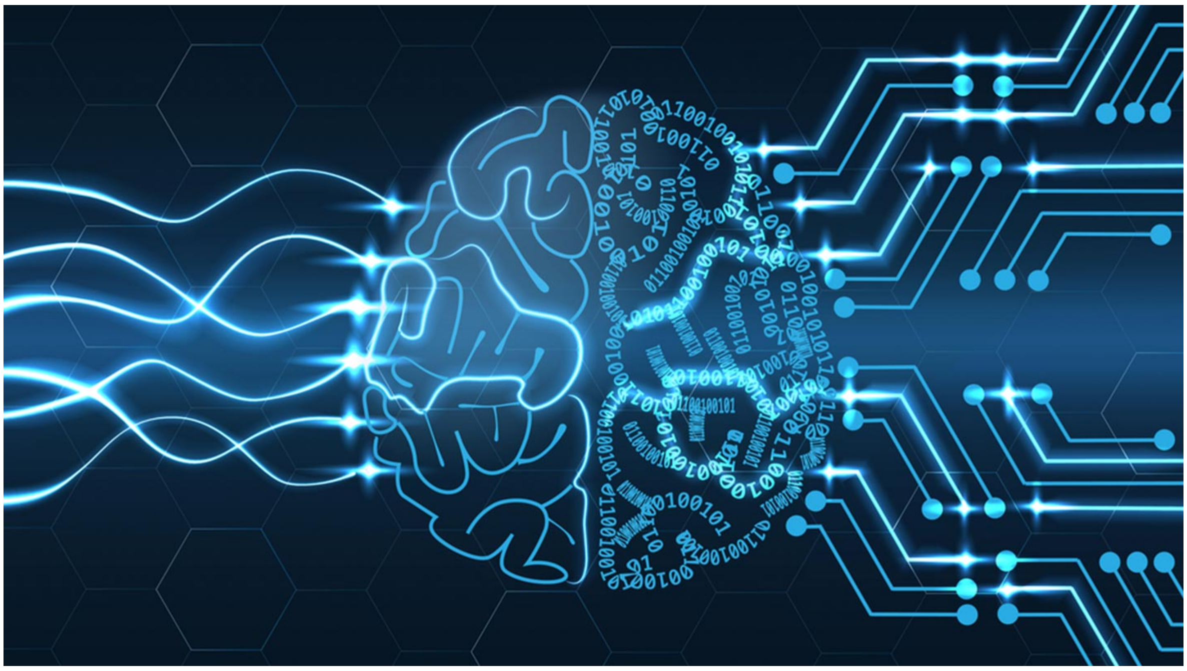
COMP3354 Group Project: Predicting repayment interval from kiva loan data

Chu Chi Hang 11 Oct 2019 Report Python, Machine Learning, Prediction, Loan



COMP3354 Group Project: Predicting repayment interval from kiva loan data

Background

Kiva is a new player in the crowd funding industry whose targeting developing countries. It is interested in predicting the repayment interval of loans such that the borrower can have a lower risk. This predictions not only applied for Kiva but also general loaning industry in developing countries. The origin of the dataset is from Kaggle, a famed data mining organization. Link: https://www.kaggle.com/kiva/data-science-for-good-kiva-crowdfunding

Goal

From eight features from `kiva_loans.csv`, including, `loan_amount`, `activity`, `sector`, `country_code`, `region`, `term_in_months`, `lender_count` and `borrower_genders` to predict `repayment_interval`.

Data Preprocessing

There are two types of data preprocessing in this project.

- One-hot encoding
- Standardization

Note that the `borower_genders` has to break down to `count_female` and `count_male`.

File

`kiva_loans_dummied.csv` refers to the dataset after one-hot encoding

`kiva_loans_standardized.csv` refers to the dataset after one-hot encoding and standardization

Validation

k-fold validation is used with k=10 after considering the processing time and stability of result.

Python Code

```python
# X = dataset without label, y = label
kf = KFold(n_splits=10) # choose k = 10
for train_index, test_index in kf.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    # do the evaluation on y_test and y_pred
```

Model Selection

There are totally six models tested for this project:

- Perceptron (linear prediction)
- Logistic Regression
- Decision Tree
- Random Forest
- K Nearest Neighbors Curse of high dimensions
- Support Vector Machine Curse of high dimensions

Note that K Nearest Neighbors and Support Vector Machine only work after feature extraction.

Advanced Prediction

There are two different advanced training method that are put into experiment:

- Feature Extraction

By changing the method and parameters of feature extraction, it is expected that the prediction can be more accurate. We use two feature extraction methods:

1. Linear Discriminant Analysis

Through extracting the eigen values from the dataset, it reconstructs a new dataset with c-1 columns for c distinct labels.

