

B.E.(Information Technology)

Final Year Project Report

Salary Predictor



Presented by:

Sherly Sinha (001711001004)

Paridhi Agarwal (001711001035)

Under the guidance and supervision of,

Prof. Utpal Kumar Ray

Department of Information Technology

Faculty of Engineering and Technology

Jadavpur University

Kolkata, India

2020-21

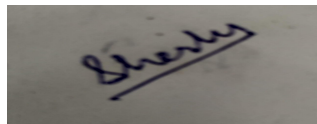
Declaration by the Students

We, Sherly Sinha & Paridhi Agarwal, hereby declare that this project report entitled "Salary Predictor" contains only the work completed by us as a part of the Bachelor of Engineering course, during the year 2020-21, under the supervision of Prof. Utpal Kumar Ray, Department of Information Technology, Jadavpur University.

All information, materials and methods that are not original to this work have been properly referenced and cited. All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

We also declare that no part of this project work has been submitted for the award of any other degree prior to this date.

Signature:



Signature:



Date: 01.06.2021

Date: 01.06.2021

Certificate

This is to certify that the project entitled "**Salary Predictor**" carried out by **Sherly Sinha (001711001004)** and **Paridhi Agarwal (001711001035)**, submitted to Jadavpur University during the year 2020-21 for the award of the degree of Bachelors of Engineering (Information Technology), is a bona fide record of work done by them under my supervision.

Signature of the supervisor:



Date : 04.06.2021

Acknowledgement

We are happy to present before you our project titled **"Salary Predictor"** . We would not have been successful without the help of certain people. Firstly, we are grateful to **Prof. Utpal Kr. Ray**, without whose guidance and supervision, this project would not have been possible. We would also like to thank our parents. Their affection and care was a constant source of inspiration for us. Lastly, We would like to thank our friends for their comments and constructive criticism that made us rectify the different loopholes in our project.

Contact:

Sherly Sinha: Phone: 7991136073

Email: sherlysinha07@gmail.com

Paridhi Agarwal: Phone: 9163592929

Email: theparidhi0@gmail.com

Abstract

Using the dataset[9] obtained from Kaggle, we performed exploratory data analysis to know more about developers around the world, salaries and salary gaps, community inclusiveness, companionship and gender distribution. We also created intuitive visualizations to understand the same analysis.

Then we experimented with various machine learning models to predict the salaries of developers around the world. We tried Linear Regression models and Gradient Boosting Machines. We discovered that the Light GBM version gave the best results.

Keywords: exploratory data analysis, Gradient Boosting Machines, Light GBM, Linear Regression

Contents

| | |
|--|----|
| 1) Introduction | 8 |
| 2) Basic Concepts & Technology Used | 9 |
| 3) Study of Similar Projects or Technology & Literature Review | 12 |
| 3.1) Salary as a Predictor for Salary : A 20 Year Study | 12 |
| 3.2) Gender and Salary Inequity | 12 |
| 3.3) The Multifactorial Achievement Scale as a Predictor of Salary | 13 |
| Growth | |
| 3.4) Faculty salary as a Predictor of student outgoing salaries from | 13 |
| MBA programs | |
| 3.5) Predicting Compensation for Job Seekers | 13 |
| 3.6) Salary Prediction in IT job market | 14 |
| 3.7) Salary Prediction using Regression | 15 |
| 3.8) Literature Review Summary | 16 |
| 4) Proposed Models & Tools | |
| 4.1) Proposed Models | 19 |

| | |
|-----------------------------------|----|
| 4.1.1) Model 1: Linear Regression | 19 |
| 4.1.2) Model 2: Gradient Boosting | 21 |
| 4.2) Tools | 24 |
| 5) Implementation and Results | 26 |
| 5.1) Implementation | 26 |
| 5.2) Results | 28 |
| 6) Conclusions | 32 |
| 6.1) Conclusion | 32 |
| 6.2) Further Steps | 33 |
| Reference | 34 |
| Appendix | 36 |

Chapter 1

Introduction

In this project, we aimed to understand the salary distribution, community inclusiveness, companionship and gender distribution of software developers around the world. We also tried to create a machine learning model that could predict salaries of software developers.

