decision tree, random forest, support vector machine (SVM), K-nearest neighbor (KNN), and naive Bayes, were developed to predict customer loyalty and to identify the most relevant opportunities for individuals. Our goals are to help Elo reduce unwanted campaigns, to create the right experience for customers, and to improve customers' lives.

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

I developed three algorithms including random forest, K-nearest, and naïve Bayes in this Project. Specifically, we did feature engineering, random forest, and model accuracy and classification report in sequential order. Firstly, we cleaned the outlier and missing value, merged individual csv files, and created features to a new csv file named train_fea_eng.csv for feature engineering. Secondly, from random forest, I developed top 25 important features and saved them as a new csv named feature_select.csv. I then utilized the csv to run K-nearest (KNN) and naïve Bayes. Thirdly, I developed model accuracy and classification report.

Random forest(RF):

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean

prediction (regression) of the individual tree. Random decision forests correct for decision trees' habit of overfitting to their training set.

Naive Bayes:

Naive Bayes is a simple classification technique that relies on conditional probability, and predicts the most probable class given a set of inputs. It is often used as a baseline for more complex models. Naive Bayes Classifiers are extremely fast and surprisingly accurate given their "naive assumptions". Naive Bayes is a simple technique for predicting the most probable class/label given a set of features/inputs.

KNN:

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriority. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other.

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.

I worked on model. I developed random forest, KNN and naive Bayes, plotted the confusion matrix, and changed classification ranging to improve the accuracy.

To transform the variable from continuous variable into categorical variable, I divided integrals ranging from -12 to 12 into

- 24 groups with an interval of 1
- 6 groups with an interval of 4;
- 3 groups with an interval of 8.

Random forest

1. Import dataset
2. Split dataset

a. 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)

b. 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, -8)', '[-8, -4)', '[-4, 0)', '[0, 4)' , '[4, 8)', '[8, 12)']

c. 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, -4)', '[-4, 4)', '[4, 12)']

data['new_target'] = pd.cut(data['target'], bins, labels=names)

3. Perform training with random forest with all columns
4. Plot feature importance
5. select features to perform training with random forest with k columns
6. perform training with random forest with k columns
7. make predictions
8. calculate metrics gini model
9. confusion matrix for gini model
10. calculate metrics entropy model
11. Confusion matrix for entropy model

Naive Bayes

1. Import packages
2. Split the dataset

a. Split dataset

i. 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)

ii. 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, -8]', '[-8, -4]', '[-4, 0]', '[0, 4]' , '[4, 8]', '[8, 12]']

ii. 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, -4]', '[-4, 4]', '[4, 12]']

3. Perform training
4. Make predictions
5. Calculate metrics
6. Confusion matrix

KNN

1. Import packages and dataset
2. Data preprocessing
3. Split the dataset into train and test
    a. Split dataset
        i. 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)

iii. 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, -8]', '[-8, -4]', '[-4, 0]', '[0, 4]' , '[4, 8]', '[8, 12]']

iv. 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, -4]', '[-4, 4]', '[4, 12]']

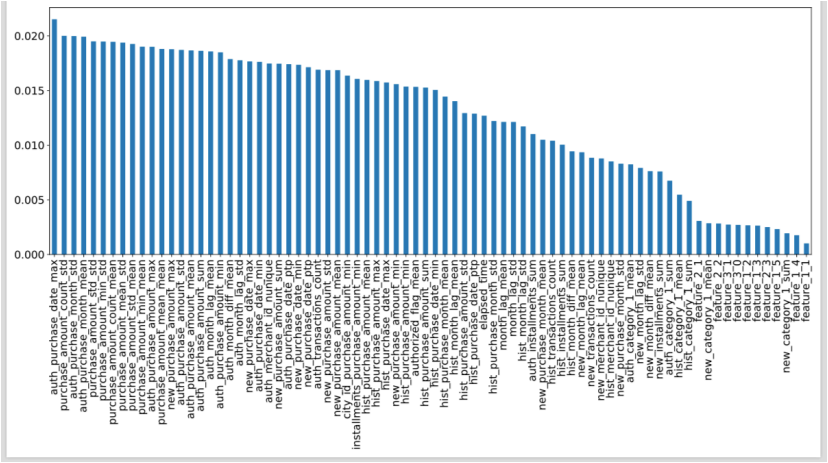
4. Standardize the data
5. Perform training
6. Make predictions
7. Calculate metrics

Plot confusion matrix

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.

- **Random Forest**



- We can see which feature is more important in this dataset, then we pick top 25 features to run the model.
    - Ranging by interval 1
        - RF all features classification report

```
Classification Report:
/Users/lancy/anaconda3/lib/python3.7/site-packages/sklearn/metrics/clas
  no predicted samples.
  'precision', 'predicted', average, warn_for)
              precision    recall  f1-score   support

         -12       0.00      0.00      0.00         1
         -11       0.00      0.00      0.00         2
         -10       0.00      0.00      0.00        14
          -9       0.00      0.00      0.00        15
          -8       0.00      0.00      0.00        41
          -7       0.00      0.00      0.00        84
          -6       0.00      0.00      0.00       242
          -5       0.00      0.00      0.00       463
          -4       0.00      0.00      0.00       927
          -3       0.11      0.00      0.00      2014
          -2       0.17      0.02      0.03      4632
          -1       0.27      0.16      0.20     11709
           0       0.38      0.81      0.51     20471
           1       0.23      0.15      0.18     11362
           2       0.15      0.03      0.05      4420
           3       0.13      0.01      0.02      1906
           4       0.05      0.00      0.00       850
           5       0.50      0.00      0.01       377
           6       0.00      0.00      0.00       197
           7       0.00      0.00      0.00       116
           8       0.00      0.00      0.00        26
           9       0.00      0.00      0.00        15
          10       0.00      0.00      0.00        11
          11       0.00      0.00      0.00         2

    accuracy                           0.34     59897
   macro avg       0.08      0.05      0.04     59897
weighted avg       0.26      0.34      0.26     59897


Accuracy :  33.93158255004424


Mean_squared_error: 2.8938678064009884
```