Python Code

```
# X = dataset without label, y = label
X_lda = LDA().fit_transform(X)
```

2. Best Subset

Best subset tests the performance of prediction by choosing only k columns from dataset where k is the desire number of columns.

Python Code

```
# X = dataset without label, y = label
X_subset = BestKSubset(f_classif, k=desire_no_of_column).fit_transform(X)
```

- Filtering

By breaking prediction into different stage or region over dataset, it is expected that a group of specialized prediction each working in an specific area outperform a single, general prediction model.

- Tuning parameters

By changing and testing the parameters, the prediction model might work better. One important method is tracking the mean and variance of training and validation.

Findings

Model Selection

Model	Performance
Perceptron	Does not work well. Fail to identify any `weekly`
Logistic Regression	Slightly better than Perceptron. Fail to identify any `weekly`
Decision Tree	Work best, highest accurate rate on labelling `weekly`
Random Forest	Rank second, less accurate rate on labelling `weekly` than Decision Tree
K Nearest Neighbors	Perform slightly worse than Random Forest. Fail to identify any `weekly`
Support Vector Machine	Perform in between Random Forest and Decision Tree. Fail to identify any `weekly`

Advanced Training

Before advanced training, it is decided that the Decision Tree works the best in model selection. Therefore, only decision tree is put into experiment for this part.

Filtering

```
              precision    recall  f1-score   support

      bullet       0.84      0.85      0.84     70728
   irregular       0.89      0.91      0.90    257158
     monthly       0.92      0.90      0.91    342717
      weekly       0.71      0.70      0.70       602

   micro avg       0.90      0.90      0.90    671205
   macro avg       0.84      0.84      0.84    671205
weighted avg       0.90      0.90      0.90    671205

[[ 59976   2585   8156     11]
 [  2475 234928  19617    138]
 [  8811  24925 308953     28]
 [     5    147     26    424]]
```

Performance of using the two-step approach

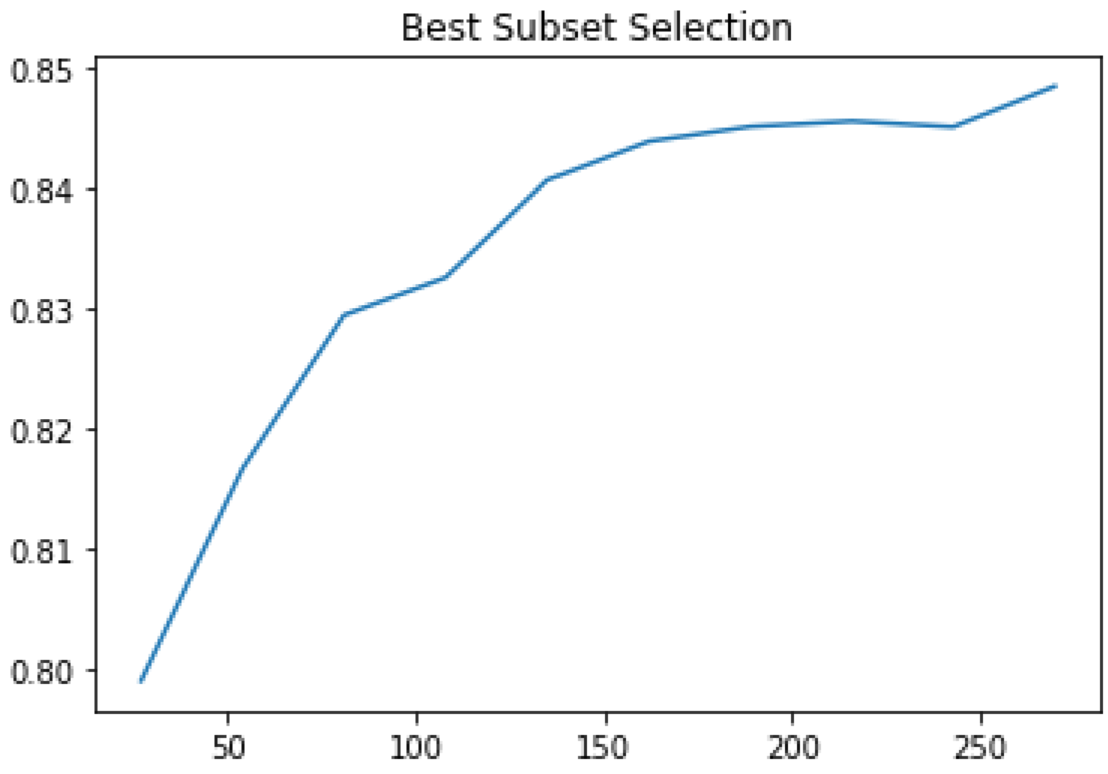
```
              precision    recall  f1-score   support

      bullet       0.84      0.85      0.85     70728
   irregular       0.89      0.91      0.90    257158
     monthly       0.92      0.90      0.91    342717
      weekly       0.73      0.70      0.72       602

   micro avg       0.90      0.90      0.90    671205
   macro avg       0.85      0.84      0.84    671205
weighted avg       0.90      0.90      0.90    671205

[[ 59982   2620   8113     13]
 [  2450 234882  19712    114]
 [  8802  24944 308942     29]
 [     3    147     29    423]]
```

