Salary prediction of software developers

A system for predicting the salary for an aspiring developer

Surya M N
Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
suryamnnitk@gmail.com

Avanish V Patil
Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
avanishpatil23@gmail.com

Kedar U Shet
Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
kedarkedu01@gmail.com

Tushar Y S
Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
ystushar18@gmail.com

Abstract*— In the IT sector various features play a vital role in defining a good career as a developer. For developers seeking jobs, it would be beneficial if they had a model to predict the salary range and roles based on various features like country, years of coding, developer type, degree, formal education, skillset, and programming languages. Job recommendation systems and salary predictors have been popular in doing this job. Hence, in this paper, we will talk about the most commonly used models for predicting role and salary based on different features and skillset of a developer, building JRMs, the factors that tend to have an impact on job satisfaction, and trying to understand the most popular languages and platforms amongst the employees so that job seekers know what's trending.

Keywords—Salary prediction, Job Prediction, Job satisfaction, Job Recommender systems

I. INTRODUCTION

Salary prediction provides an important insight to an aspiring developer about what the most in-demand skills are, jobs that are high-paying. Also, for an organization looking to hire developers, a recommender system can provide a suggestion of the salary which can be offered to an individual given their set of skills, experience in coding, and various other factors. Thus a salary predictor serves a dual purpose.

For our project, we have chosen the dataset **"Stack Overflow 2018 Developer Survey"** from Kaggle which includes features of developers like country, years of coding, salary, formal education, gender, developer type, skill set, Undergrad Major, languages, frameworks and database worked with, etc. Using all these attributes we can get various insights like most popular languages, most popular frameworks, trending skill sets, and many more. But which one of these factors affects the salary a developer gets?

Why is the problem important to solve?

This is what we are interested in. Is it the years of coding experience, the country where the developer stays, his/her degree, or the major taken in undergrad? Keen observation would tell us that the combination of all these factors is what finally affects the job role and the salary as a developer.

II. PREVIOUS WORK

There has been a good amount of work in this space already. A lot of articles and research papers indicate the same. The salary prediction has been done on various jobs offered on online sites like Career Builder, Monster, etc. The most popular models for salary prediction are Linear Regression models(LM), Logistic Regression(LR), K-Nearest Neighbours (KNN), Multi-layer perceptrons(MLP), Support Vector Machines (SVM), Random Forest (RF), Adaptive Boosting (AB or AdaBoost) with Decision Trees along with the combination of the multiple models which helps to increase accuracy and other factors of the model. What is also usually tried is auto evaluation of the resume along with these models. Extensive feature selection models also have been deployed to check the influence of each feature on the salary and then wisely choose the final attributes.

Another interesting method for feature selection is the backward elimination method where initially all the attributes are considered and then taking out one feature at a time and checking whether or not the loss has increased significantly and finally taking a call on whether to keep a feature or not. This method is generally slow and has a few problems but tends to give decent results to the overall fit.

There has also been an attempt to run an experiment on 3 regression models namely Random Forest, Support Vector Machines, and Decision Trees to select the best Data Mining Models for salary prediction on the final preprocessed dataset. The outcome of this result was the Random Forest Regression model clearly outperforms the other 2 and also has added benefits like it is more powerful to numeric data and fits very well.It is a good choice for discrete data also.

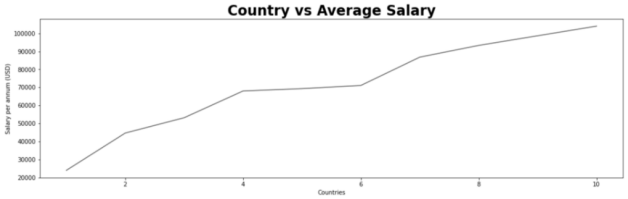
There was yet another interesting thing we found out about the already existing solution is that auto inferencing i.e., for example if someone knows how to program in “Scala” he must definitely know “Java” even if the user fails to mention “Java” explicitly in the resume/CV. This simply means that it is okay even if the subtle and most obvious information is missed out and the resume is mostly information that proves how the candidate stands apart from others and at the same time ensuring the resume/CV isn’t clumsy.

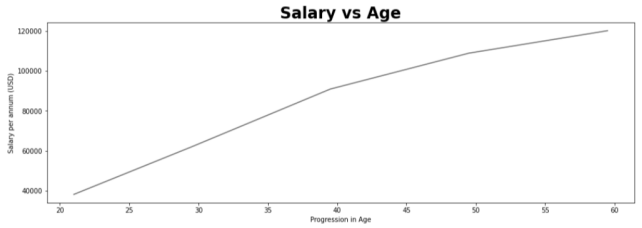
There were some drawbacks also that have been mentioned as a part of the articles and research papers we referred to and the most common was that attributes considered to predict the salary of an employee were very less and we would like to improve in this area by adding more important and relevant features for predicting the salary or job.