Budding software developers around the world struggle to understand their true market value. Traditional education system trains students for the skills required to begin their careers. However, it often fails to educate them about the job market. This project aims to solve this problem for millions of software developers around the world and help them make informed decisions before choosing their careers.

We used various Python libraries like Pandas, NumPy, Matplotlib, Plotly and Seaborn. We also used Jupyter Notebooks to create our project. For prediction, we tried Linear Regression models and Gradient Boosting Machines. We discovered that the Light GBM version gave the best results.

Chapter 2

Basic Concepts & Technologies Used

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. We used Python version 3.9.0.

First, we performed data cleaning to remove inconsistencies within the data and handle the missing values.

Then, we performed data exploration to form hypotheses about our defined problem. We also visually analysed the data using Matplotlib, Plotly and Seaborn libraries.

Matplotlib[1]:

The matplotlib Python library, developed by John Hunter and many other contributors, is used to create high-quality graphs, charts, and figures. The library is extensive and capable of changing very minute details of a figure.

Plotly[2]:

The Plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

Seaborn[3]:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

We performed feature engineering to select important features and construct more meaningful ones using the raw data. We used Pandas and NumPy libraries for this purpose.

Pandas[4]: pandas is a popular Python-based data analysis toolkit which can be imported using `import pandas as pd`. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a NumPy matrix array. This makes pandas a trusted ally in data science and machine learning.

NumPy[5]: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Then, we did predictive modeling by training machine learning models evaluating their performance and using them to make predictions. We used linear regression models and gradient boosting machines.

Linear Regression[8]: It is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).

Gradient boosting[7]: It is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

We used the Scikit-Learn library for creating these models.

Scikit-Learn[6]: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

Chapter 3

Study of Similar Projects or Technology &

Literature Review

In this section, we present some existing projects or systems similar to our current project. We go through a complete thought process of the project subject and its viability.

3.1) Salary as a predictor of salary: A 20-year study [13]

This study investigated the value of salary after a few yr. in an organization as a predictor of, and therefore as an intermediate criterion for, salary at a later date. Salary data were collected for each yr. of the 20-yr. careers of 52 aircraft engineers. The salaries were combined to yield yearly distributions, 1 for beginning salary and 1 for each yr. of experience.

The resulting distributions were intercorrelated and the following results obtained: (a) 92% of the intercorrelations were significant at the .01 level, (b) correlations between equidistant yr. became larger as tenure increased, and (c) the variance of salaries increased with increasing tenure.

It is concluded that salary early in a man's career can be used as an intermediate criterion.

3.2) Gender and Salary Inequity: Statistical Interaction Effects [14]

The gender-related inequity in the salaries of social workers has been reported in numerous studies. To specify further the nature of that inequity, the author included

gender-related interaction factors in an assessment of the relationship of salary and its predictor variables in a sample of 827 social workers in Pennsylvania.

3.3) The Multifactorial Achievement Scale as a Predictor of Salary Growth. [15]

Sixty-nine middle managers employed by an Australian manufacturing company completed the 49-item Multifactorial Achievement Scale (MAS) designed to assess six discrete aspects of need for achievement, and indicated the size of their salary increases since joining the firm. Superior ratings of their work motivation were also obtained. A principal components factor analysis yielded a single factor that accounted for 61.4 percent of the variance, rather than the six factors hypothesized to underlie the MAS. However, significant correlations were obtained between most of the subscales of the MAS an index of salary growth and the ratings of work motivation. These findings are interpreted as offering support to the MAS as an alternative scale for assessing need for achievement among managerial samples.

3.4) Faculty salary as a Predictor of Student Outgoing Salaries from MBA Programs. [16]

The authors' purpose was to investigate the predictive value of faculty salaries on outgoing salaries of master of business administration (MBA) students when controlling for other student and program variables. Data were collected on 976 MBA programs using Barron's *Guide to Graduate Business Schools* over the years 1988–2005 and the Princeton Review's *The Best 295 Business Schools* 2014 edition.