Performance of a default decision tree with unlimited depth

An attempt is made when trying to first identify whether the loan application is `weekly` or not, then continue to predict it is `bullet`, `monthly` or `irregular`. All two steps are done by decision trees, as they have the best performance. However, this approaches even has lower accuracy than that of a single decision tree using all features.

Feature Extraction

By testing across different feature extraction methods, only the one-hot encoded and standardized dataset works well.



For the Best Subset, it is found that the accuracy is almost straightly increasing with the number of features. Thus, it is preferred not to have drop any features.

```
              precision    recall  f1-score   support

      bullet       0.78      0.79      0.79     70728
   irregular       0.84      0.85      0.84    257158
     monthly       0.86      0.85      0.86    342717
      weekly       0.58      0.51      0.54       602

   micro avg       0.84      0.84      0.84    671205
   macro avg       0.77      0.75      0.76    671205
weighted avg       0.84      0.84      0.84    671205

[[ 56009   3326  11368     25]
 [  3484 219483  34032    159]
 [ 12194  39407 291081     35]
 [    18    223     55    306]]
```

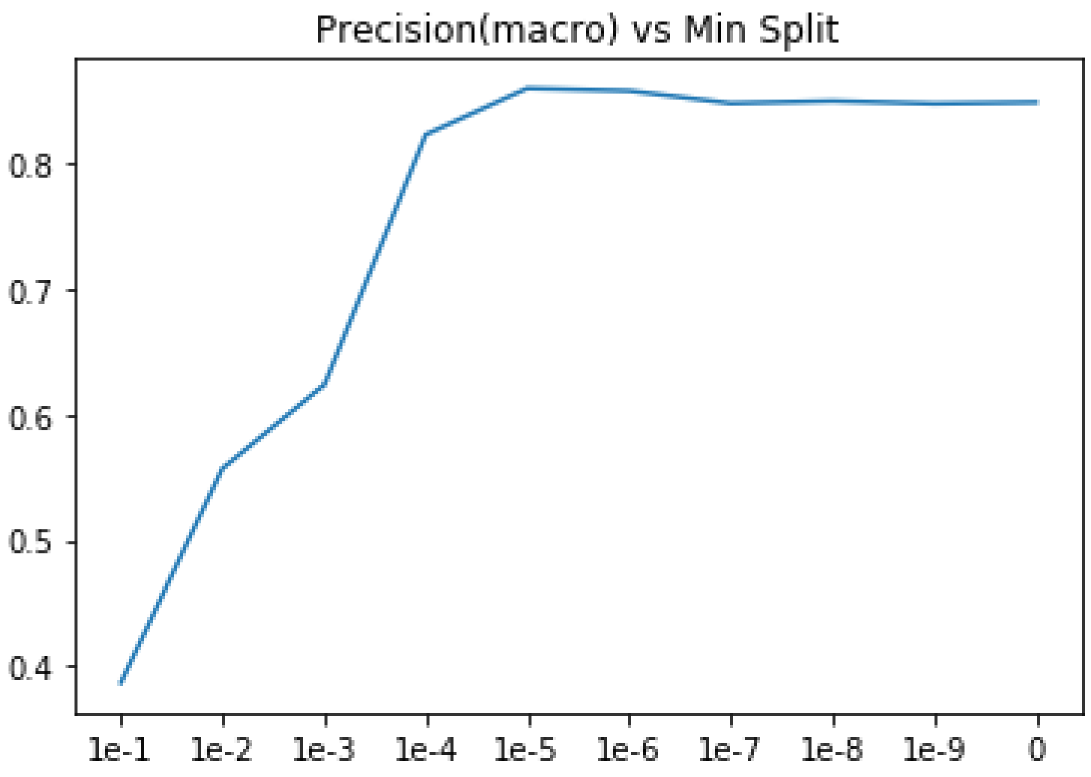
Performance of a default decision tree with LDA transformed dataset

```
              precision    recall  f1-score   support

      bullet       0.84      0.85      0.85     70728
   irregular       0.89      0.91      0.90    257158
     monthly       0.92      0.90      0.91    342717
      weekly       0.73      0.70      0.72       602

   micro avg       0.90      0.90      0.90    671205
   macro avg       0.85      0.84      0.84    671205
weighted avg       0.90      0.90      0.90    671205

[[ 59982   2620   8113     13]
 [  2450 234882  19712    114]
 [  8802  24944 308942     29]
 [     3    147     29    423]]
```

