Who Earns the most?
Salary Prediction for Programmers

Team 13
Chen Yulin 3035447398
LU Kexin 3035329869
Wu Zhengzhe 3035449279

April 2019

Abstract

A datasets about programming related questions was uploaded on Kaggle by Stack overflow in 2018 which contained 129 explainatory variables and around 100K observations. We used this datasets figure out an interesting question: predict salaries for programmers. First, we did data exploration to simplify this dataset and made it handle by computers. Next, we build salary prediction models using different machine learning methods such as Lasso, Random Forest, and Neural Network. Then, we found some interesting results by analysing our models. Finally, we listed some potential problems that need to be solved in the future.

Contents

1 Introduction

With the development of Artificial Inteligence, Machine Learning, Big Data and many other technologies, more and more students regard IT developer and data scientist as target occupations after graduation. We noticed this trend and wanted to help those students to figure out two concerning questions: what kind of people could earn the most in IT industry and could we predict the salary for programmers using related information?
In January 2018, Stack overflow conducted an online questionary survey whcih contained 129 questions about programming and got information from around 100K developers. We obtained this dataset from Kaggle to build our three regression models: Lasso model, Random Forest model and Neural Network model. These three models all performed pretty well with R^2 around 0.7. We analyzed these three models and found some important explainatory variables which significantly influenced the prediction results such as 'Age', 'YearsOfCoding', 'CompanySize' and so on. We further analyzed the relationship between these variables and the salary and made some interpretation. For example, generally large companies gave programmers significantly higher salary than startups because large companies had much more sufficient capital and they were more willing to use high salary to attract talents.

2 Data Exploration

The shape of original dataset is (98855,129) and it has lots of missing values, subjective columns, multiple choice catogorical variables and many other problems. So data preprocessing is urgently needed. We mainly did this work from two aspects: columns and rows.

2.1 Preprocessing Columns

2.1.1 Dealing with Subjective Columns

In order to get an objective model, we dropped subjective features such as 'LanguageDesireNextYear', 'HopeFiveYears', 'HackathonReasons' and so on. Besides, all of the ordinal features were dropped because different people have different evaluation criterion. For instance, there was a question about ranking 10 job opportunities from 1 to 10 in order of importance. Some people could put slary at the first place while others may think work-life balance is more important.

2.1.2 Dealing with Multiple-choice Columns

Many questions in questionnaire provided more than one choices. For example, 'IDE' provided more than 50 choices and developers could choose many of them because it was normal for a developer to use many IDEs at the same time. For all of these columns, we created a new column for each choice and used get_dummies function in pandas package to transform the string type value into 0 and 1 which could be easily processed by computers.

2.1.3 Dealing with Highly Various Columns

Values in some columns were highly various which could produce lots of new columns after the above step. In order to simplify our model, we grouped some columns with similar characteristics into a new column. For instance, we grouped different countries into developed country and developing country because generally salary level in developed country is significantly higher than that in developing countries.

2.1.4 Dealing with Collinearity

Several columns in the dataset were highly correlated with other columns. So we dropped those columns which have more than 0.95 correlations with other columns to simplify our model.

2.2 Preprocessing Rows

2.2.1 Dealing with NA values

First, we dropped some useless rows which had too much missing values(>0.9). For the rest missing values, we filled in 0 in categorical features and filled in median values in numerical features.

2.2.2 Dealing with outliers

After analyzing the dataset we found some extremely high or low salary values such as 2 million US dollars per year. So we dropped outliers which falled in 1.5 inter quantile range(IQR) below the first quantile and above the third quantile.

3 Machine Learning Methods

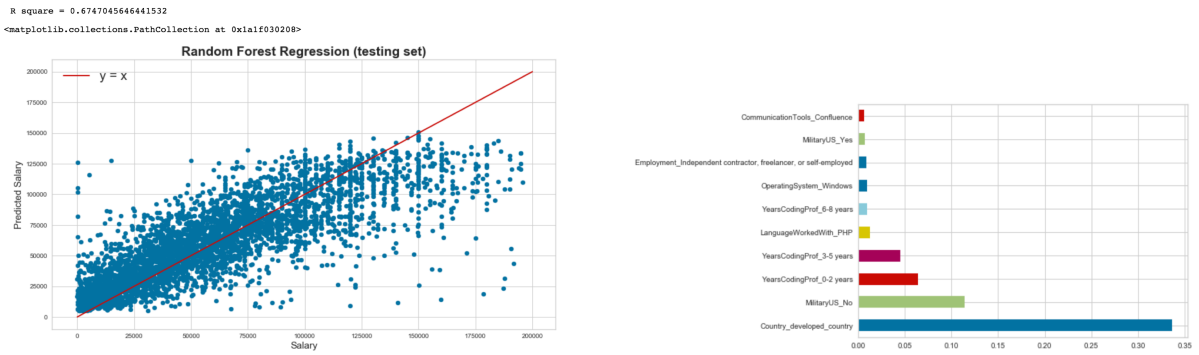
In our analysis of the programmers' salary data, we assume subjective answers will not provide a good explanation with the objective prediction model. Given the assumption, the models we employ are Lasso regression model, random forests (RF) regression method and multilayer perceptron (MLP).

We construct the models to capture the most influential features to programmers' salary. To obtain a more predictive and valid model, we analyze the results from the first fit which used all the remaining features after data exploration to construct the model. Then, we divide our data set according to the observations and obtain two sub-models for developed countries and developing countries separately, later we perform model diagnostic to verify that our division is valid and reasonable.

3.1 Full Model

In this dataset, there are 41 categorical variables that are under consideration to construct a salary prediction model. The modeling process becomes complicated after getting dummies for all the 41 variables,

where we get 413 features in total. Therefore, it is crucial to choose a method that can find the most explanatory features before further investigate into the dataset. Random forest with its auxiliary variable importance measure (VIM) is one of the best methods to use at the first stage of examining features, from which we can obtain a features' rank based on their importance given by VIM.
We employ RandomForestRegressor model from scikit-learn with 34978 samples (∼80%) for training set and 8745 (∼20%) samples for testing set. The RF model is trained by the training dataset with 138 (∼413/3) randomly selected features taken for each split, 200 numbers of bootstrapped training sets and maximum depth of 20 for each tree. The model is assessed within testing set and the prediction accuracy is quite high where the correlation of determination (R^2) reaches .67 between predicted salary and observed salary from testing set. Fig. 1 shows the prediction results and reveals the 10 most import features.



