CS6140 Assignments

Instructions

1. In each assignment cell, look for the block:

```
#BEGIN YOUR CODE
raise NotImplementedError.new()
#END YOUR CODE
```

2. Replace this block with your solution.
3. Test your solution by running the cells following your block (indicated by ##TEST##)
4. Click the "Validate" button above to validate the work.

Notes

- You may add other cells and functions as needed
- Keep all code in the same notebook
- In order to receive credit, code must "Validate" on the JupyterHub server

---

In [ ]:
```
1
```

Final Project Report

Write your final project report in the cells below. All written material must be completed prior to your presentation slot. Including figures, appendices, etc. a printed version of this document should be no more than 10 pages and no less than 6 pages.

The following cell is used for overall feedback and deductions for length, content, and style.

YOUR ANSWER HERE

Introduction

(Why would people want to study this dataset and what is the primary task. Find out what the "target" variable means and why the customer is interested in running a competition on this dataset.) 2-3 paragraphs

Many people suffers from getting loans due to insufficiency of credit histories. Sometimes they can face very difficult situations such as unemployment or diseases. These senarios can happen more frequently especially in this pandemic/lockdown situation.

A better prediction for the clients' ability to repay loans would facilitate banks and insurance companies to provide better loan experience for the customers as well as reducing the company's risk due to bad loans.

In this dataset, the training application data comes with the TARGET indicating 0: the loan was repaid or 1: the loan was not repaid. Although banks and insurance companies are currently using various statistical and machine learning methods to make these predictions, it is still challenging to unlock more potential of their data and build a more powerful prediction tool. Doing so will help the clients with payment abilities get the loan, thus could enable them live though a difficult situation or even become successful with their business.

Propsed method

(Present an 1-paragraph description of your method and why you believe it is better that the other things you have tried)

In this project, we propose a random forest tree learner on 10 selected freatures from the training dataset. The training dataset was pre-proceesed with downsampling transformation to make a balanced dataset. The random forest model has 20 number of trees, with minimum node size of 500 and maximum tree depth of 5. This model results in a test AUC of 0.72.

Related Work

(Find (5-6) examples of people who have worked on similar dataset from the literature. Note: Literature == Published paper in a conference (not stack overflow). Briefly describe in 1-2 sentences the kinds of features, algorithms, or other methods they applied. Also explain why you believe your method is better. Provide a numbered reference id that will appear later in the references section. 3-5 paragraphs

Credit risk prediction has been a hot topic. Credit risk prediction models seek to predict quality factors such as whether an individual will default (bad applicant) on a loan or not (good applicant). Twala (2010) [1] explored the predicted behaviour of five classifiers for different types of noise in terms of credit risk prediction accuracy.

Wang. et, al (2012) [2] proposed a hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine. Experimental results reveal that their approach can be used as an alternative method for enterprise credit risk assessment.

Abedin (2018) [3] conducted credit default prediction using a support vector machine and a probabilistic neural network. They concluded that neural network based method had a general better performance.

Related implementations

(Find (2-3) examples of what people in Kaggle have done on this particular dataset [2] (https://www.kaggle.com). Reference the URL of their kernel, post, etc. Describe in 1-2 sentences what they have done and why you think your method is better.) 2-3 paragraphs

The first related implementation was by Will Koehrsen [4]. He combined training dataset from `application_train` and `bureau`. He did feature engineering with dropping missing values and one-hot-encoding. He trained and tested a light gradient boosting model (LGBM) with cross validation that resulted in auc=0.759.

Data Analysis

(Data Analysis: Describe the data analysis you have completed, include 1-2 plots of the most useful features or learnings you have obtained from the dataset. Do not include the code, but do include formulas to anything you have calculated such as different feature combinations, feature selection, or analysis methods. You must use at least one clustering algorithm we have seen in class for an analysis of the data. Provide a link to the specific notebook cell in previous notebooks as a reference.) 5-6 paragraphs

First I studies the target distribution, where ~92% is labeled `0` and ~8% as `1`. This is reasonable because the data comes from approved loans and most of which would be successfully repaid. This gives a hint that down sampling can be a good option for data pre-processing.

I explored the features on their distributions and correlations. For distributions, I studied Age and ext_source features. The Age distribution between targets has a difference where clients who unpaid loans were skewed toward younger ages. For correlations I studied ext_source and amount features including `amt_income_total`, `amt_credit` and `amt_goods_price`.