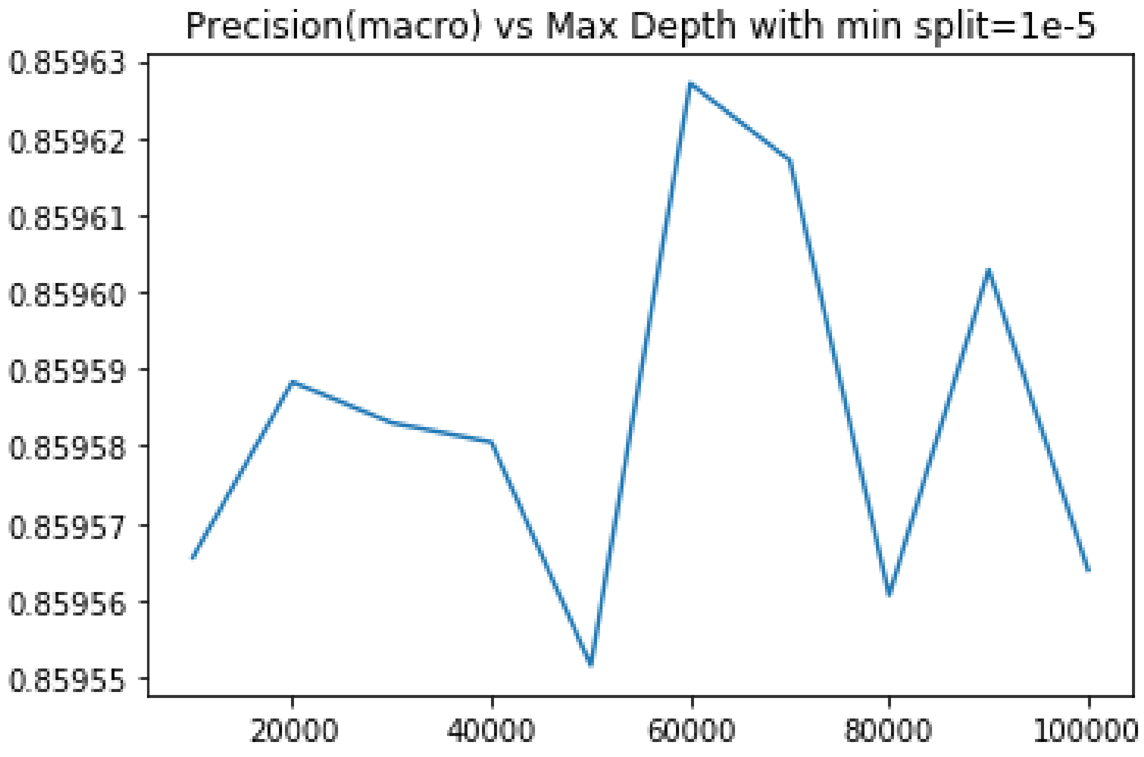
Performance of a default decision tree with one-hot encoded and standardized dataset

For Linear Discriminant Analysis, the performance of decision tree decreases for around 6-8%. It is suggested not to use Linear Discriminant Analysis as feature extraction.