(a) Visualization of relationship between prediction result and observations.

(b) Variable importance measurement obtained by Random Forest method

Figure 1: Results from Random Forest Regression Methods

It demonstrates on the feature importance plot (Fig. 1b) that country type explains to salary far more than any other features. One possible explanation is that the salary level in developed countries is generally higher than that in developing countries, which results in two separate groups.
Therefore, we extract those observations which belongs to developed countries and combined them into a new dataframe developed_df, the rest of the observations are combined into another new dataframe developing_df.
Then we develop two sub-model to the aforementioned two dataframes using Lasso regression model, random forests regression method and multilayer perceptron.

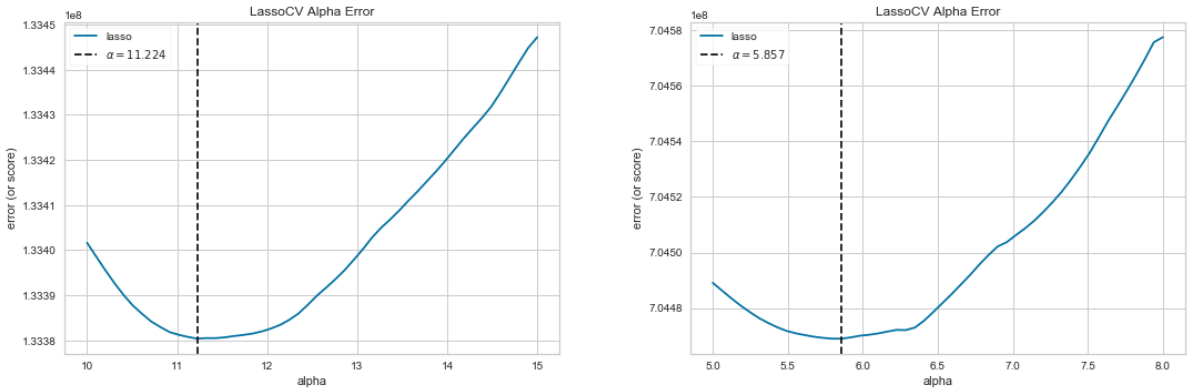
3.2 Partitioned Models

3.2.1 Lasso Regression Model

We employ the Lasso model because it possesses a good variables selection ability by shrinking the coefficients of less important features to 0, which improves the interpretability and prediction accuracy of the model, especially for high dimension dataset. In addition, comparing with RF and MLP, the lasso model allows us to check the relationship between each feature and response.

The Lasso models are developed separately for developing countries (denoted as M1) and developed countries (denoted as M2) with 10659 samples (∼80%) for training set and 2665 (∼20%) samples for testing set in M1; 24980 samples (∼80%) for training set and 6246 (∼20%) samples for testing set in M2.

To select the value of the tuning parameter λ that gives the best model under consideration, we use 5-th fold cross-validation to pick out the best one that minimize the cross-validation error out of various candidates, which yields 11.224 for M1 (Fig. 2a) and 5.857 for M2 (Fig. 2b).



(a) Cross-validation error that result from applying lasso regression to the developing countries data set with various values of λ.

(b) Cross-validation error that result from applying lasso regression to the developed countries data set with various values of λ.

Figure 2: Results of cross-validation error with various values of λ

Finally, the models are re-fit using the selected λ and observations from training set as shown in Fig. 3, with the testing set observations in x-axis and predicted salary in y-axis. Models are assessed within testing sets where R^2 reaches 0.494 in M1 (top right graph in Fig. 3) and 0.575 in M2 (bottom right graph in Fig. 3).

Noticeably, the tails behave badly in both M1 and M2 where each point should ideally scatter around the line y = x. The model diagnostic plots illustrate negatively skewed residual distributions for both countries (Fig. 4a for developing countries, Fig. 4b for developed countries) where M1 skewed heavier than M2. Furthermore, we observe that the two models perform non-linear trends (see Fig. 3).

Therefore, we apply non-linear transformation to our models by taking square root of the response variable and reconstruct the lasso models. The results of the adjusted lasso model are shown in Fig. 5 where the models' fit improve significantly.

The R^2 in testing set increases from 0.494 to 0.539 for M1 and from 0.575 to 0.606 for M2. In addition, the distributions of the residual in both models improve as well where they approximately exhibit a normal distribution (left graph in Fig. 6 for developing countries, right graph for developed countries).

Some of the interesting features selected by the adjusted lasso models will be chosen to elaborate in section 4.

