Predict a developer’s salary based on questionnaire

EE 660 Course Project

Project Type: Design a system based on real-world data

Number of student authors: 1

Muhao Guo and muhaoguo@usc.edu

Dec 6, 2019

1. Abstract

Salary always be an important issue one would consider about before they enter to a new company. How to predict one’s salary? It is a considerable question for both employers and employees. In my project, I transformed this practical life problem to a machine learning classification problem. Based on more than 60,000 questionnaires, through data preprocessing including feature selection, feature expansion and feature transformation I get a formal dataset. Then split this formal dataset to training dataset, validation dataset and test dataset. After that, I designed my system based on Random Forest, Support Vector Classification and Logistic Regression to solve this problem. In the end, my system had an over 89% accuracy to predict correctly.

2. Introduction

2.1. Problem Type, Statement and Goals

As one of the most important factors an employee usually considered, the salary level always plays a decisive factor about whether an employee want to enter to a company or not in workplace. How much salary an employee should get is a considerable question for both employee and employer. In my project, I got a dataset from Stack Overflow Developer Survey Results 2019, which based on over 60,000 questionnaires which collected every respondent’s age, gender, salary, work hours, favorite operation system and so on, total 83 survey questions. Based on the dataset, I extracted the salary column as label “Y”, the rest data used for sample “X”. Then, I designed a classification problem by part the label into five classes to present different salary levels. They are “less than 10,000”, ”10,000-50,000”, ”50,000-100,000”, ”100,000-200,000”, ”more than 200,000” respectively, the unit is dollars per year. My project’s goal is to predict one’s salary level based on other given information.

The problem is important and interesting since it is about what we most care about – salary. The problem is also nontriviality because:

(1) the High dimensionality of feature space. There are total 83 features in the original dataset. Even if I deleted some useless feature, however, I also add some features by one-hot encoding. There are still 60 features in the final training and test dataset.

(2) Nonlinear behaviors. I cannot separate the whole data into five classes based on a linear model. Through perceptron classification, I proved that the dataset is not linear separately.

(3) Significant amounts of preprocessing required. Since the original dataset is based on the questionnaires, there are many informal data. I selected the useful features column and deleted the blank rows. For some features, I transform string type to numbertic type, for some features, I used one hot encoding method and for other features, I used label encoding method. After these steps, I got a formal dataset.

2.2. Literature Review (Optional)

2.3. Our Prior and Related Work (Mandatory)

Prior and Related Work - None.

2.4. Overview of Our Approach

I used Random Forest, Logistic Regression with l1 penalty, Logistic Regression with l2 penalty and Support Vector Machine. I used both F1 score and general score to compare the results. For multilabel, I used one hot encoding and label encoding.

3. Implementation

3.1. Data Set

The dataset was from "Stack Overflow Developer Survey Results 2019". Based on a questionnaire, every developer answered several questions and provide some personal information. After collected and organized these questionnaires, we get an original dataset. In the first step, I selected 37 useful features out of the total 83 features. The preliminary selected features are as follows:

Feature Name	Type	Cardinality	Means
Hobbyist	binary		Coding is a hobby?
OpenSourcer	categorical	3	How often do you contribute to open source?