3.5) Predicting compensation for job seekers[17]

This paper uses a large set of salary data from Glassdoor, a well-known site that collects information from its users about their employment experiences in terms of job satisfaction, compensation, and more. Since their desired output was a real-number value, they focused on exploring a variety of regression techniques to build their predictions.

They tried 2 machine learning models:

- Simple regression model
- Tree-based regression model (Decision trees and Random forests)

Interesting conclusions:

- Training error is only slightly smaller than test error in most cases, suggesting (unsurprisingly) that their linear regression model has relatively high bias, but also that overfitting is not a major problem.
- Regularization with small values of λ delivers some small gains in predictive accuracy, but no major boosts.
- Predicting the log of salary values is substantially more accurate than predicting the original value
- Salaries in the real world have a lot of variation that can't be fully accounted for by only the features in this dataset.

3.6) Salary Prediction in the IT job market - a Spanish case study[18]

This paper formulated the salary prediction problem as a classification task in order to have better accuracy by focusing on discrete ranges instead of a continuous salary value. They

propose manual feature preprocessing to clean, format and standardize the collected data and reduce dimensionality by 10 times while improving the prediction accuracy.

They focus on the data of the Spanish IT job market.

Models used:

- Support Vector Machines
- Multi Layer Perceptron
- Random Forests
- AdaBoost
- Ensembles of above

Interesting conclusion:

- Ensembles based on the above models behave better and lead to a best case accuracy of ~84%.
- Some features, such as experience, job stability or certain job roles (i.e, Team Lead and IT Architect) contribute significantly to the final salary.

3.7) Salary Prediction using Regression Techniques[19]

Took salary data from a dataset. Then, plotted points corresponding to the points in the dataset in a 2-dimensional space. They used linear regression to draw lines between the plots and polynomial regression to draw curves. Used this to predict salary.

Models used:

- Linear regression

Interesting conclusion:

- They used only 2 dimensions to predict salary - which is clearly insufficient for such a complex prediction task.
- Model can be improved by using a larger dataset.
- Model can be improved by using more data points.

3.8) Literature Review Summary

| S.No | Paper title | Technologies | Conclusion | Issues | Author | Year | Publisher |
|------|---|--|--|---|-----------------------------------|------|--|
| 1. | Salary after a few years in an organization as a predictor for salary at a later date. (A 20 Year Study) | The salaries were combined to yield yearly distributions, 1 for beginning salary and 1 for each yr. of experience. The resulting distributions were intercorrelated. | Correlations between equidistant yr. became larger as tenure increased, and the variance of salaries increased with increasing tenure. | Limited sample size that over optimized on an unrelated field | Brenner, M. H., & Lockwood, H. C. | 1965 | <i>Journal of Applied Psychology</i> , 49(4), 295-298 |
| 2. | Gender-related inequity in the salaries of social workers | Simple statistical model that compares the salaries between the 2 genders | There was a huge gap between the salaries for men and women. | Outdated data that focuses on a single city | Hide Yamatani | 1982 | <i>Social Work Research and Abstracts</i> , Volume 18, Issue 4, Winter 1982, Pages 24-27 |

| | | | | | | | |
|----|--|---|--|--|--|------|--|
| 3. | The Multifactorial Achievement Scale as a Predictor of Salary Growth. | Multivariate linear regression. | Significant correlations were obtained between most of the subscales, an index of salary growth and work motivation. | A single factor that accounted for 61.4 percent of the variance, rather than all factors. | Orphen, Christopher | 1995 | Social Behavior and Personality: an international journal, Volume 23, Number 2, 1995, pp. 159-162(4) |
| 4. | Faculty salary as a Predictor of Student Outgoing Salaries from MBA Programs | Hierarchical Linear Regression Analysis. (Program characteristics as control variable; Faculty salary as the predictor variable; Avg outgoing salary as the dependent variable) | Higher faculty salaries were associated with higher starting salaries for MBA students upon graduation. | Correlation might not signify causation | Karla H. Hamlen, William A. Hamlen | 2015 | Journal of Education for Business, Volume 91, 2015 -Issue 1 |
| 5. | Predicting compensation for job seekers. | Simple regression model , Tree based regression model | Predicting the log of salary values is substantially more accurate than | Regularization with small values of λ delivers small gains in accuracy, no major boosts. | Megan Fazio , Jen Kilpatrick, Darren Baker | 2016 | CS 229 Project Final Report Autumn 2016 |