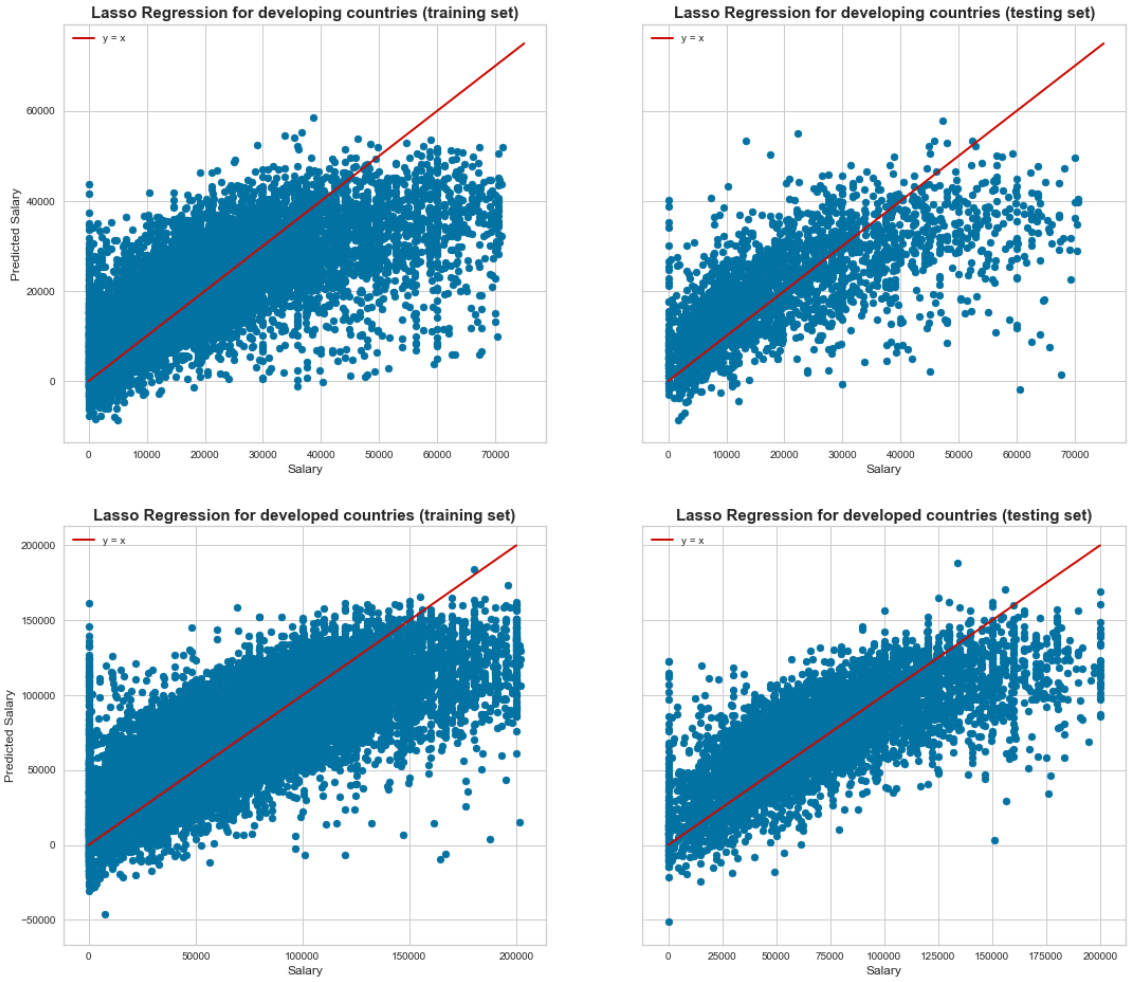


Figure 3: Lasso regression model for developing and developed countries

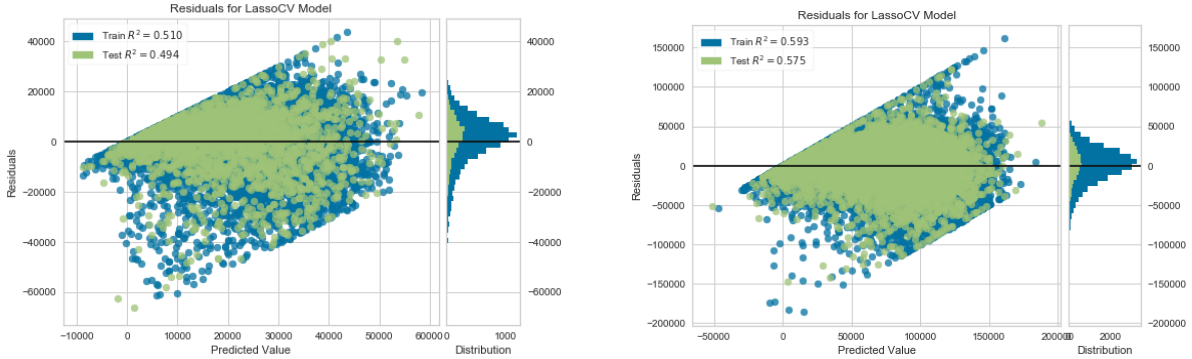


Figure 4: Model diagnostic plot

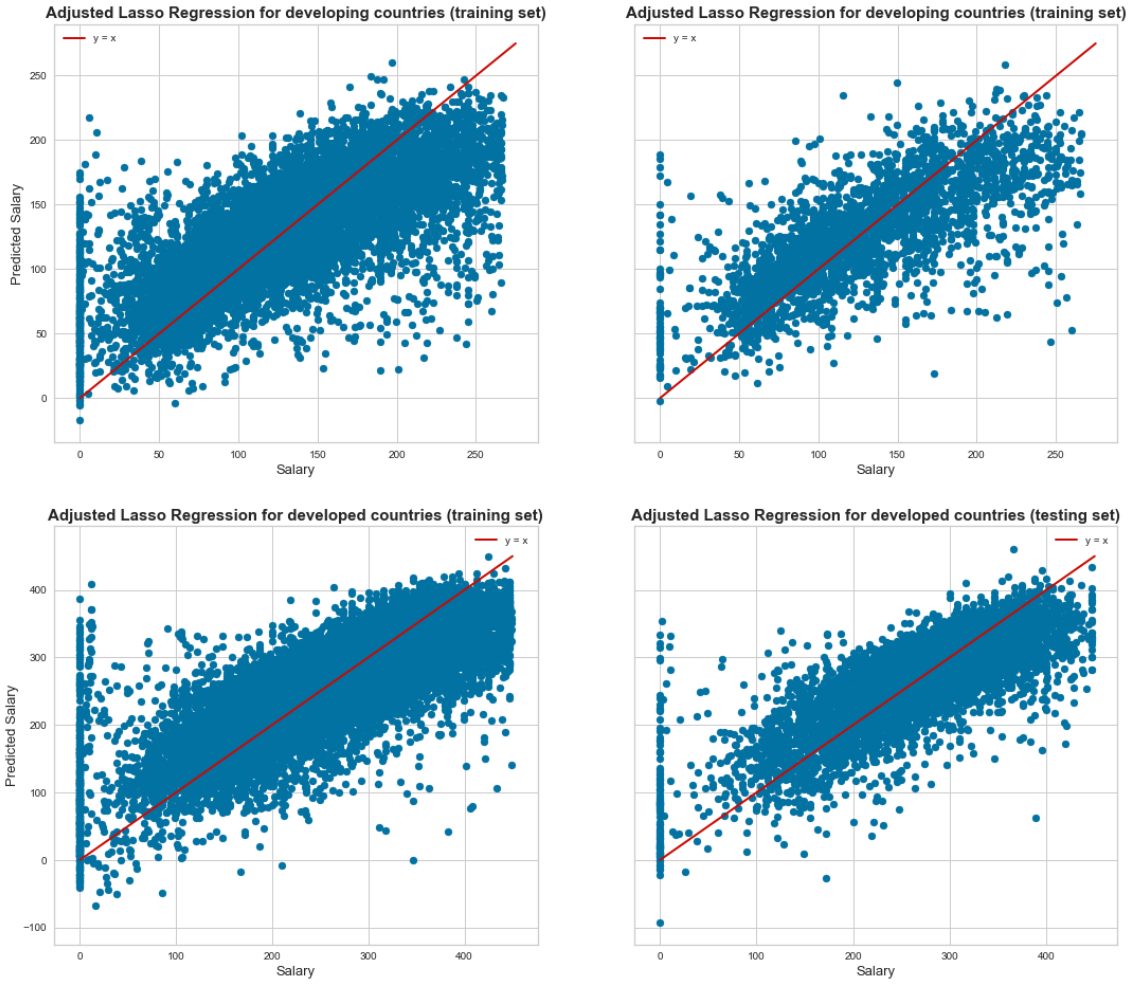


Figure 5: Adjusted Lasso Models

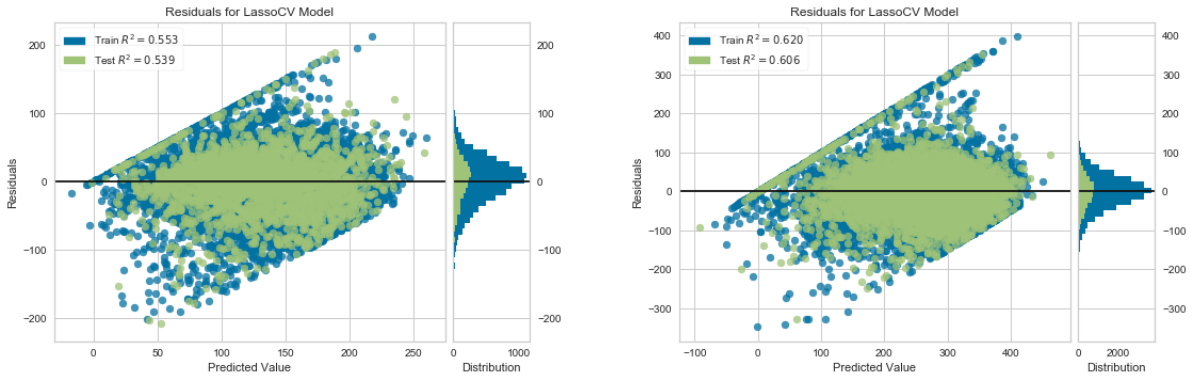


Figure 6: Model diagnostics for Adjusted Lasso Model

3.2.2 Random Forest Regression Model

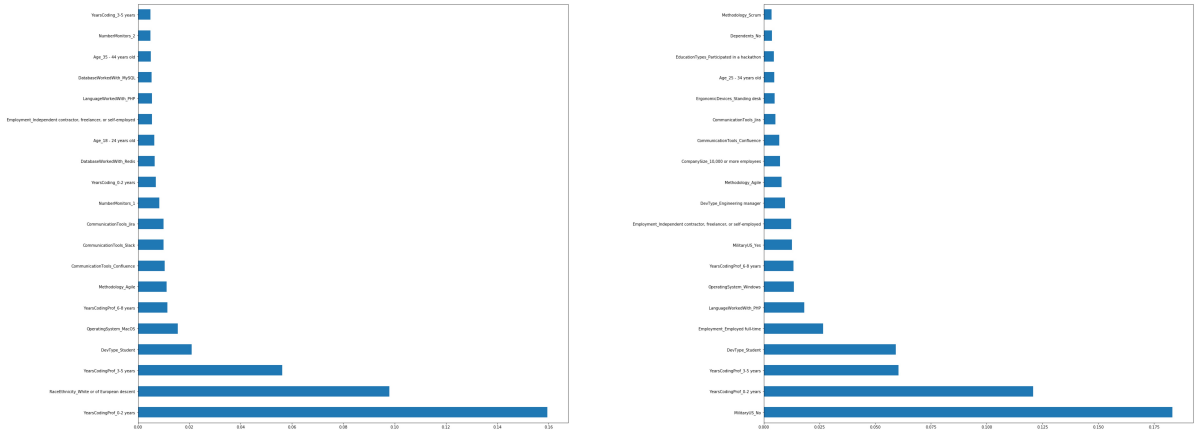
The second method we use is random forest regression model as stated in 3.1 This time, we reapply this model focusing on the differences between feature importance of developing and developed countries.

For developing countries, the RF model is trained by the training dataset with 118 ($\sim$p/3 for regression model) randomly selected features taken for each split, 150 numbers of bootstrapped training sets and maximum depth of 15 for each tree. The model is assessed within testing set and the prediction accuracy is quite high where the R^2 is 0.485 between predicted salary and observed salary from testing set (left graph in Fig. 7).