OpenSource	categorical	3	Feel about the quality of open source software
Employment	categorical	7	Employment status
Country	categorical		country
EdLevel	categorical	9	Highest level of formal education
UndergradMajor	categorical	12	Major
OrgSize	categorical	9	Size of company
DevType	categorical	24	Career
YearsCode	integer		How many years have you been coding
Age1stCode	integer		First coding age
YearsCodePro	integer		Years you coded professionally
CareerSat	categorical	5	How satisfied with career
JobSat	categorical	5	How satisfied with current job?
MgrIdiot	categorical	4	How confident are you know your manager
MgrMoney	categorical	3	Need to be a manager to earn more money
MgrWant	categorical	3	Want to become a manager
JobSeek	categorical	3	Current job seeking status
LastHireDate	categorical	6	Last time you took a job
ConvertedComp	integer		Salary
WorkWeekHrs	integer		Hours per week do you work
WoekPlan	categorical	4	Structured or planned work
WorkRemote	categorical	7	How often do you work remotely
WorkLoc	categorical	3	Where would you prefer to work
ImpSyn	categorical	5	Rate your own level of competence
CodeRev	categorical	3	Review code as part of your work
CodeRevHrs	integer		Hours per week spend on code review
PurchaseWhat	categorical	3	Influence over new technology purchases
OpSys	categorical	4	Primary operating system
Containers	categorical	5	How do you use containers
BlockchainOrg	categorical	6	Have blockchain technology?
BetterLife	binary		Better life than before
ITperson	categorical	4	"IT support person" for your family
Extraversion	categorical	3	Chat method
Age	integer		Age
Gender	categorical	3	Gender
Dependents	categorical	3	Have dependents?

3.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

1. select the useful features:

In the original dataset, there are total 83 features columns, I selected the useful features intuitively. The features name and corresponding column indexes are:

Column index	Feature name	Column index	Feature name	Column index	Feature name
2	'MainBranch'	17	'CareerSat'	39	'CodeRev'
3	'Hobbyist'	18	'JobSat'	40	'CodeRevHrs'
4	'OpenSourcer'	19	'MgrIdiot'	43	'PurchaseWhat'
5	'OpenSource'	20	'MgrMoney'	55	'OpSys'
6	'Employment'	21	'MgrWant'	56	'Containers'
7	'Country'	22	'JobSeek'	57	'BlockchainOrg'
9	'EdLevel'	23	'LastHireDate'	59	'BetterLife'
10	'UndergradMajor'	32	'ConvertedComp'	60	'ITperson'
12	'OrgSize'	33	'WorkWeekHrs'	63	'Extraversion'
13	'DevType'	34	'WorkPlan'	78	'Age'
14	'YearsCode'	36	'WorkRemote'	79	'Gender'
15	'Age1stCode'	37	'WorkLoc'	83	'Dependents'
16	'YearsCodePro'	38	'ImpSyn'		

2. Based on the “MainBranch” column. I dropped the sample belong to students/hobby, only selected the professional developer since the goal of the project is predicting a developer’s salary.

3. Based on the “DevType” column. I added 24 new features through one hot encoding method. Then, I deleted the original column “DevType”. All of the 24 new features are binary type, as following:

New feature name	Type	New feature name	Type
Academic researcher	binary	Developer, QA or test	binary
Data or business analyst	binary	DevOps specialist	binary
Data scientist or machine learning specialist	binary	Educator	binary
Database administrator	binary	Engineer, data	binary
Designer	binary	Engineer, site reliability	binary
Developer, backend	binary	Engineering manager	binary
Developer, desktop or enterprise applications	binary	Marketing or sales professional	binary
Developer, embedded applications or devices	binary	Product manager	binary
Developer, frontend	binary	Scientist	binary
Developer, fullstack	binary	Senior Executive (CSuite, VP, etc.)	binary
Developer, game or graphics	binary	Student	binary
Developer, mobile	binary	System administrator	binary

4. Based on the “YearsCode” column. First, I handled the special cases: if the data is “Less than 1 year”, then I transformed it to “0”; if the data is “More than 50 years" then I transformed it to”50”. After that, I transformed str type to int type directly.

5. Based on the “Age1stCode” column. First, I handled the special cases: if the data is “Younger than 5 years”, then I transformed it to “5”. After that, I transformed str type to int type directly.

6. Based on the “YearsCodePro” column. First, I handled the special cases: if the data is “Less than 1 year”, then I transformed it to “0”. After that, I transformed str type to int type directly.

7. Based on the “ConvertedComp” column. This is the label “Y” column. First, if the data is “0” , I dropped this sample directly since one developer’s salary is 0 dollars/year seems unrealistic. Second, to make the problem a classification problem, I set different classes based on the range of the salary. If the data less or equal to 10000, it belongs to the first class “less than 10000”; if the data large than 10000 and less or equal to 50000, it belongs to the second class “10000-50000”; if the data large than 50000 and less or equal to 100000, it belongs to the third class “50000-100000”; if the data large than 100000 and less or equal to 200000, it belongs to the forth class”100000-200000”; if the data large than 200000, it belongs to the fifth class ”more than 200000”.