| | | | | | | | |
|----|---|---|--|---|---|------|--|
| | | | predicting the original value. | | | | |
| 6. | Salary Prediction in the IT job market - a Spanish case study | Classification Models (SVM, Multi Layer Perceptron, Random Forests, AdaBoost) | Ensembles based on the above models behave better and lead to a best case accuracy of ~84%. | Multi Layer Perceptron performs poorly due to limited data in the sample. | Nacho Martin, Andrea Mariello, Robert Battiti | 2018 | International Journal of Computational Intelligence Systems 11(1):1192 |
| 7. | Salary of a person after a certain year. | Linear Regression | Used only 2 dimensions to predict salary - which is clearly insufficient for such a complex prediction task. | Need a larger dataset and more data points. | Rupashri Barik | 2020 | SSRN Electronic Journal |

Chapter 4

Proposed Models and Tools

In this section, we take a look at the proposed models to predict the salaries and the tools that we used to implement the models. We implemented Linear Regression and Gradient Boosting Machines and compared the results.

4.1 Proposed Models

4.1.1. Model 1: Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. A scatterplot[10] can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e. The scatter plot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient[11], which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

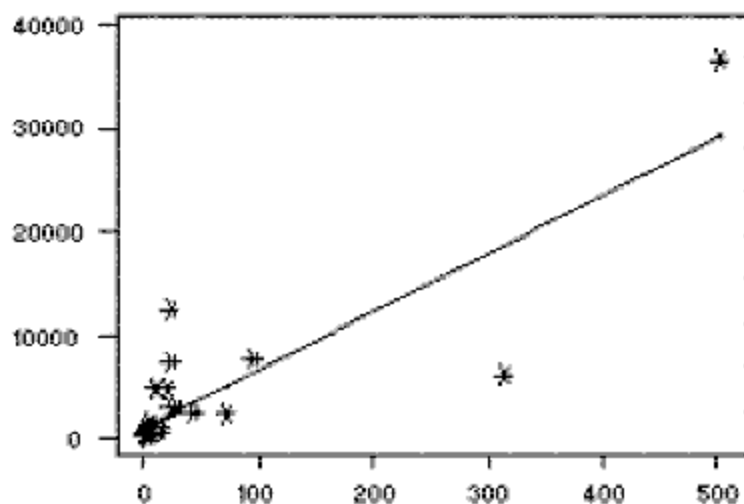
A linear regression line has an equation of the form $\mathbf{y} = \mathbf{a} + \mathbf{bx}$, where \mathbf{x} is the explanatory variable and \mathbf{y} is the dependent variable. The slope of the line is \mathbf{b} , and \mathbf{a} is the intercept (the value of \mathbf{y} when $\mathbf{x} = 0$).

Least-Squares Regression

The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values.

Example

The dataset "Televisions, Physicians, and Life Expectancy" contains, among other variables, the number of people per television set and the number of people per physician for 40 countries. Since both variables probably reflect the level of wealth in each country, it is reasonable to assume that there is some positive association between them. After removing 8 countries with missing values from the dataset, the remaining 32 countries have a correlation coefficient of 0.852 for number of people per television set and number of people per physician. The r^2 value is 0.726 (the square of the correlation coefficient), indicating that 72.6% of the variation in one variable may be explained by the other. Suppose we choose to consider the number of people per television set as the explanatory variable, and number of people per physician as the dependent variable.



To view the fit of the model to the observed data, one may plot the computed regression line over the actual data points to evaluate the results. For this example, the plot appears to the right, with number of individuals per television set (the explanatory variable) on the x-axis and number of individuals per physician (the dependent variable) on the y-axis. While most of the data points are clustered towards the lower left corner of the plot (indicating relatively few individuals per television set and per physician), there are a few points which lie far away from the main cluster of the data. These points are known as *outliers*, and depending on their location may have a major impact on the regression line (see below).