Similarly, for developed countries, 118 features are randomly selected for each split, 200 numbers of bootstrapped training sets and maximum depth of 20 for each tree are set. The R^2 is 0.568 between predicted salary and observed salary from testing set (right graph in Fig. 7).

Figure 7: Random Forest Regression model

The feature importance given by VIM will be discussed in section 4.



(a) Developing countries (b) Developed countries

Figure 8: Feature importance

3.2.3 Multilayer Perceptron Model

Finally, we train a multilayer perceptron regression model with 4 layers including 2 hidden layers with 400 and 10 nodes respectively. The R^2 is 0.504 for developing countries and 0.60 for developed countries. The results are shown in Fig. 9.

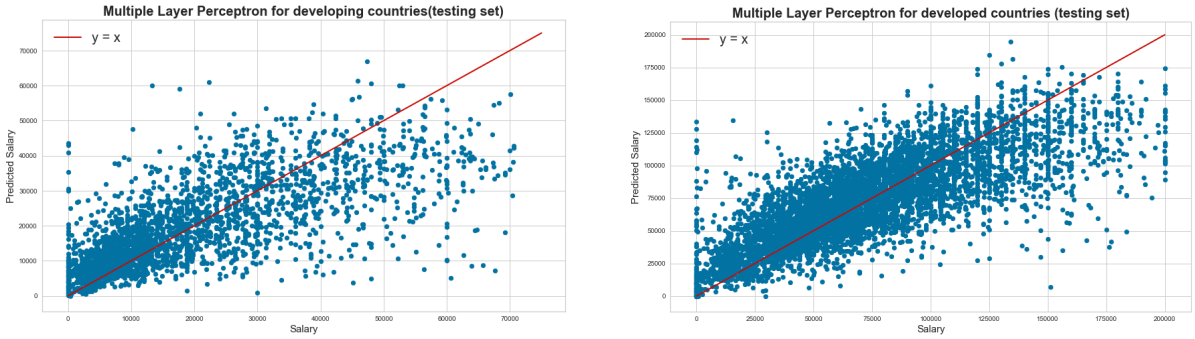


Figure 9: Multilayer Perceptron Regression Model

4 Results

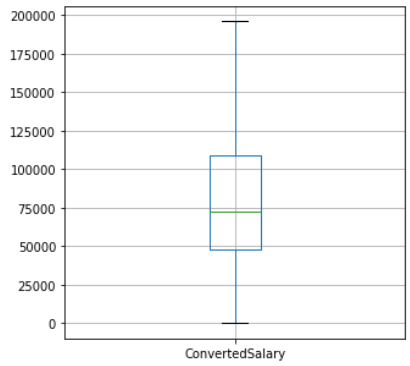
In the two models for developed and developing countries, we have two set of features that have the most significant impact on the programmers' salary. Most of them appear in both the models, but some of them are different.