8. Based on the column “WorkWeekHrs” column. I transformed the str type to int type.

9. Based on the column “CodeRevHrs” column. I transformed the str type to int type.

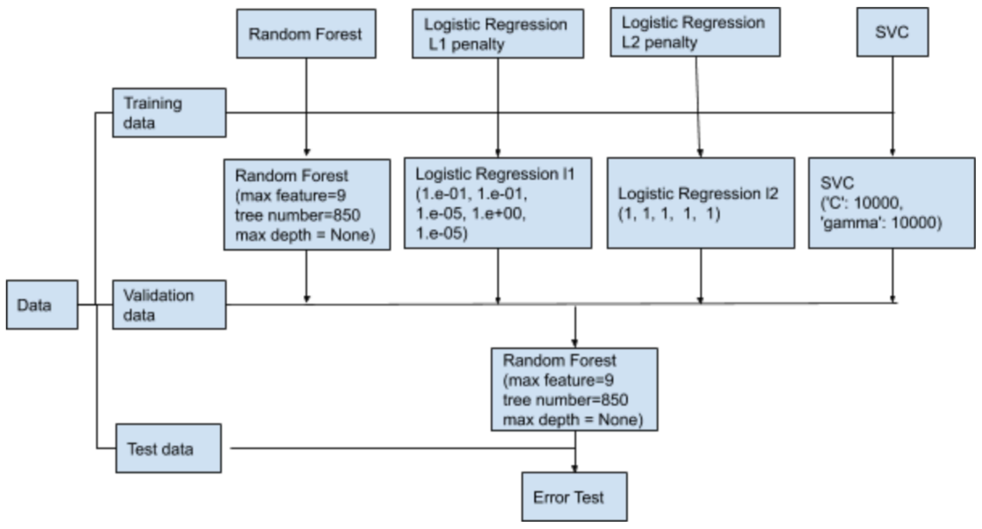
10. Based on the column “Age” column. If the data is “0”, I dropped this sample since it is unrealistic.

11. For the rest 29 features list below, all of them are categorical type and I used “labelencode” to transform the string data to different int type data with different values represent different labels respectively.

Feature name	Feature name	Feature name
Hobbyist	MgrIdiot	PurchaseWhat
OpenSourcer	MgrMoney	OpSys
OpenSource	MgrWant	Containers
Employment	JobSeek	BlockchainOrg
Country	LastHireDate	BetterLife
EdLevel	WorkPlan	ITperson
UndergradMajor	WorkRemote	Extraversion
OrgSize	WorkLoc	Gender
CareerSat	ImpSyn	Dependents
JobSat	CodeRev	

12. After the processing above, I split the features “X” and label “Y”, there are total 59 features in X.

3.3. Dataset Methodology


Random Forest
Logistic Regression L1 penalty
Logistic Regression L2 penalty
SVC
Training data
Random Forest (max feature=9 tree number=850 max depth = None)
Logistic Regression l1 (1.e-01, 1.e-01, 1.e-05, 1.e+00, 1.e-05)
Logistic Regression l2 (1, 1, 1, 1, 1)
SVC ('C': 10000, 'gamma': 10000)
Data
Validation data
Random Forest (max feature=9 tree number=850 max depth = None)
Test data
Error Test

There are 9900 ,5658, 3301 data points used for training, validation, and test respectively.

Then I used training data set for the following four models:

1. **Random Forest model:**

I used cross validation to choose the best parameters ”n_estimators”, “max_features”, and ”max_depth”.
The tree number 'n_estimators', start from 600 to 1200, the interval is 50. There are 12 different “n_estimators”.
The max features “max_features”, start from 5 to 10, the interval is 1. There are total 5 different “max_features”.
The max depth ”max_depth”, there are 3 different values: None, 10 and 20.
These three cross validation loops were nested, for all the cross validation, I used 10-folder validation.