Other features such as Gender, `name_education_type` and `emergency_status` are also important.

Proposed Methods

(Describe the ML algorithms you used for questions 3.1 (part-3.ipynb#Question-3.1), 4.1 (part-3.ipynb#Question-4.1), and all others. Focus on the formulas, any feature extractions, parameter tuning, etc. Explain how the algorithm works. E.g., if you used a decision, don't say "I used a decision tree", explain briefly how a decision tree works and why it was ideally suited for the dataset you chose.) 3-5 paragraphs

Here is a formula:

$$\arg\min_{w\in\mathbb{R}^d} w^2 + \sum_i \xi_i$$

1. Logistic Regression Learner

Logistic regression can be used for classification problems where the labels are 0 or 1. To train a logistic regression learner, we compute the weight vector w^* that maximize the log likelihood function. We also used L2 norm regularization to avoid overfitting. The log-loss function for logistic regression with L2 regularization is:

$$LL(X, y, w) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \ln \sigma_i + (1 - y_i)\ln(1 - \sigma_i) + \frac{\lambda}{2}||w||^2)$$

where $\sigma_i = \sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}}$, λ is the regularization parameter.

To calculate $w*$, we use stochastic gradient descent method. Until w converges, we update w for each batch of examples based on its gradient: $w = w\eta\frac{\partial LL}{\partial w}$, where η is the learning rate and

$$\frac{\partial LL(X, y, w)}{\partial w^{(j)}} = -\frac{1}{N}\sum_{i=1}^{N}(y_i - \sigma_i)x_i^{(j)}$$

To predict an example x_i, we have

$$\hat{y}_i = \sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}}$$

Logistic regression is sensitive to the value of numeric features and is not applicable to non-numeric features. So we used z-normalization and log-normalization for numeric features and we used one-hot encoding for categorical features.

Logistic regression is suited for the dataset in this project since it is ideal for binary classification. We can also use the weights learned from logistic regression to get insights of different features.

2. Decision Tree Learner

A decision tree is a prediction model based on tree datastructure. Each node contains a subset of the training dataset, where the root node contains all data and a leaf node is labeled with one class label. To train the tree model we grow the tree node with a greedy algorithm: we explore all possible splits among all features at the current node and select the best split as the one with the maximum informatioin gain. The information gain of splitting node q on feature V can be calculated as:

$$IG(q, V) = H(q) - \sum_{i=1}^{|V|}\frac{N_i}{N_q}H(i)$$

where $H(i) = -\sum_j p_j \log(p_j)$ is the entropy of the ith node, N_i is the example size in the ith node.

To predict an example from the tree model, start at the root and follow the split criteria until reaching the leaf node, where the class label would be the prediction.

One property of decision tree models is the invariance to monotonic transformations of features. For example, z-normalization and log-transformation are strictly monotonic functions: $x_1 < x_2 => f(x_1) < f(x_2)$, therefore would give rise to the same tree model. This property also makes tree models robust to outliers in the dataset.

There are two ways to control the size of the tree. A pre-pruning strategy sets a stopping criterion to prevent the tree from further growth. A post-pruning strategy constructs a very deep tree at first, then cut the tree to avoid overfitting. In this project we follow the pre-pruning strategy because of the lower time complexity.

3. Random Forest Learner

A random forest learner uses bagging strategy to train a number of trees and take the average prediction. Random forest learners can improve performance and lower the variace of a single decision tree learner. To prevent all trees from learning the same result, each tree was trained with 3 randomly selected features to reduce the correlation between pairs of trees in the forest.

Analysis

(Provide some insights as to why you think that the proposed algorithms and features are good for this dataset. Explain whether you believe these are general properties that might be helpful for similar datasets--what makes them similar and why. What about this dataset made your solution successful. Could we use this for other datasets, if so, what types and why?) 3-5 paragraphs

The features selected would be useful for this dataset since they have a clear difference of distribution between different targets. Features like `Age` and `Gender` might be helpful for similar datasets since they may reflect some behavioral charisteristics among clients with different age and gender. Features like `amt_credit` , `amt_income` and `emergency_state` are also important, since they reflect the clients' financial status directly, which could provide a good reference for other financial datasets.

Logistic regression is suited for the dataset in this project since it is ideal for binary classification. We can also use the weights learned from logistic regression to get insights of different features.

Tree-based methods are also useful for this dataset. We can learn important features from the model and form a prodcedure for credict default prediction.