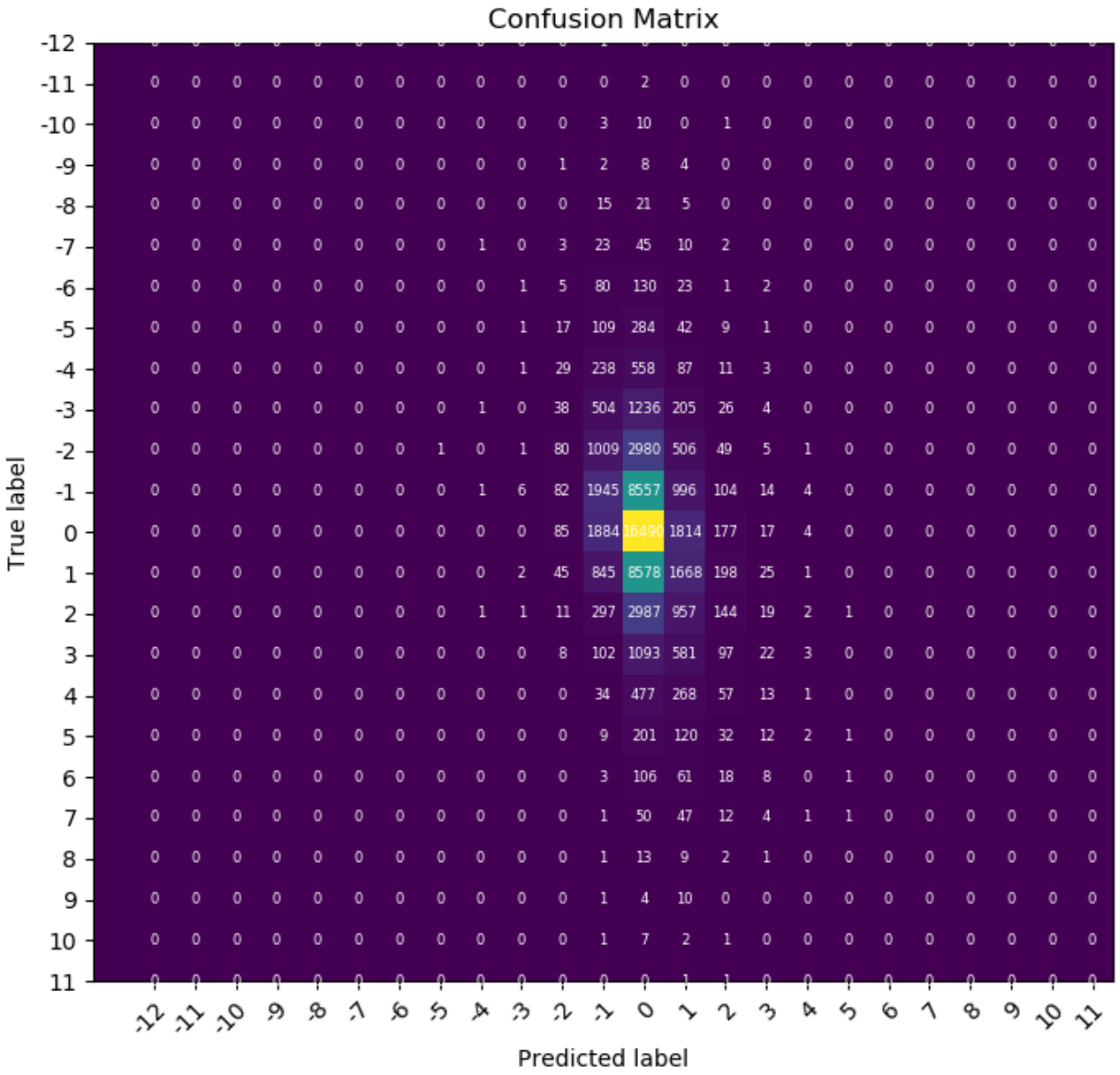
This graph show the accuracy is 33.93%, mean squared error is 2.89. Even though the accuracy is very low, but our MSE is quite ok. Because we have so much features. All F-score is closer to 0 except 0 category. The 0 category is the most data support part.

- RF k features classification report

```
Results Using K features:

Classification Report:
/Users/lancy/anaconda3/lib/python3.7/site-packages/sklearn/metrics/c
  no predicted samples.
  'precision', 'predicted', average, warn_for)
              precision    recall  f1-score   support

         -12       0.00      0.00      0.00         1
         -11       0.00      0.00      0.00         2
         -10       0.00      0.00      0.00        14
          -9       0.00      0.00      0.00        15
          -8       0.00      0.00      0.00        41
          -7       0.00      0.00      0.00        84
          -6       0.00      0.00      0.00       242
          -5       0.00      0.00      0.00       463
          -4       0.20      0.00      0.00       927
          -3       0.12      0.00      0.00      2014
          -2       0.19      0.02      0.04      4632
          -1       0.25      0.16      0.19     11709
           0       0.37      0.81      0.51     20471
           1       0.21      0.12      0.15     11362
           2       0.14      0.02      0.03      4420
           3       0.08      0.00      0.01      1906
           4       0.07      0.00      0.00       850
           5       0.00      0.00      0.00       377
           6       0.00      0.00      0.00       197
           7       0.00      0.00      0.00       116
           8       0.00      0.00      0.00        26
           9       0.00      0.00      0.00        15
          10       0.00      0.00      0.00        11
          11       0.00      0.00      0.00         2

    accuracy                           0.33     59897
   macro avg       0.07      0.05      0.04     59897
weighted avg       0.25      0.33      0.25     59897


Accuracy :  33.288812461392055
Mean_squared_error: 2.8938678064009884
```