2. **Logistic Regression with L1 penalty model:**
I used cross validation to choose the best parameter C, which describes the inverse of regularization strength. There are total 11 different C values as following:
10 ^ -5, 10 ^ -4, 10 ^ -3, 10 ^ -2, 10 ^ -1, 1, 10, 100, 10 ^ 3, 10 ^ 4, 10 ^ 5
Also, I use 10 floder validation.

3. **Logistic Regression with L2 penalty model:**
I used cross validation to choose the best parameter C, which describes the inverse of regularization strength. There are total 11 different C values as following:
10 ^ -5, 10 ^ -4, 10 ^ -3, 10 ^ -2, 10 ^ -1, 1, 10, 100, 10 ^ 3, 10 ^ 4, 10 ^ 5
Also, I use 10 floder validation.

4. **Support Vector Classification model:**
I used cross validation to choose the best parameter “gamma” and “C” which represent Penalty parameter and the parameter of Gaussian Kernel respectively.There are total 5 different “gamma” and 5 different “C” as following:
'gamma': 10000, 1000, 100, 10, 1, 0.1;
‘C’: 10000, 1000, 100, 10, 1, 0.1
Also, I used 10 folder validation.

Then, validation dataset was used:

After I chose a best Random Forest model, a best Logistic Regression with L1 panelty model , a best Logistic Regression with L2 panelty model and a best SVC model, I used validation dataset for these four model and chose the best one, it is the Random Forest model ('n_estimators'=850, ‘max_features’=9, ‘max_depth’=None)

Finally, I used the test data set for the best one model and got the test score.

3.4. Training Process

Linear Regression with L1 penalty:

Loss function of Linear Regression with L1 penalty as following:

$$\min_{\omega,c} \parallel \omega \parallel_1 + C \sum_{i=1}^{n} \log\left(\exp\left(-y_i(X_i^T \omega + c)\right) + 1\right)$$

We want to minimize the loss function and add the l1 penalty to avoid overfitting. I used cross validation to choose the best parameter C, which describes the inverse of regularization strength. There are total 11 different C values as following:

10 ^ -5, 10 ^ -4, 10 ^ -3, 10 ^ -2, 10 ^ -1, 1, 10, 100, 10 ^ 3, 10 ^ 4, 10 ^ 5
However, this model did not worked well since the data set is not linear separatable.

Linear Regression with L2 penalty:

Like Linear Regression with L1 penalty, we want to minimize the loss function and add the l2 penalty to avoid overfitting. The loss function as following:

$$\min_{\omega,c} \frac{1}{2}\omega^T \omega + C \sum_{i=1}^{n} \log\left(\exp\left(-y_i(X_i^T \omega + c)\right) + 1\right)$$

I used cross validation to choose the best parameter C, which describes the inverse of regularization strength. There are total 11 different C values as following:

10 ^ -5, 10 ^ -4, 10 ^ -3, 10 ^ -2, 10 ^ -1, 1, 10, 100, 10 ^ 3, 10 ^ 4, 10 ^ 5
However, this model also did not worked well since the data set is not linear separable.

SVC:

I used Support Vector Classifier because the data set is not linear separable and Support Vector Classifier can worked well under this situation. The objective of SVC is to fit the data we provided and return a best fit hyperplane which can divide the dataset into different class.

I chose the general kernel function “RBF” kernel and used cross validation to get the best parameter “gamma” and “C”, which represent the kernel coefficient for “RBF” and regularization parameter respectively. The effect of the “C” is to avoid overfitting. The values of “gamma” and “C” I chose are list as following:

'gamma': 10000, 1000, 100, 10, 1, 0.1; 'C': 10000, 1000, 100, 10, 1, 0.1

However, this model did not work well.