One of the biggest difference between the two lists is that *MilitaryUS_No* and *MilitaryUS_Yes* only appears in developed countries. This is because if one indicates that he or she is a US soldier or is not a US soldier, then it is highly likely that he or she actually resides in the US. For the people who are not US citizens, they will not answer this question so the response will be NA. Since US is a developed country, this option mostly only appears in the data that came from developed countries.

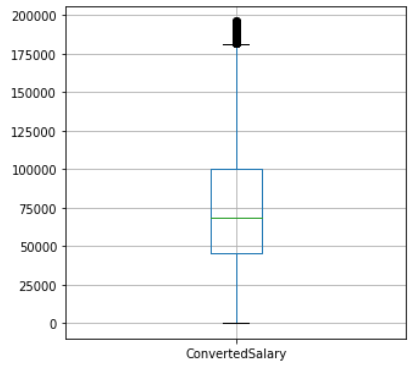
Another difference is that the options for *RaceEthnicity* only appears in developing countries. As we can see from the graph, for sast Asian and white people, their salary do not have much difference in developed countries. But in developing countries, white people or European descents have much higher salary. But we have to be careful to get the conclusion that racism is more prevailing in developing countries than in developed countries. Because the salary difference between each racial groups can also come from geological factors. Even restricted in developing countries only, there are still large gaps of economical development among different regions.

There are also many features in common, for example, *Age*, *YearsCoding* and *YearsCodingProf* (years coding as a professional) appear with high ranking among all features in both the lists. They are all related to how experienced a programmer is. It is expected that a programmer with more experience will earn more. And the salary will drop after retirement.

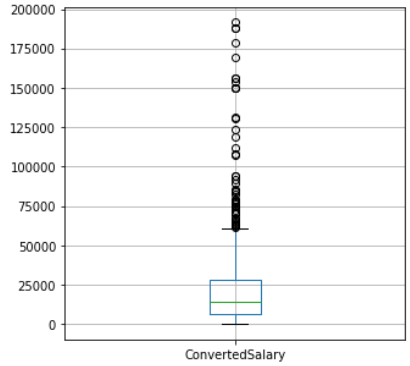
Company size is another expected important feature. People all want to go to big companies like BAT (Baidu, Alibaba, Tencent), and FLAG (Facebook, LinkedIn, Amazon, Google). Large company means