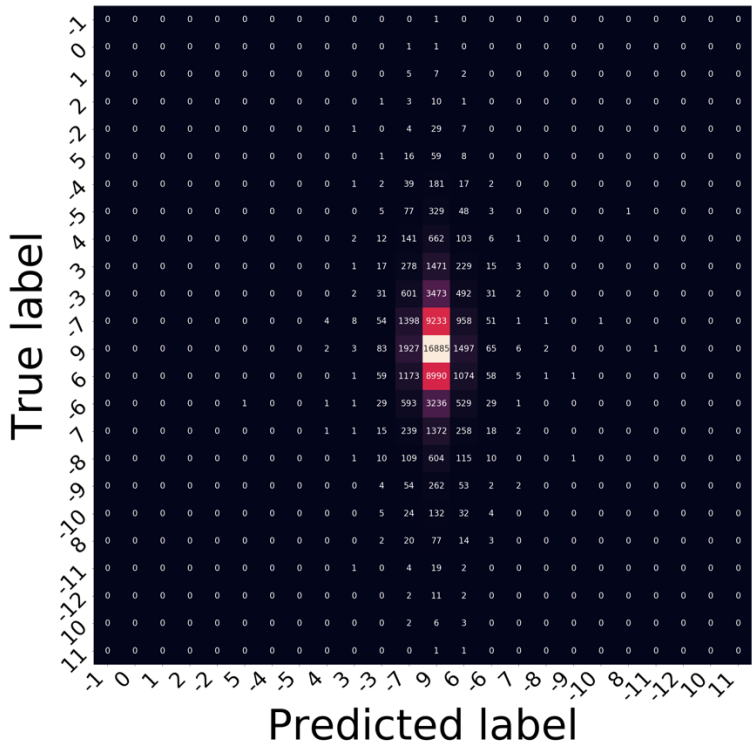
This graph show the accuracy is 33.29%, mean squared error is 2.89. Even though the accuracy is very low, but our MSE is quite ok. Because we have so much features. All F-score is closer to 0 except 0 category. The 0 category is the most data support part.

- RF all features confusion matrix



We can see on the confusion matrix, 0 is the highest number of correct prediction.

RF k features confusion matrix



We can see on the confusion matrix, 9 is the highest number of correct prediction. Which it corresponding to 0 category in our Y(new_target).

- Ranging by interval 4
  - RF all features classification report

```
Results Using All Features:

Classification Report:
/Users/lancy/anaconda3/lib/python3.7/site-packages/sklearn/metrics/cla
  and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
              precision    recall  f1-score   support

   [-12, -8)       0.00      0.00      0.00        32
    [-4, 0)        0.56      0.26      0.35     19282
    [-8, -4)       0.00      0.00      0.00       830
     [0, 4)        0.68      0.91      0.78     38159
     [4, 8)        0.40      0.00      0.00      1540
    [8, 12)        0.00      0.00      0.00        54

    accuracy                           0.66     59897
   macro avg       0.27      0.19      0.19     59897
weighted avg       0.62      0.66      0.61     59897


Accuracy :  66.07342604804916
```

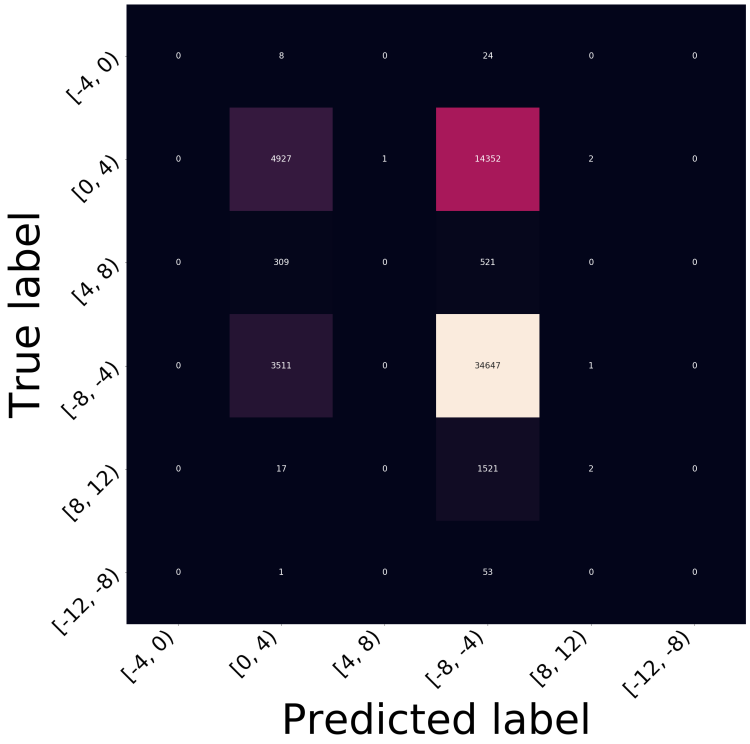
This graph show the accuracy is 66.07%. It is better than before. All F-score is closer to 0 except 0 category. The 0 category is the most data support part.