4.1.2. Model 2: Gradient Boosting

Gradient boosting involves three elements:

A loss function to be optimized.

A weak learner to make predictions.

An additive model to add weak learners to minimize the loss function.

Element 1. Loss Function

The loss function used depends on the type of problem being solved.

It must be differentiable, but many standard loss functions are supported and you can define your own.

For example, regression may use a squared error and classification may use logarithmic loss.

A benefit of the gradient boosting framework is that a new boosting algorithm does not have to be derived for each loss function that may want to be used, instead, it is a generic enough framework that any differentiable loss function can be used.

Element 2. Weak Learner

Decision trees are used as the weak learner in gradient boosting.

Specifically regression trees are used that output real values for splits and whose output can be added together, allowing subsequent models outputs to be added and “correct” the residuals in the predictions.

Trees are constructed in a greedy manner, choosing the best split points based on purity scores like Gini or to minimize the loss.

Initially, such as in the case of AdaBoost, very short decision trees were used that only had a single split, called a decision stump. Larger trees can be used generally with 4-to-8 levels.

It is common to constrain the weak learners in specific ways, such as a maximum number of layers, nodes, splits or leaf nodes.

This is to ensure that the learners remain weak, but can still be constructed in a greedy manner.

Element 3. Additive Model

Trees are added one at a time, and existing trees in the model are not changed.

A gradient descent procedure is used to minimize the loss when adding trees.

Traditionally, gradient descent is used to minimize a set of parameters, such as the coefficients in a regression equation or weights in a neural network. After calculating error or loss, the weights are updated to minimize that error.

Instead of parameters, we have weak learner sub-models or more specifically decision trees. After calculating the loss, to perform the gradient descent procedure, we must add a tree to the model that reduces the loss (i.e. follow the gradient). We do this by parameterizing the tree, then modify the parameters of the tree and move in the right direction by (reducing the residual loss.

Generally this approach is called functional gradient descent or gradient descent with functions.

One way to produce a weighted combination of classifiers which optimizes [the cost] is by gradient descent in function space.[12]

The output for the new tree is then added to the output of the existing sequence of trees in an effort to correct or improve the final output of the model.

A fixed number of trees are added or training stops once loss reaches an acceptable level or no longer improves on an external validation dataset.

4.2 Tools

First, we performed data cleaning to remove inconsistencies within the data and handle the missing values.

Then, we performed data exploration to form hypotheses about our defined problem. We also visually analysed the data using Matplotlib, Plotly and Seaborn libraries.

Matplotlib[1]:

The matplotlib Python library, developed by John Hunter and many other contributors, is used to create high-quality graphs, charts, and figures. The library is extensive and capable of changing very minute details of a figure.

Plotly[2]:

The Plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

Seaborn[3]:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

We performed feature engineering to select important features and construct more meaningful ones using the raw data. We used Pandas and NumPy libraries for this purpose.

Pandas[4]: pandas is a popular Python-based data analysis toolkit which can be imported using `import pandas as pd`. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a

NumPy matrix array. This makes pandas a trusted ally in data science and machine learning.

NumPy[5]: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Then, we did predictive modeling by training machine learning models evaluating their performance and using them to make predictions. We used linear regression models and gradient boosting machines.

Scikit-Learn[6]: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

Chapter 5

Implementation and Results

In this section, we take a look at the implementation and results achieved. We conducted exploratory data visualizations to understand the data better.

5.1 Implementation

Dataset

We used the StackOverflow 2019 Developers survey as the dataset. It contains:

- Responses of 100,000 developers who took the 30-minute survey released by StackOverflow
- Information about their professional coding habits, tools used and salary.
- 129 attributes

Data Cleaning

First, we performed data cleaning to remove inconsistencies within the data and handle the missing values.

Exploratory Data Analysis

Then, we performed data exploration to form hypotheses about our defined problem. We also visually analysed the data using Matplotlib, Plotly and Seaborn libraries.