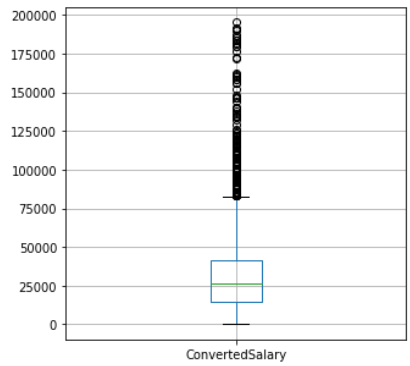
(a) East Asian in Developed Countries



(b) White or European Descent in Developed Countries



(c) East Asian in Developing Countries



(d) White or European Descent in Developed Countries

Figure 10: Race Ethnicity Difference

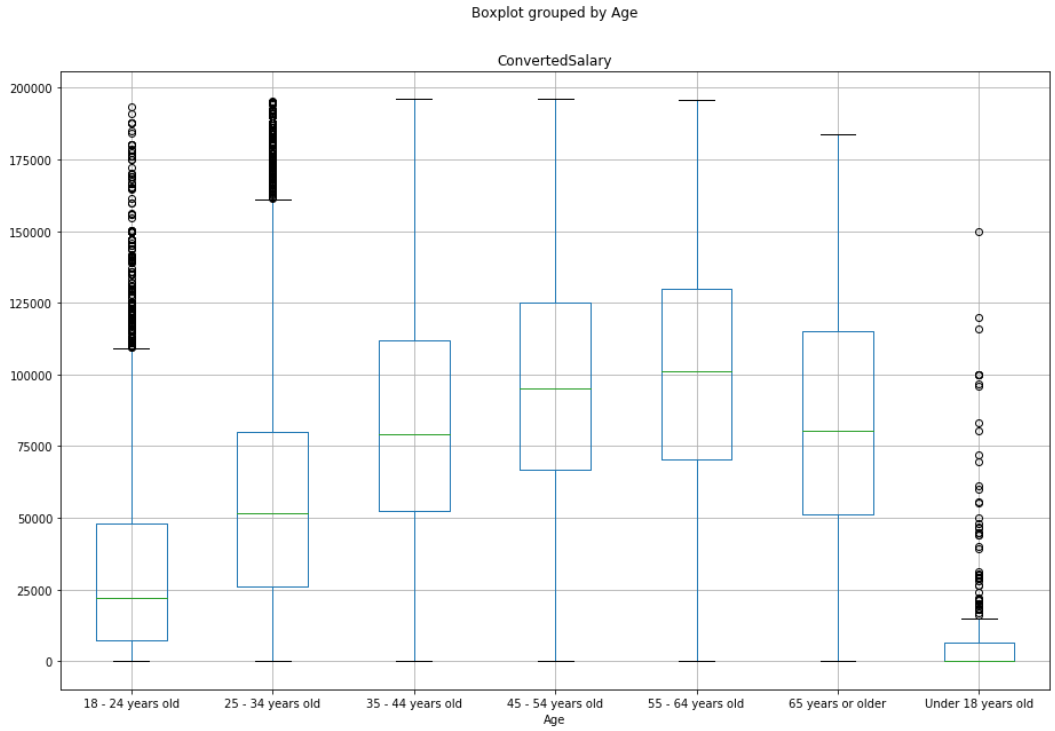


Figure 11: Salary grouped by Age

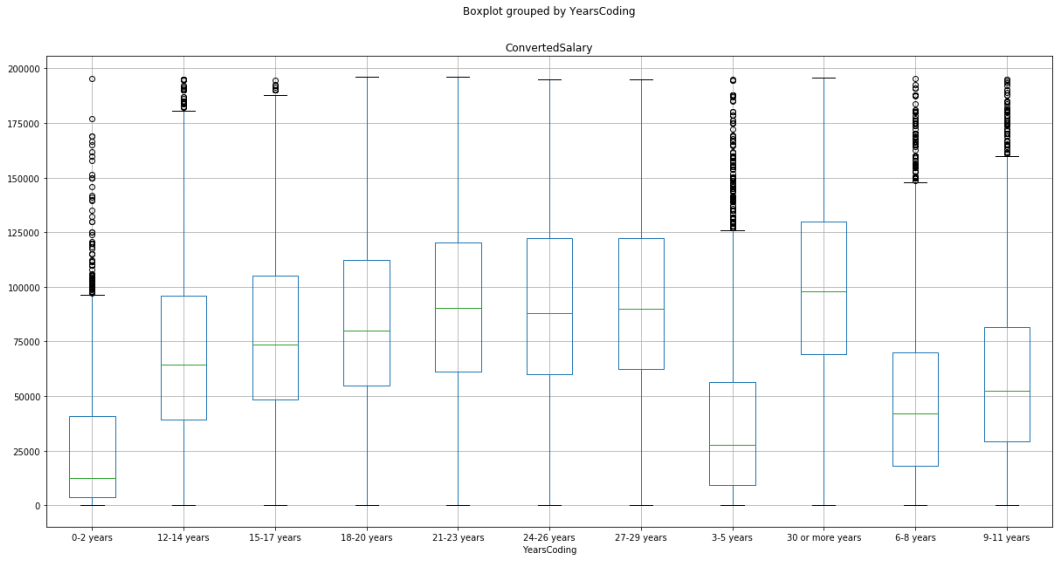


Figure 12: Salary grouped by Years Coding

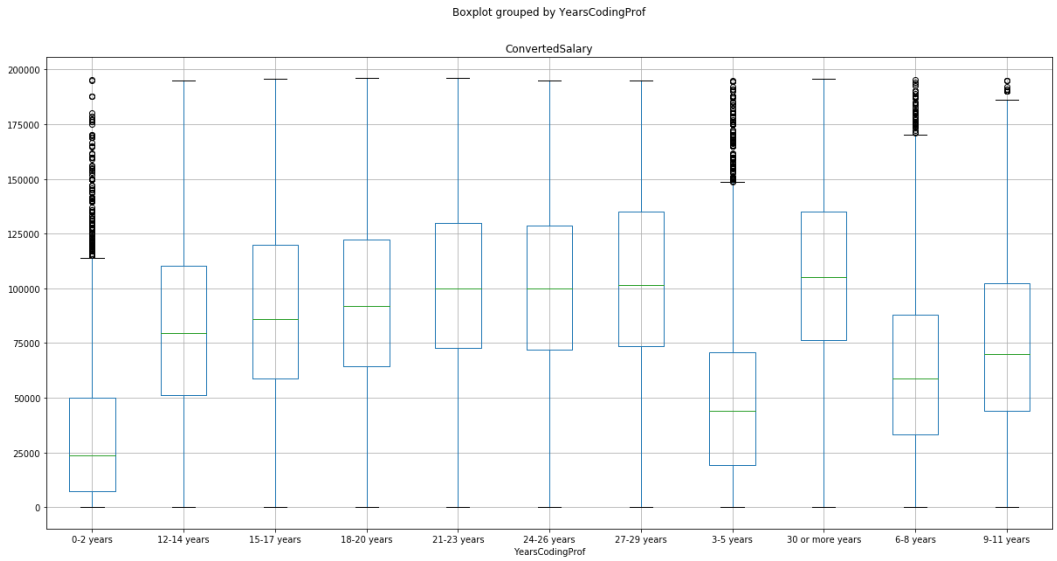


Figure 13: Salary grouped by Years Coding Professional

powerful and successful. These companies can pay higher salary to attract the most talented programmers.

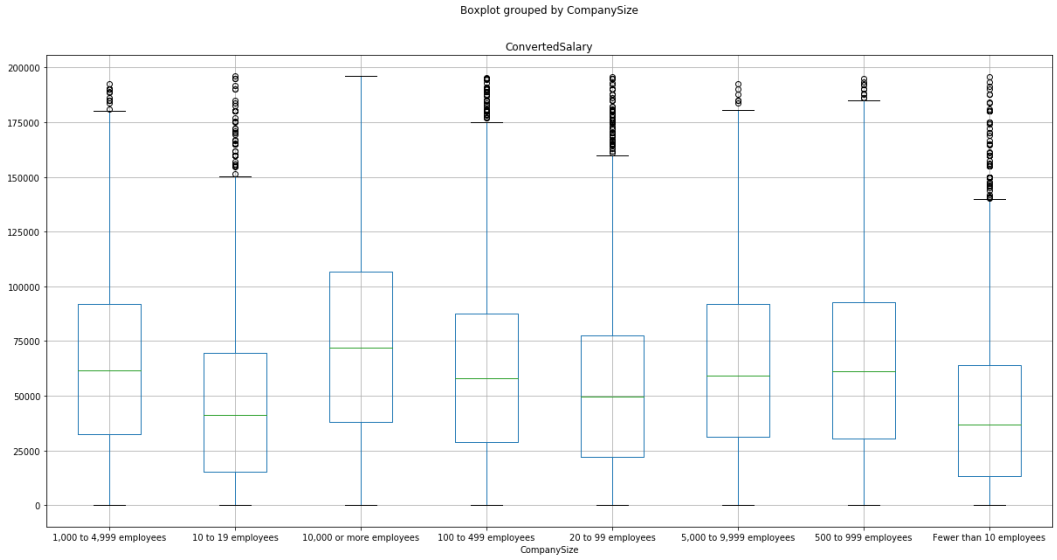


Figure 14: Salary grouped by Company Size

Employment type plays important roles too. This is natural and expected. Full-time workers get payed more than part-time workers and unemployed people earn the least.