Tuning Parameters



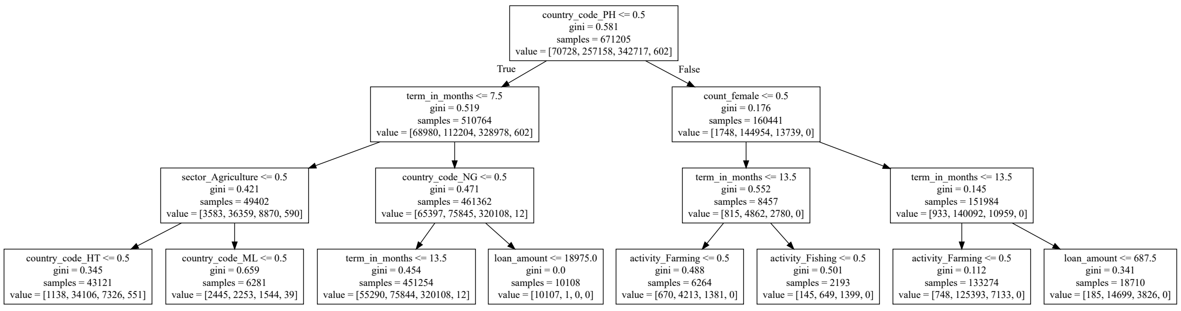
Minimum impurity decrease is one of the most important parameter in decision tree. It controls the minimum reduce in entropy or Gini coefficient for splitting a node, which directly control the degree of generalization. By varying the degree, the performance fluctuates for more than 40%. This is a significant changes using the same prediction model. According to the above result, the optimal minimum impurity decrease is 1e-5.



Maximum depth is another important parameter in decision tree. For the default, which keep splitting until resolved, it has more than 100,000 nodes. After testing for all ranges of maximum depth, the best range of depth lies between 60,000 to 70,000.

Interpretation

To have an accurate prediction, a decision tree must has at least depth of 10,000. The visualization is impossible for such many nodes. In order to visualize the tree, a manual extraction of first 4 layers are visualized.



As from the above illustration, The most important split is the `country_code_PH`, while the other commonly appeared split are `term_in_months`, `sector_Farming` and `loan_amount`. This match with our common sense that instead of region and genders of applicants, the nature of loan application is more important.

Note: the above tree is trained by using default parameter, and trained by all data from one-hot dataset.

File

`dtree.dot` refers to the Microsoft DOT file of the complete extracted decision tree trained with one-hot encoded and standardized dataset.

`dtree2.dot` refers to the Microsoft DOT file of the complete extracted decision tree trained with one-hot encoded dataset.

`dtree3.dot` refers to the Microsoft DOT file of the first three layers of the complete extracted decision tree trained with one-hot encoded dataset.

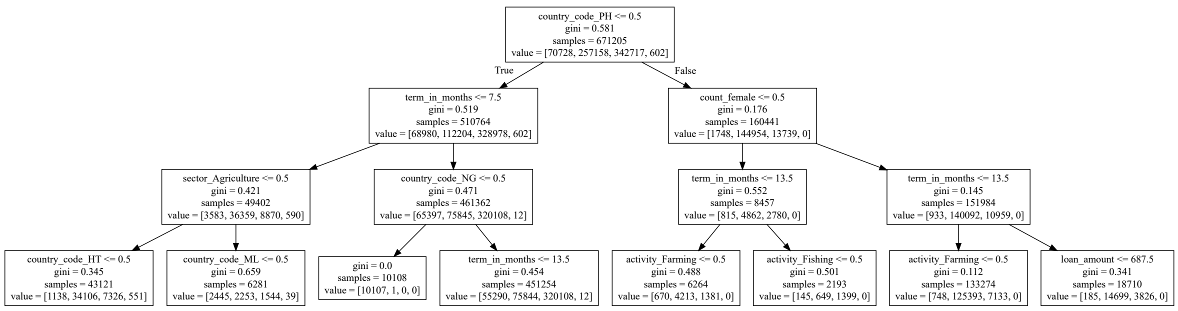
Conclusion

The prediction performance is the best having a decision tree with maximum depth of 61,000 and minimum impurity decrease of 1e-5. The exact result is descripted below:

```
               precision    recall  f1-score   support

      bullet       0.85      0.88      0.86     70728
   irregular       0.92      0.92      0.92    257158
     monthly       0.93      0.92      0.92    342717
      weekly       0.74      0.84      0.79       602

   micro avg       0.92      0.92      0.92    671205
   macro avg       0.86      0.89      0.87    671205
weighted avg       0.92      0.92      0.92    671205

[[ 61996   1683   7034     15]
 [  2526 236842  17638    152]
 [  8158  19401 315147     11]
 [     5     65     25    507]]
```

As one can see, even for `weekly`, which has the smallest cases in `repayment_interval`, has an accuracy over 70%.

The first four layers of the optimal decision tree trained by all data of `kiva_loan_dummied.csv` are visualized as below.



Apart from prediction, the decision tree reveals the most important factors for determining the repayment interval: country, term in months and purpose of loan. The result helps to identify possible irregular repayment and grant loan for reliable borrow request.

file

`dtree4.dot` refers to the Microsoft DOT file of the complete extracted decision tree trained with one-hot encoded dataset with best parameter mentioned in conclusion.