- RF k features classification report

```
Results Using K features:

Classification Report:
              precision    recall  f1-score   support

   [-12, -8)       0.00      0.00      0.00        32
    [-4, 0)        0.54      0.25      0.34     19282
    [-8, -4)       0.33      0.00      0.00       830
     [0, 4)        0.67      0.90      0.77     38159
     [4, 8)        0.23      0.00      0.00      1540
    [8, 12)        0.00      0.00      0.00        54

    accuracy                           0.66     59897
   macro avg       0.30      0.19      0.19     59897
weighted avg       0.62      0.66      0.60     59897


Accuracy :  65.53249745396263
```

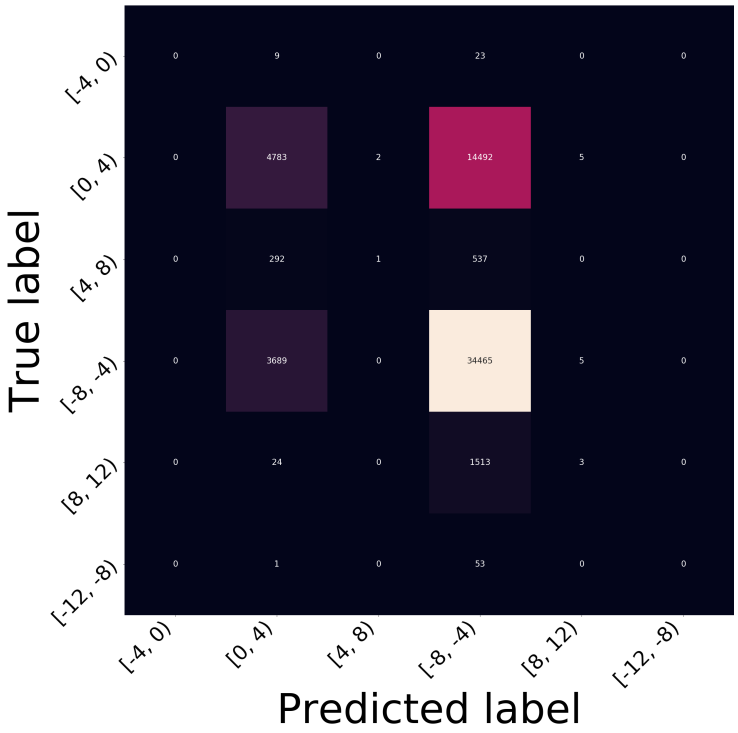
This graph show the accuracy is 65.52%. It is better than before. All F-score is closer to 0 except 0 category. The 0 category is the most data support part.

- RF all features confusion matrix



We can see on the confusion matrix, [-8,-4] is the highest number of correct prediction. Which it corresponding to [0,4) category in our Y(new_target).

- RF k features confusion matrix



We can see on the confusion matrix, [-8,-4) is the highest number of correct prediction. Which it corresponding to [0,4) category in our Y(new_target).

- Ranging by interval 8
  - RF all features classification report

```
Results Using All Features:

Classification Report:
              precision    recall  f1-score   support

   [-12, -4)       0.50      0.00      0.00       862
     [-4, 4)       0.96      1.00      0.98     57441
     [4, 12)       0.50      0.00      0.00      1594

    accuracy                           0.96     59897
   macro avg       0.65      0.33      0.33     59897
weighted avg       0.94      0.96      0.94     59897


Accuracy :  95.89962769420839
```

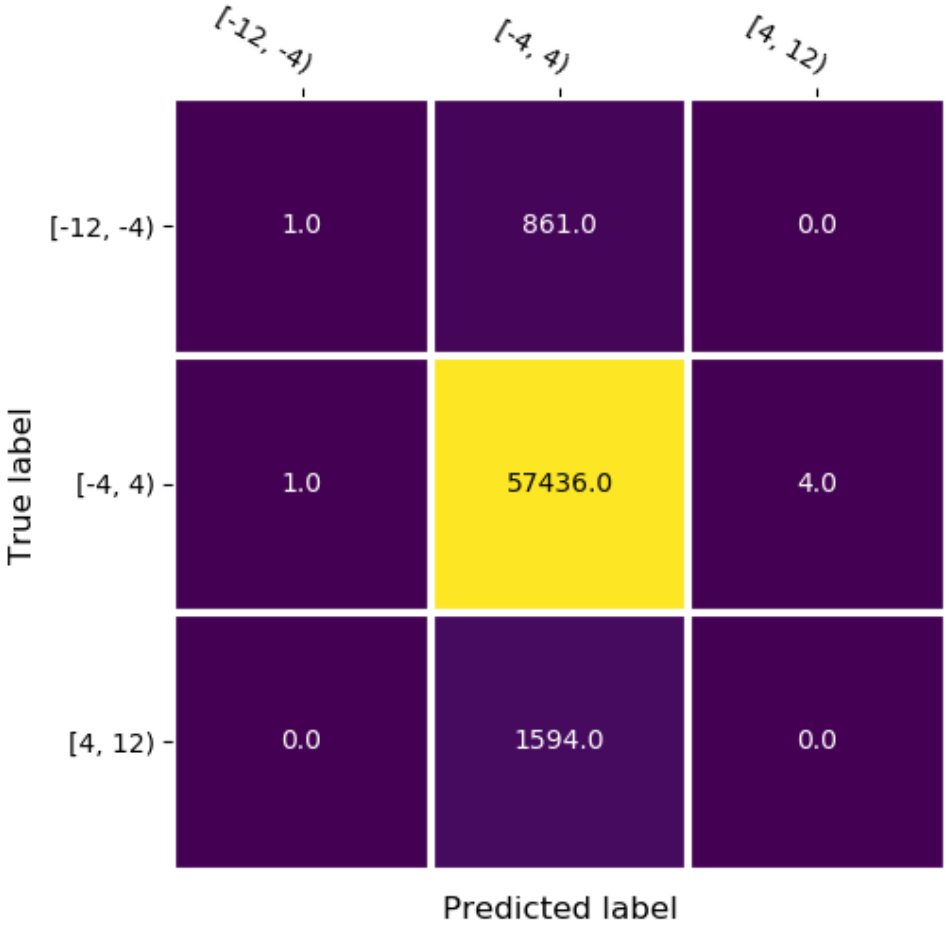
This graph show the accuracy is 95.89%. It is very good model. Also, we have 3 categories, it have 95.89% accuracy, it is very good. -4 categoy have highest F score. The -4 category is the most data support part.