III. PROPOSED SOLUTION

The dataset is taken from Kaggle, the link to which can be found in the references section of this paper. We started off with Data Preprocessing. Here the dataset was visualized and we understood the gist of it. The null values in the dataset were cleaned up by deleting the rows. The outliers were also handled by deleting the rows. The data is mostly categorical so we had to convert the categorical data into numerical data (For example, replace the Age value like 18-22 years with 20 years which is the mean). Also, columns like languages known, Framework, Developer type had multiple attributes and we handled that by creating a column of all the unique attributes and 1.0 or 0.0 is assigned to the column depending on whether the language is learned by the person and similarly for the developer type attribute, basically One Hot Encoding. The categorical data like working_hours, frameworks, and salary_range were converted to discrete attributes by taking the average of the range.

In the Exploratory Data Analysis part we have done various visualizations of things like most popular languages, most popular frameworks, most popular IDEs, countries with the average salary they offer, salary distribution and to see how different attributes affect salary we have plotted graphs of salary vs hoursComputer, salary vs country, salary vs age, salary vs YearsCoding and we observed a linear relationship with salary.





The salary prediction was always tweaking the models, trial-and-error to get the best possible result. In the process, we learned more as to what not to do while predicting salary. When it comes to building models to train the data we tried to implement quite a few models. Linear regression, Multi Linear Regression, Ridge Regression, Decision Trees, AdaBoost, and Artificial Neural Network. After the EDA pre-processing the attributes that were finally chosen to train the model were:

1. **Age**
2. **Country**
3. **Gender**
4. **Years of Coding Experience**
5. **Hours Working on Computer Per Day**
6. **Programming Languages Known**
7. **Type of Developer.**

The process of creating a model is the most fascinating part where we tend to do trial and error mostly by tweaking the models a bit in order to increase the accuracy and reduce RMSE. In the course of finding the best fit model we tried out quite a few models that include Multiple Linear Regression, Ridge Regression, AdaBoost, Support Vector Machine Regression, Decision Tree Regression and Artificial Neural Network.

Models :

1) **Multiple Linear Regression (MLR)** is the first model we tried as it is used extensively in econometrics and financial interface. As mentioned above there is a linear relation between all the metrics and the salary. Hence, we thought it should be the first method we should start with.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

But the issue we faced, it was more sensitive to outliers and few features were not linear to expected salary. All the attributes were not independent of each other.

Accuracy : 58.75%

RMSE : 32138.38

2) **Support Vector Regression** is similar to linear regression where the straight line is the hyperplane. This model gave the results similar to our previous multiple linear regression model. SVR manages to fit the best line within a threshold of values(epsilon-intensive tube). Even though we tried to minimize the error using this model it was not that an improvement to the previous model to great extent. So we tried with other different models in aiming to minimize the RMSE

Accuracy : 59.19%

RMSE : 33774.45

3) **Ridge Regression** was the next model we used to train our dataset and to make a prediction on test data. This was used to predict the unknown variable (salary) keeping in mind that the predicting variables could have correlation among themselves. But as we went to check the root mean squared error of this model, it was almost equal to the multiple linear regression. We could now draw an inference that the predicting variables are not correlated. Age for instance had no correlation with the gender of a person. It was worth giving it a try. Nonetheless,it was still not better than multiple linear regression to a considerable extent. The search for the better model continues.

Accuracy : 58.66%

RMSE : 32138.16

4) **Decision Trees** was the next model. We now thought of using a weak learner in order to have low variance and high bias. Given the fact that our prediction deviated very much from the actual value(high standard deviation), a model which can produce low variance was very much necessary,that was the idea behind using decision trees for regression(as it was a weak learner). As we used this to make a prediction on our test data,it was able to predict a little better than multiple linear regression model and any other previously tried model for that matter. But then, a significant improvement of this prediction was very much necessary. Trying out some other models was the only way to achieve the same.

Accuracy : 55.82%

RMSE : 32138.38

5) **Adaboost** is one of the most popular ensemble techniques which involves sequential training of the data with weak learners which are then and then increasing the weights of the tuples which were wrongly predicted so that they can be predicted correctly in the next iteration. Here we used the AdaBoostRegressor module from sklearn.ensemble.

Listed below are a few important formulas in AdaBoost Model.

$$\text{Performance of Stump} = 1/2 \ln \left[\frac{1-TE}{TE} \right]$$

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

AdaBoost has a lot of prerequisites like having a quality dataset which means handling of null values, no outliers, etc.

Accuracy : 59.19%

RMSE : 33774.45

6) **Artificial Neural Network** was the final model we implemented. First we tried to implement a sklearn model named MLPRegressor (Multi Linear Perceptron Regressor) which trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters. As it did not give satisfactory results we shifted to customized ANN.