Next, let's look at educational factors that are close related to us students. You can see that a PhD degree holder definetely earns more than others. Bachelors and masters do not have a big gap. So maybe pursuing a master's degree is not an economical idea. And also Education of parents has the same behaviour. It seems that the degree can not only influence one's own salary but also the children's salary.

It is surprising to find that arts is the major that earns the most. And people in humanities and social sciences majors have higher salary too. One possible explanation is that, the industry prefer people with

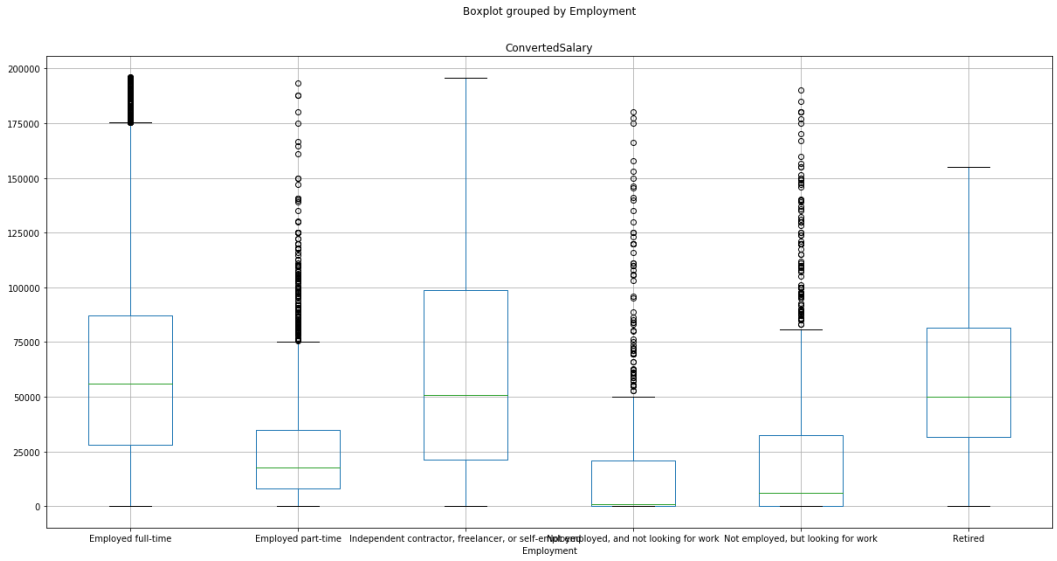


Figure 15: Salary grouped by Employment Type

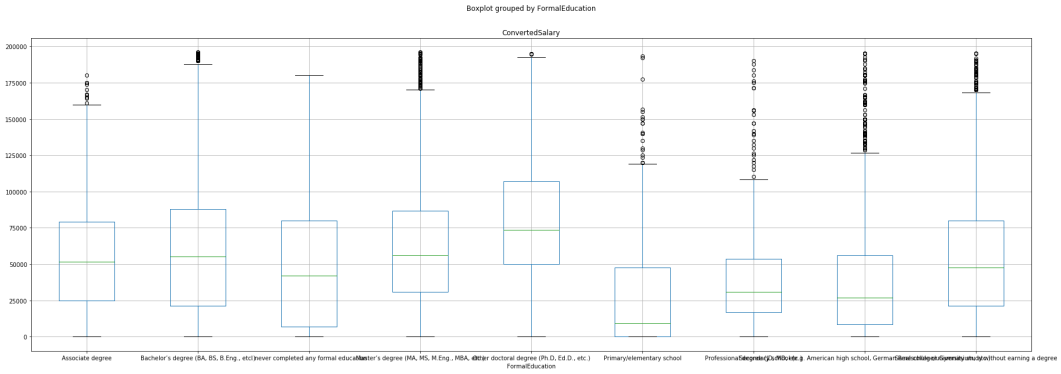


Figure 16: Salary grouped by Education

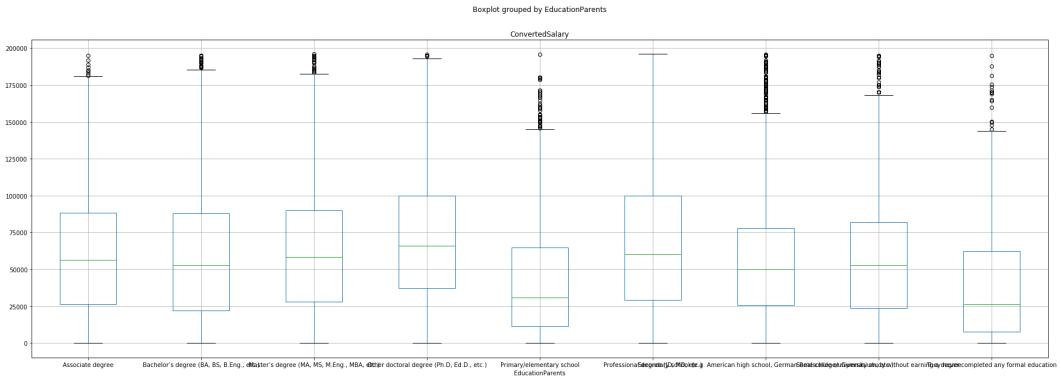


Figure 17: Salary grouped by Education of Parents

interdisplinary background. If a person has coding experience together with expertise in other areas, he or she will be more competitive.

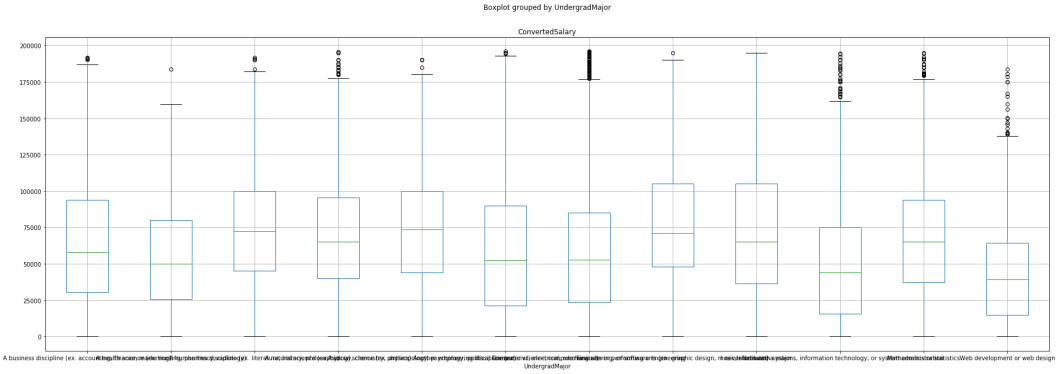


Figure 18: Salary grouped by Undergraduate Major

The important technical features in common are operating system, the methodology Agile, and language PHP. Let's look at them one by one.