  - RF k features classification report

```
Results Using K features:

Classification Report:
              precision    recall  f1-score   support

   [-12, -4)       0.00      0.00      0.00       862
     [-4, 4)       0.96      1.00      0.98     57441
     [4, 12)       0.33      0.00      0.00      1594

    accuracy                           0.96     59897
   macro avg       0.43      0.33      0.33     59897
weighted avg       0.93      0.96      0.94     59897


Accuracy :  95.89795816151059
```

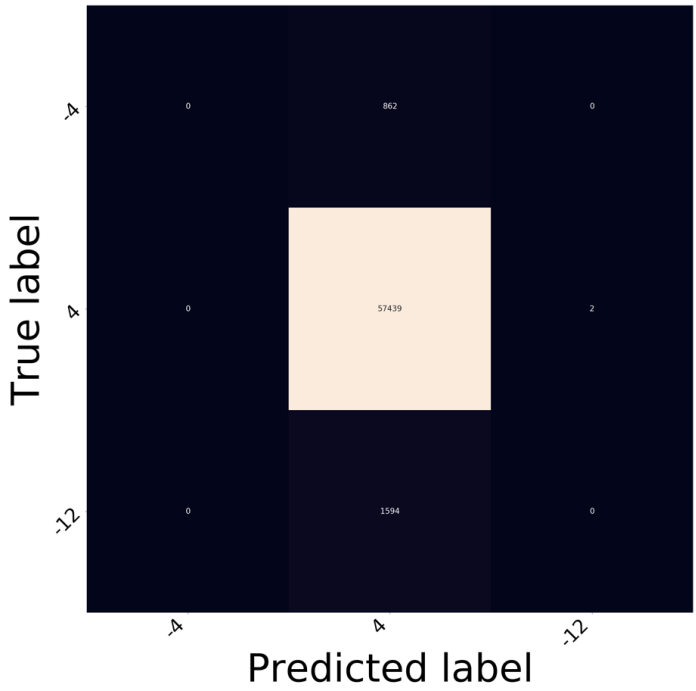
This graph show the accuracy is 95.89. It is very good model. -4 categoy have highest F score. The -4 category is the most data support part.

- RF all features confusion matrix



We can see on the confusion matrix, [-4,4] is the highest number of correct prediction.

- RF k features confusion matrix



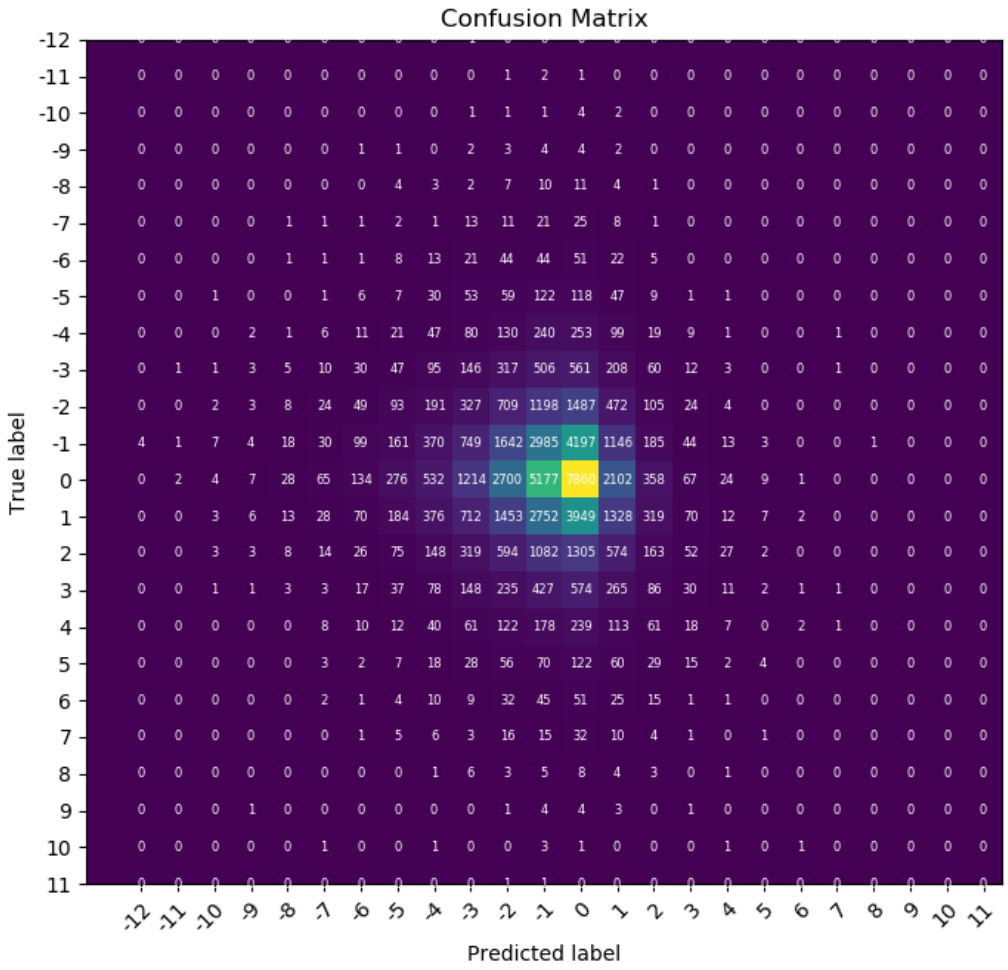
We can see on the confusion matrix, 4 is the highest number of correct prediction. Which it corresponding to -4 category in our Y(new_target).