Random Forest:

Random Forest’s predict result based on vote of every tree in the forest.

$$\hat{y}(\underline{x}) = \underset{C}{\operatorname{argmin}} \sum_{b=1}^{B} I(\widehat{y^b}(\underline{x}) = C)$$

Where I represent the indicator function. $\widehat{y^b}(\underline{x})$ represent the predict result of each tree.

I used Random Forest Classifier since it can process the data set with features space well. I used cross validation to choose number of trees in the forest, number of features every iteration and the max depth of the tree. Because there are total 59 features and about half of these features have multilabel, it should have a relatively large tree number. Then I set the interval of tree number 50 and the range of tree number start from 600 to 1200. Since the number of features should less than the total features, I also used cross validation to choose the number of features every iteration, the number of features start from 6 to 10 and the interval is 1. For the depth of the tree, I use cross validation as well. I set three values for the depth of the tree: None, 50 and 100.

3.5. Model Selection and Comparison of Results

I have four potential models, the Random Forest model, Logistic Regression with l1 penalty, Logistic Regression with l2 penalty and Support Vector Classification. I used cross validation to choose the best model from the four potential models respectively. Then I used validation data set to evaluate the four selected models and finally I got only one best model.

The result of different models under different datasets are as follow:

	LR with L1	LR with L2	SVC	Random Forest
Best Training score	0.467	0.478	0.354	1
Validation score	0.438	0.465	0.350	0.984
Validation F1 score	0.385	0.433	0.181	0.893
Test score	NA	NA	NA	0.891
Test F1 score	NA	NA	NA	0.890

From the results we know, the Random Forest model worked much better than the other three models. Through comparing the validation F1 score, we choose the Random Forest model.

Same as expected, the training error of the Random Forest model got 1 since the max depth is "None", which means the tree in the forest can grow as deep as possible. The F1 score and general score of the Random Forest model are almost same, which means the slightly imbalance of classes had little effect on the predict results.

4. Final Results and Interpretation

My system's performance as following, my model's predict result have 89% accuracy.

Test predict score	0.891
Test F1 predict score	0.890

It means if given some information about a developer, my system can predict his or her salary belong to a range with the accuracy above 89%. The out of sample performance should be 0.89 accuracy or less than 0.11 error rate.

However, In the original data, the five classes' distribution as following:

Class	Proportion
Class1 ("less than 10000")	0.0683
Class2 ("10000-50000")	0.304
Class3 ("50000-100000")	0.352
Class4 ("100000-200000")	0.187
Class5 ("more than 200000")	0.089

This means the baseline result at most be 0.352. If you guess one developer's salary belong to a range, you only have at most 35.2% probability to get the right result. So, my system improved at less 53.8% accuracy for this classification problem.

Interpretation:
In this problem, Random forest algorithm have more advantages to handle the dataset with a large number of features than Logistic Regression and SVC. What's more, the Random Forest random selects features and gets a predict result from the vote of all the trees, this makes the random forest always performs well and do not overfit. In my training process, I got 0 error rate in Random Forest model and it still worked well in validation dataset and test dataset. It showed that the Random Forest can overcome the overfit in some cases. However, when the number of features are large enough, it will cost much time to training the Random Forest because parameters like "depth", "number of trees" will increasing rapidly. For example, in my system, there are total 59 features and the number of trees reach to 850, I used cross validation to select best parameters and it will cost me several hours to train once.
To improve my system in the future, first, I should consider using sequential cross validation to improve the efficiency of my system rather than nested cross validation. Second, due to the limitation of time and energy, I dropped some features that I thought useless, intuitively. If I added these features, especially use one hot encoding, the number of features would attained to several hundreds. I think the training time would attained to 24 hours. However, these features I dropped may have surprising effects on my system. So, I will research the influence of these features I neglected in the future.
Third, I did not expect the SVC algorithm worked so poorly. I think maybe it because I chose the unfitted kernel or the distribution of the data do not fit SVC. I will research this in the future.

5. Contributions of each team member

Personal project

6. Summary and conclusions

My system based on Random Forest worked well on predict a developer's salary. Since what I did was a classification problem, next I will try to a regression prediction of one's salary.

7. References

NA

8. Appendix