Experimental Setup

(Did you use all the data, cross-validation, training / test split, etc? Give enough details on how you setup the experiment so that your colleague can read this section and write their own algorithm to produce the same setup. Provide a link to the cells in the notebooks that contain the experimental setup.) 3-4 paragraphs

1. Building the Training and Evaluation Datasets

To build dataset for training and evaluation, features and target was selected from `application_train` dataset. Since the dataset is biased, a downsampling with a sample rate of 0.08 is applied when building the training dataset. For building the evaluation dataset, same features were selected while downsampling was not applied.

2. Logistic regression learners

Logistic regression learners were trained with following features:

```
ext_source_1, ext_source_2, ext_source_3, amt_income_total, amt_cre
dit,
days_birth, code_gender
```

Following transforms were applied in sequence with corresponding features:

- age_range_as_vector: `days_birth`
- z_score: `ext_source_1, ext_source_2, ext_source_3`
- mean_imputation: `ext_source_2, ext_source_3`
- one_hot_encoding: `code_gender`
- log_transform + z_score: `amt_income_total, amt_credit`
- L2 Normalization: all selected features

The logistic regression learner with highest AUC was trained with parameters:

```
"regularization"=>0.01,
"learning_rate"=>0.01,
"epochs"=>1,
"batch_size"=>200
```

The notebook containing the experimental setup can be found in project part3-2:
http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-2.ipynb
(http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-2.ipynb)

3. Decision Tree-Based Models

Decision tree_based models were trained with following features:

```
ext_source_1, ext_source_2, ext_source_3, amt_income_total, amt_cre
dit, days_birth, code_gender, flag_own_car, name_education_type, em
ergencystate_mode
```

No transform was applied before training.

A random forest learner with highest AUC was trained with parameters:

```
"num_trees": 20,
"min_size": 500,
"max_depth": 5
```

The notebook containing the experimental setup can be found in project part3-4:
http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-4.ipynb
(http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-4.ipynb)

Results

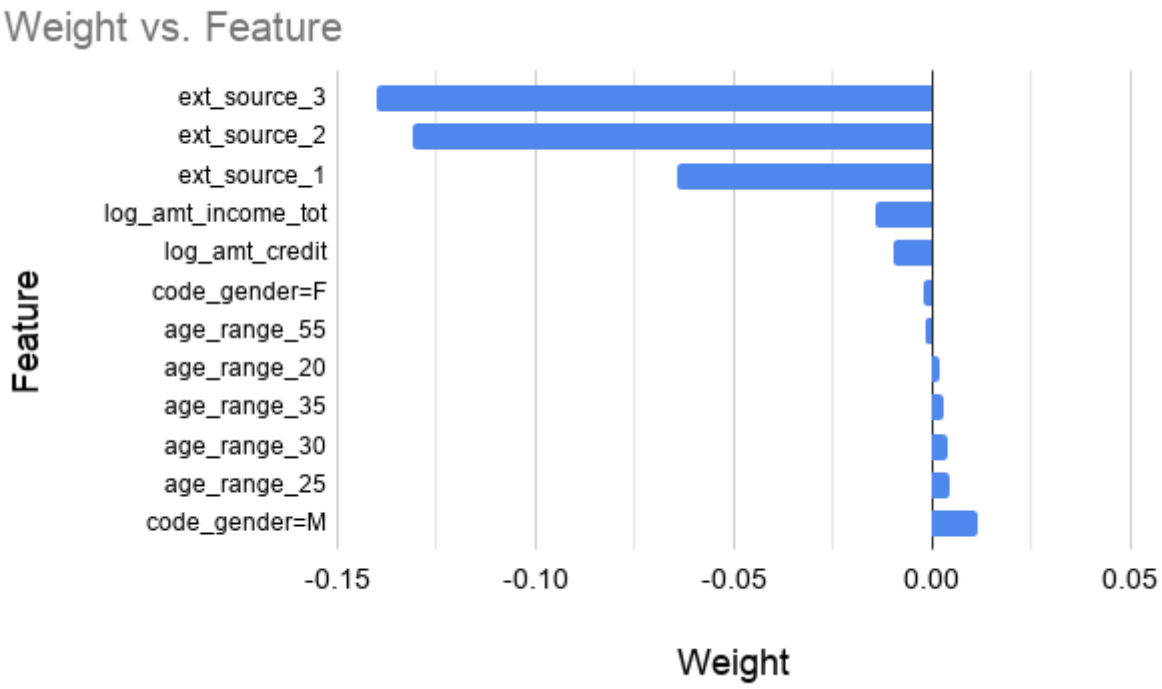
(Write a table containing the results of your experiments, which were calculated in the notebooks. Include all algorithms included in questions 3.1 (part-3.ipynb#Question-3.1) and above in Part 3 of the notebooks. Provide some interpretation of these results. Do you think you could have done better? If so, why did you not pursue those ideas? Add any pictures you think approprate here.) 5-6 paragraphs

In this project, metrics are analysed based on the area under ROC(AUC).