- **KNN**
    - Ranging by interval 1
        - KNN classification report

```
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.00      0.00      0.00         4
           2       0.00      0.00      0.00         9
           3       0.00      0.00      0.00        17
           4       0.00      0.00      0.00        42
           5       0.01      0.01      0.01        85
           6       0.00      0.00      0.00       211
           7       0.01      0.02      0.01       455
           8       0.02      0.05      0.03       920
           9       0.04      0.07      0.05      2006
          10       0.09      0.15      0.11      4696
          11       0.20      0.26      0.22     11659
          12       0.38      0.38      0.38     20560
          13       0.20      0.12      0.15     11284
          14       0.11      0.04      0.06      4395
          15       0.09      0.02      0.03      1920
          16       0.06      0.01      0.01       872
          17       0.14      0.01      0.02       416
          18       0.00      0.00      0.00       196
          19       0.00      0.00      0.00        94
          20       0.00      0.00      0.00        31
          21       0.00      0.00      0.00        14
          22       0.00      0.00      0.00         8
          23       0.00      0.00      0.00         2

    accuracy                           0.22     59897
   macro avg       0.06      0.05      0.05     59897
weighted avg       0.23      0.22      0.22     59897


Accuracy :  22.184750488338313


5.843948778736832
```

This graph show the accuracy is 22.18%, mean squared error is 5.84. It is not a good model. MSE is higher. Because of lots of features. All F-score is closer to 0 except 12^{th} category. It is very worst. The 12th category is the most data support part.

- KNN classification confusion matrix



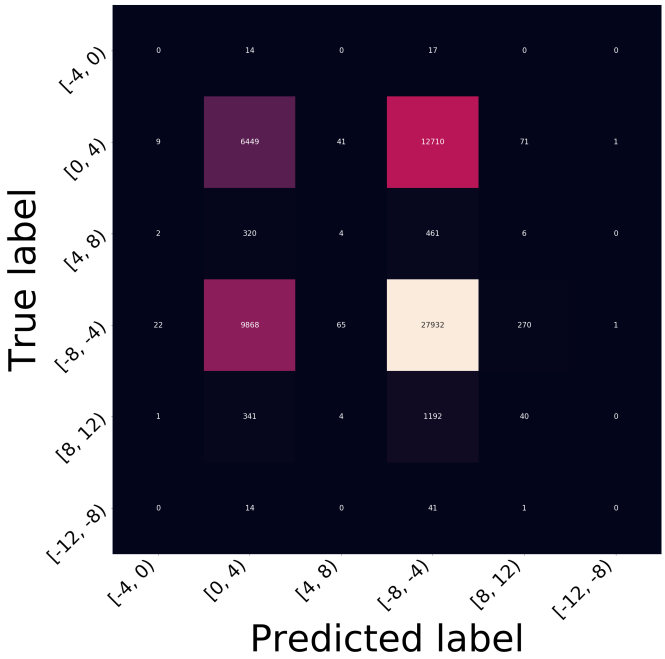
We can see on the confusion matrix; 0 part is the highest number of correct prediction.

- Ranging by interval 4
  - KNN classification report

```
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        31
           1       0.03      0.04      0.04       793
           2       0.38      0.32      0.35     19281
           3       0.66      0.73      0.69     38158
           4       0.11      0.03      0.04      1578
           5       0.00      0.00      0.00        56

    accuracy                           0.57     59897
   macro avg       0.20      0.19      0.19     59897
weighted avg       0.55      0.57      0.56     59897



Accuracy :  56.95610798537489


Mean_squared_error: 0.5176219176252567
```

This graph show the accuracy is 56.96. It is better than before. All F-score is closet to 0 except 0 category. The 0 category is the most data support part.

- KNN classification confusion matrix



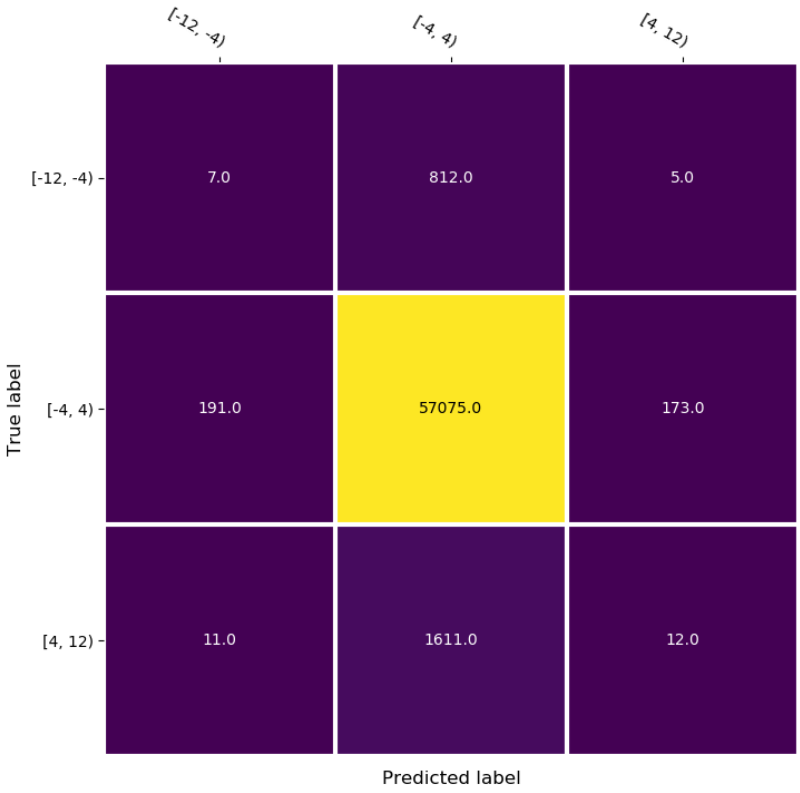
We can see on the confusion matrix, [-8,-4] is the highest number of correct prediction. Which it corresponding to [-4,0] in our Y(new_target).