We decided to shift to customized ANN so that we could have more control over the model by setting the learning rate, number of epochs and other hyper parameters.We customized our ANN which had 4 hidden layers and an output neuron. First layer was the input layer with around 60 neurons in it. 4 Hidden layers has 64,32,32,16 neurons in them respectively. Relu is chosen as the activation function for all the layers.

$$\textbf{ReLU}$$
$$\max(0, x)$$

This was the best one we could come up with. This outperformed every other model we tried with as it had the least root mean squared error while predicting the salary for the validation data..

Though all of the models had done significantly well in predicting the salary, this seems to be the

best as the size of the training data is very high and neural networks tend to perform well on huge data as this one and moreover neural networks can learn the parameters itself very well after tuning the required parameters like the number of neurons in each layer, required activation function, number of epochs and certain other hyperparameters.

Accuracy : 61.016%

RMSE : 32054.62

IV. EVALUATION METRICS

Initially we thought of dividing the salary into various brackets of 25000 or so. If the Actual Salary bracket and the Predicted Salary bracket match then the prediction is considered as correct. But we realised a fault in this method. Suppose one of the salary brackets was 25000-50000 and the actual salary was 26000 whereas prediction ws 24000 although we missed the prediction by a close margin our metric would consider it as a wrong prediction as actual salary and predicted salary lie in different brackets. So in order to overcome this problem we now check if the predicted salary is within the range of +/- 20,000 and if it is, it is considered as a right prediction else it is wrong. One more metric that we have chosen is RMSE.

V. RESULTS AND CONCLUSIONS

After having compiled all the models there were interesting results we observed. Infact, there was more to learn as to what not to do while predicting salary than what to do while predicting salary. We got insights on how categorical variables can be used to predict the numerical target variable which is Salary which we found was hard initially. Methods

The artificial neural network was the best among all the models, it had the best accuracy of 61.4% and also least root mean squared error while predicting the salary. So we finally used the same ANN model to predict the salary of a new person with certain given attributes.

Predictions are only as good as the dataset. The prediction did not perform extremely well for the dataset although we got pretty good results for lots of records

Sample Predictions :

ActualSalary	PredictedSalary
73619.0	67583.437500
44460.0	64721.218750
44287.0	46444.457031
125000.0	99882.664062
55562.0	71900.609375
10175.0	19587.482422
65000.0	64104.632812
79552.0	78182.054688
140000.0	101407.773438
112716.0	27743.046875
159740.0	78532.421875
60000.0	94026.359375
110000.0	98357.929688
122100.0	29983.691406
80004.0	75597.828125
90000.0	102610.335938
57975.0	67874.734375
47105.0	30576.433594
52000.0	79765.843750
80000.0	92005.843750

VI. Contribution

1) Avanish V Patil — Adaboost, Preprocessing

2) Surya M N — ANN, Decision Tree

3) Kedar U Shet — Preprocessing, SVR

4) Tushar Y S — Ridge Regression, Multiple Linear Regression

VII. LINKS FOR OUR WORK

Github Organization Link:

https://github.com/data-analytics-217-525-096-545/Data-Analytics-Project

Kaggle Notebook Link:

https://www.kaggle.com/suryamn/da-hashcoders-545-096-525-217

VI. References

[1] *Job Recommender systems: A survey*
By: Juhi Dhameliya and Nikita Desai
Publisher: IEEE Xplore

[2] S. T. Al-Otaibi and M. Ykhlef, "A survey of job recommender systems," 26 July 2012. [Online]. Link:https://academicjournals.org/journal/IJPS/article-full-text-pdf/B19DCA416592.pdf.

[3] *Salary Prediction in the IT Job Market with Few High-Dimensional Samples: A Spanish Case Study*
Ignacio Martın, Andrea Mariello, Roberto Battiti , Jose Alberto Hernandez Departamento de Ingeniería´ıa Telemática, Universidad Carlos III de Madrid, ´ Avenida Universidad 30, Edificio Torres Quevedo, Leganés (Madrid), 28911, Spain.
Link: https://www.atlantis-press.com/article/25899235.pdf

[4] Salary Prediction in It Job Market
By: Navyashree M, Navyashree M K, Neetu M, Pooja G R, Arun Biradar

[5] Analysis of data from the survey with developers on Stack Overflow: A Case Study, by Yogesh Beehary, Manish Ganoo.

[6] What Makes Geeks Tick? A Study of Stack Overflow Careers, by Lei Xu, Tingting Nian, Luis Cabral**.**

[7] Multiple regression models and Artificial Neural Network (ANN) as prediction tools by J.Stangierski, D.Weiss, A.Kaczmerek.

[8] Artificial neural network versus multiple linear regression by T.Brey, A.Jarre - Teichmann, O.Borlich.