In operating systems, the people who use MacOS earn the most while Windows, Linux and BSD/Unix do not have much difference. I think this is possibly due to that MacOS is nearly only used on Apple products, whose prices are higher than other consumer electronics. So only those who can afford mac computers will use MacOS, rather than that using MacOS makes them earn more.

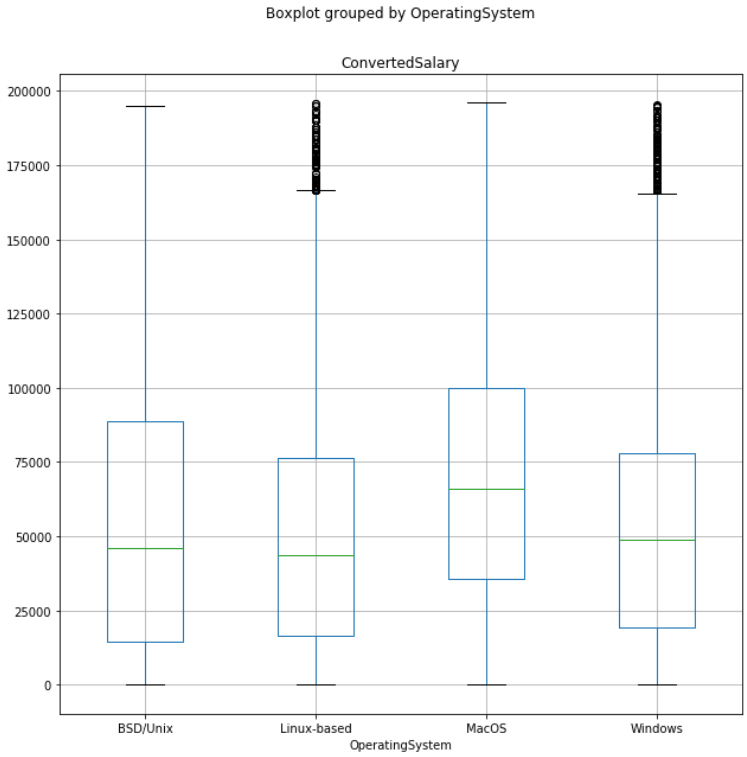
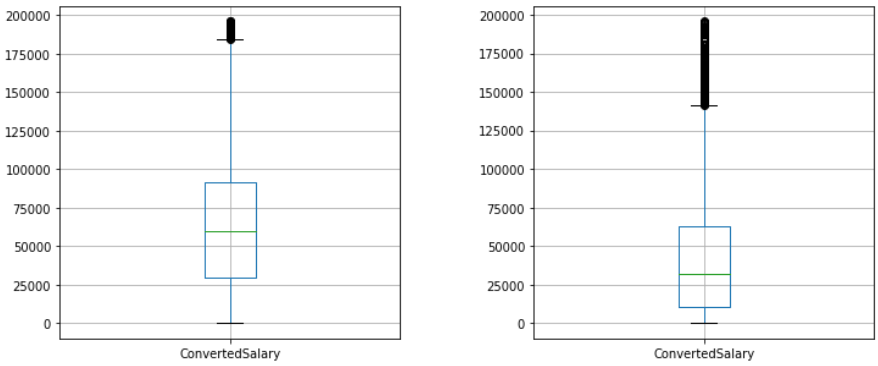


Figure 19: Salary grouped by Operating System

Agile is a very popular developing methodology in the industry. It basically means to develop products

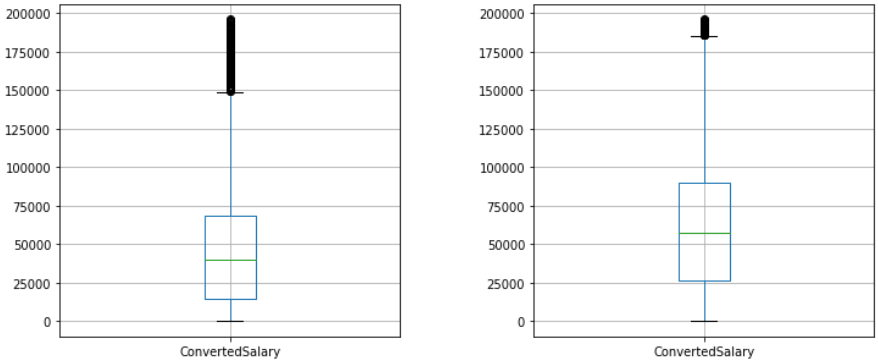
quickly. This method requires programmers' experience and can reflect the strength of a company. The companies that implement Agile method are also more efficient and are able to update their codes more frequently.



(a) Agile (b) Not use Agile

Figure 20: Methodology Agile

PHP is a language mostly used in front-end development. Because front-end developers have lower salary than other positions, it is not surprising to see that people working with PHP have lower salary than the people who don't.



(a) PHP (b) Not use PHP

Figure 21: Language PHP

5 Conclusion

In this project, we fitted several prediction models for the programmers' salary in developed and developing countries, using the dataset from StackOverflow user survey. We also make interpretations about our models and find what the important features are in the model and how they affect the salary prediction. The dataset can provide a practical guidelines on how to earn more money. Furthermore, it reflects what are the prosperous areas in the industry now.
But due to the limited time, there are still some problems need to be solved in the future. First, after dividing the datasets into developing_df and developed_df the R^2 in all models reduced. Second, there are

still many features could be grouped but we did't because we are lack of related knowledge. Third, the method for filling missing value could be improved.

Acknowledgement

This project is to fulfill a requirement of STAT3612 Data Mining course in the University of Hong Kong. We would like to thank Prof.ZHANG, Aijun for his guidance and advice to this project as well as elaborating different models for us.

Reference

Kaggle. (2018, May 15). Stack Overflow 2018 Developer Survey. Retrieved April 12, 2019, from https://www.kaggle.com/stackoverflow/stack-overflow-2018-developer-survey