- Ranging by interval 8
  - KNN classification report

```
Classification Report:
              precision    recall  f1-score   support

           0       0.03      0.01      0.01       824
           1       0.96      0.99      0.98     57439
           2       0.06      0.01      0.01      1634

    accuracy                           0.95     59897
   macro avg       0.35      0.34      0.33     59897
weighted avg       0.92      0.95      0.94     59897


Accuracy :  95.32029984807252


Mean_squared_error: 0.04759837721421774
```

This graph show the accuracy is 95.32%. It is very good model. 1 categoy have highest F score. The 1 category is the most data support part.

  - KNN classification confusion matrix



We can see on the confusion matrix, [-4,4] have the highest number of correct prediction.

- **Naive Bayes**
  - Ranging by interval 1
    - Naive Bayes classification report

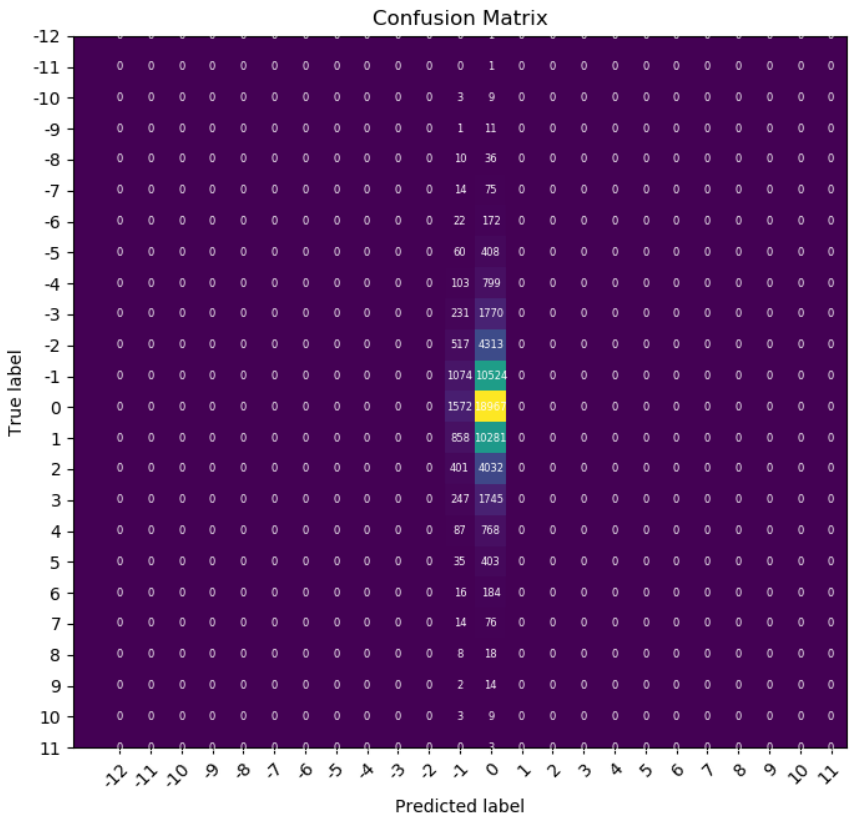
Classification Report:

	precision	recall	f1-score	support
-12	0.00	0.00	0.00	1
-11	0.00	0.00	0.00	1
-10	0.00	0.00	0.00	12
-9	0.00	0.00	0.00	12
-8	0.00	0.00	0.00	46
-7	0.00	0.00	0.00	89
-6	0.00	0.00	0.00	194
-5	0.00	0.00	0.00	468
-4	0.00	0.00	0.00	902
-3	0.00	0.00	0.00	2001
-2	0.00	0.00	0.00	4830
-1	0.20	0.09	0.13	11598
0	0.35	0.92	0.50	20539
1	0.00	0.00	0.00	11139
2	0.00	0.00	0.00	4433
3	0.00	0.00	0.00	1992
4	0.00	0.00	0.00	855
5	0.00	0.00	0.00	438
6	0.00	0.00	0.00	200
7	0.00	0.00	0.00	90
8	0.00	0.00	0.00	26
9	0.00	0.00	0.00	16
10	0.00	0.00	0.00	12
11	0.00	0.00	0.00	3
accuracy			0.33	59897
macro avg	0.02	0.04	0.03	59897
weighted avg	0.16	0.33	0.20	59897

Mean_squared_error: 3.061572365894786
Accuracy : 33.44908760038066

This graph show the accuracy is 33.45%, mean squared error is 3.06. It is not a good model. MSE it is a little bit higher. All F-score is closer to 0 except 0 category. It is very worst. The 0 category is the most data support part.