1. Results for Logistic Regression Models

The logistic regression model results in a cross validation test AUC of 0.71 and the evaluation test AUC of 0.68.

The figure below shows the weights for most important features:



To interprete from weights trained by this linear model, down sampling transformation is necessary, otherwise most weights would be negative since most examples was labeled negative. From the above weights we can see the feature with most negative weight is `ext_source_3` and weight with most positive weight is `code_gender=M` . We can also see a correlation of age_range with the weights: the younger the age, the larger the weight. The correlations of target with features of `ext_source` , `age` and `gender` is consistent with pre-training data analysis.

More combinations of features, transforms and parameters could be tried to improve the result for the linear model. However, this is not done because the evaluation test AUC cannot pass 0.7 and tree-based models would be more worth to explore.

The notebook containing results for linear models can be found in project part3-2:
http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-2.ipynb
(http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-2.ipynb)

2. Results for Tree-Based Models

First, I started with decision tree models. By tuning node size from 100 to 500 and tree depth from 5 to 10, the cross valication test auc was in range of (0.689,0.701), the evaluation test auc was in range of (0.686, 0.691). Since there is no significance improvement on any combinations, I chose the fastest one (size=500, depth=5) for the random forest model.

For the random forest model, I tried num_trees ranging from 5 to 20. As number of trees increases, the cross validation test AUC increases from 0.706 to 0.726, and eval test AUC increases, too. This makes sense because the more trees we trained, the more combinations of features we would have, therefore the result would be better.

The best random forest model has `num_trees = 20` and results in a cross validation test AUC of 0.726 and the hidden test AUC of 0.723.

The notebook containing results for random forest models can be found in project part3-4: http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-4.ipynb (http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-4.ipynb)

Conclusion

(Summarize your findings. If someone wanted to use your solution, which would you recommend? What could you do if you had more data, etc. What should a company seeking to run this at high scale choose if they were to use your method.) 3 paragraphs

In this project, I experimented with logistic regression models, and decision tree-based models on different combinations of parameters. All models showed the most important features to be `ext_source` , `age` , `amt_credit` , etc.

For the model training, I recommend to first run a fast linear model and get insights on different weights. Besides, I recommend the random forest model among all models I experimented with because I lowers the bias and variance of the prediction, thus would give a better result.

If I had more data, I would try more different combinations of features and models. Besides the models listed above, neural network models and combined models are also popular for credit risk prediction.

References

(Add a numbered list of the referenced articles, notebooks, etc. that you cited in the above notebooks. Pay attention that the numbers you used correspond to the list below.)

Consider the MLA or APA style, which should be available in Google Scholar.

Example

[1] (https://arxiv.org) Bob Smith, John Doe. My amazing method. In _Proceedings of WWW 2018_, Lyon France, 2018.

[1] Twala, Bhekisipho. "Multiple classifier application to credit risk assessment." Expert Systems with Applications 37.4 (2010): 3326-3336.

[2] Wang, Gang, and Jian Ma. "A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine." Expert Systems with Applications 39.5 (2012): 5325-5331.

[3] Abedin, Mohammad Zoynul, et al. "Credit default prediction using a support vector machine and a probabilistic neural network." Journal of Credit Risk, Forthcoming (2018)

[4] https://www.kaggle.com/willkoehrsen/introduction-to-manual-feature-engineering (https://www.kaggle.com/willkoehrsen/introduction-to-manual-feature-engineering)

Appendix

Add links to your part 1, part 2, and part 3 notebooks (using absolute links). Add anything else you want to here.

Part3 notebooks:

http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-1.ipynb
(http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-1.ipynb)
http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-2.ipynb
(http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-2.ipynb)
http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-3.ipynb
(http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-3.ipynb)
http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-4.ipynb
(http://notebooks.learnml.cool:31757/user/yutingsun/notebooks/final-project-3/part-3-4.ipynb)

In [ ]:
```
1
```