    - Naïve Bayes confusion matrix



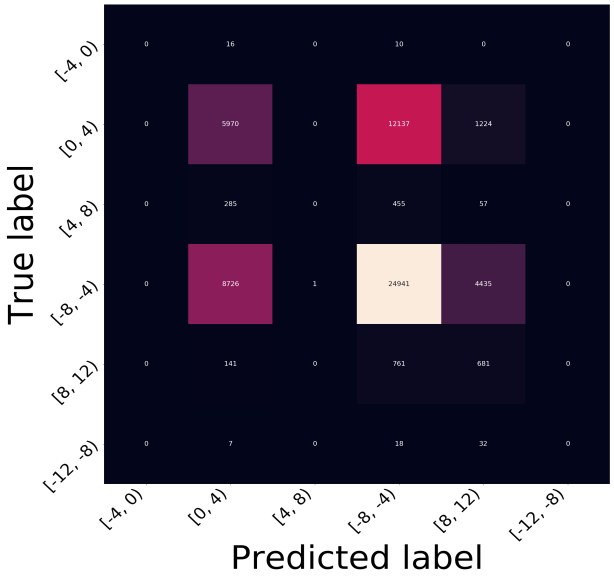
We can see on the confusion matrix, 0 have the highest number of correct prediction.

- Ranging by interval 4
  - Naive Bayes classification report

```
Classification Report:
              precision    recall  f1-score   support

   [-12, -8)       0.00      0.00      0.00        26
    [-4, 0)        0.39      0.31      0.35     19331
   [-8, -4)        0.00      0.00      0.00       797
    [0, 4)         0.65      0.65      0.65     38103
    [4, 8)         0.11      0.43      0.17      1583
   [8, 12)         0.00      0.00      0.00        57

    accuracy                           0.53     59897
   macro avg       0.19      0.23      0.19     59897
weighted avg       0.54      0.53      0.53     59897


Accuracy :  52.74387698883083
```

This graph show the accuracy is 52.74%. It is better than before.

  - Naïve Bayes confusion matrix



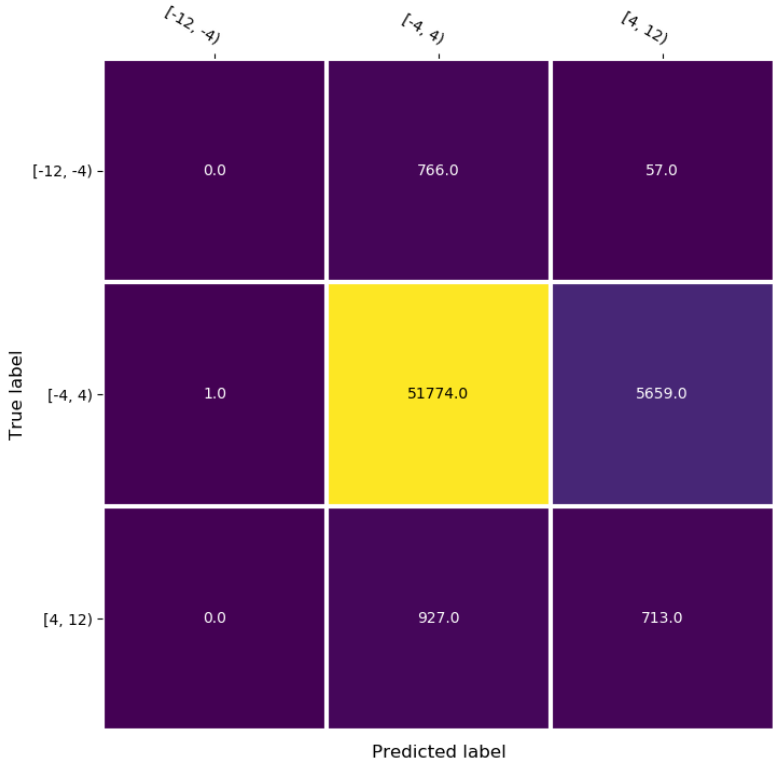
We can see on the confusion matrix, [-8, -4) is the highest number of correct prediction. Which it corresponding to [0,4) category in our Y(new_target).

- Ranging by interval 8
  - Naive Bayes classification report

```
Classification Report:
              precision    recall  f1-score   support

    [-12, -4)       0.00      0.00      0.00       823
     [-4, 4)        0.97      0.90      0.93     57434
     [4, 12)        0.11      0.43      0.18      1640

    accuracy                           0.88     59897
   macro avg       0.36      0.45      0.37     59897
weighted avg       0.93      0.88      0.90     59897


Accuracy :  87.62876270931767
```

This graph show the accuracy is 87.63%. It is good model.

  - Naïve Bayes confusion matrix



We can see on the confusion matrix, [-4,4) have the highest number of correct prediction.

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

Results

The accuracy for interval of 1 is very low for all three models (Random Forest, KNN, Naïve Bayes). But MSE it is ok except KNN. KNN is not a good model to deal with a lots of features dataset. I believe that the reason for this problem is that the raw data (the target column y_train) is continuous variable, so we changed this column to category. For ranging by interval 1, we have 24 features. It is a lot of features; it will decrease the accuracy. The accuracy for interval of 4 and 8 was improved, suggesting that less category will influence the accuracy. Our model is quite ok. In ranging interval of 8, we still have 3 calories, each model has high accuracy. Which means improve our model very successful.

Acquisitions

learned how to run the model, to interpret number meanings, to use python, and to code. Know how to programming. I have no background about python before.

Improvements

I think the confusion matrix has some problem (bug) about x and y axis and I used a class example for reference. It has the same problem. The number of correction in the confusion matrix is correct. I hope I can code by myself more than now. Know more how to interpret the code and data.

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

KNN: 0%

Random Forest: 13%

Naïve Bayes: 11%

7. References.

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