Exploring Job Count:

We explored the different job titles that are present in the dataset and their count.

```

# Split the jobs and count them
df_jobs = pd.DataFrame.from_records(df['DevType'].dropna().apply(lambda x: x.split(';')).values.tolist()).stack().reset_index(drop=True).value_counts()

# Create plot
df_jobs.plot(kind='barh', figsize=(10,7.5))
plt.title('Stack Overflow Survey Job-Count')
plt.xlabel('Job-Count')
plt.ylabel('Job')
plt.grid()
plt.show()

```

Exploring Salaries:

We explored the salary distribution of responders and created a histogram.

```

# Create histogram
df['ConvertedSalary'].hist(bins=100, ax=axarr[0])
axarr[0].set_title('Salary Histogram')
axarr[0].set_xlabel('Salary')
axarr[0].set_ylabel('Count')

```

Exploring Countries:

We explored the countries that the developers belong to.

```

# Top n countries
n = 20

# Empty values
print('Empty Values:\t{}'.format(df['YearsCoding'].isna().sum()))

# Create plot
df_country = df['Country'].value_counts().head(n)
df_country.plot(kind='barh', figsize=(10,7.5))
plt.title('Count For The Top {} Countries'.format(n))
plt.xlabel('Count')
plt.ylabel('Country')
plt.grid()
plt.show()

```

Exploring Gender

We also explored the gender of the developers.

```
:  
# Empty values  
print('Empty Values:\t{}'.format(df['Gender'].isna().sum()))  
  
# Create plot  
df['Gender'].value_counts().plot(kind='barh', figsize=(10,7.5))  
plt.title('Gender Count')  
plt.xlabel('Count')  
plt.ylabel('Gender')  
plt.show()
```

Creating predictive models

We divided the dataset into 2:

- Train set: used for training the models
- Test set: used for testing the accuracy of our trained models

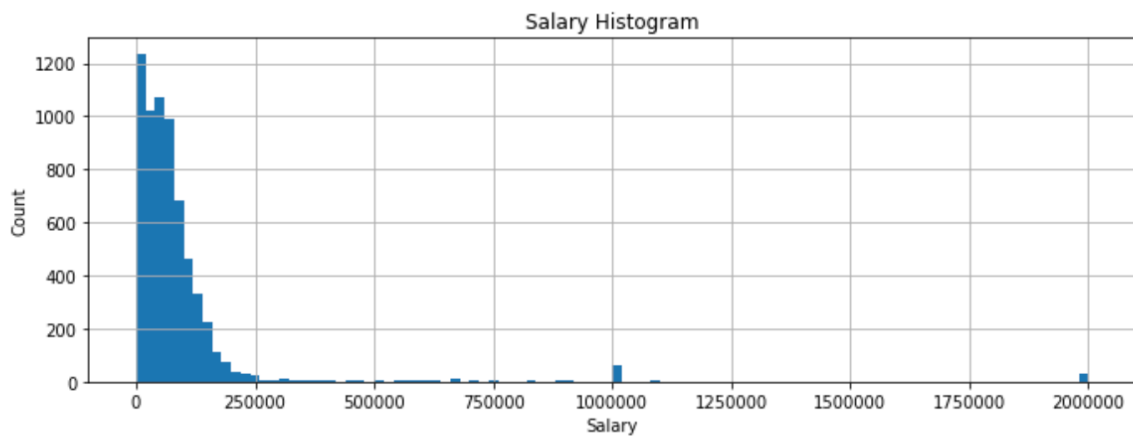
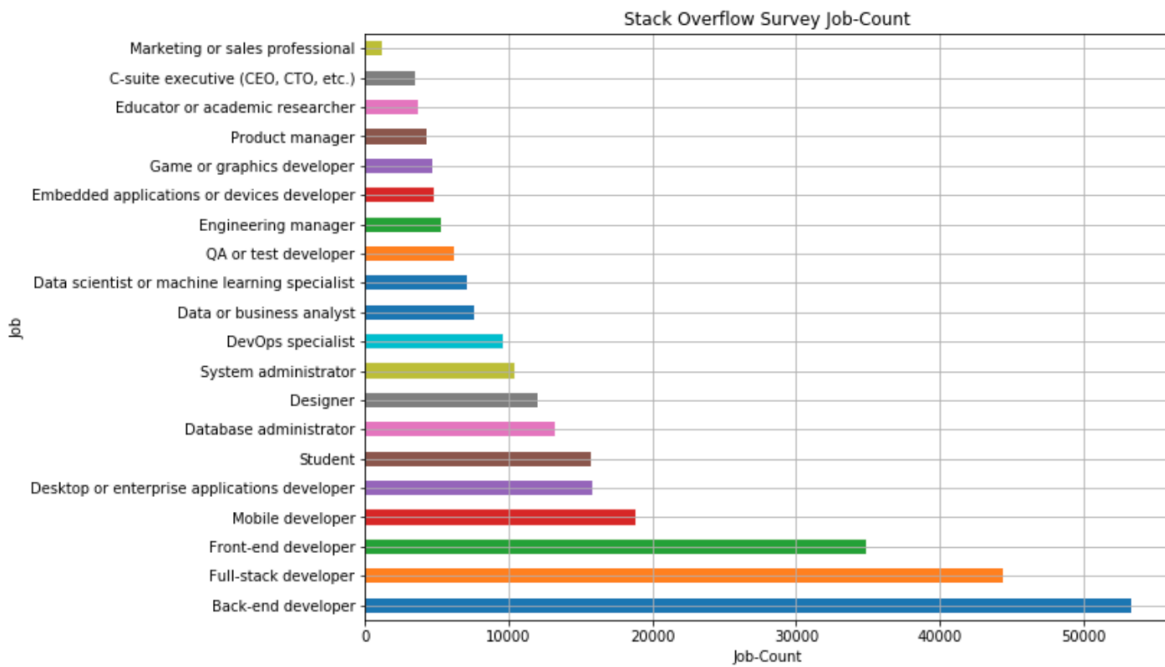
Using this dataset, we trained 2 regression models:

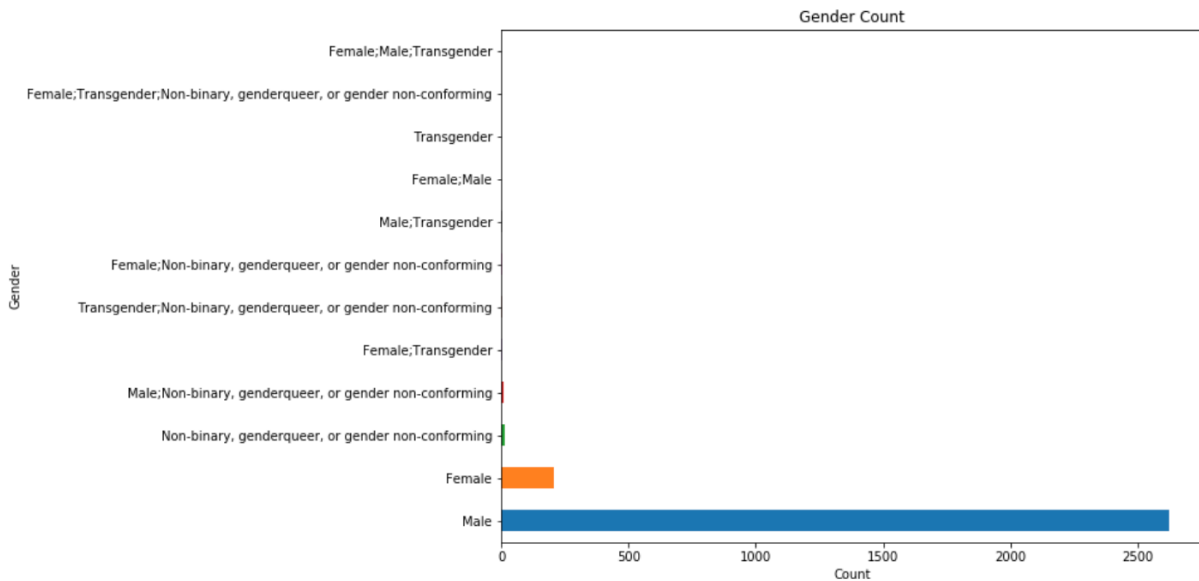
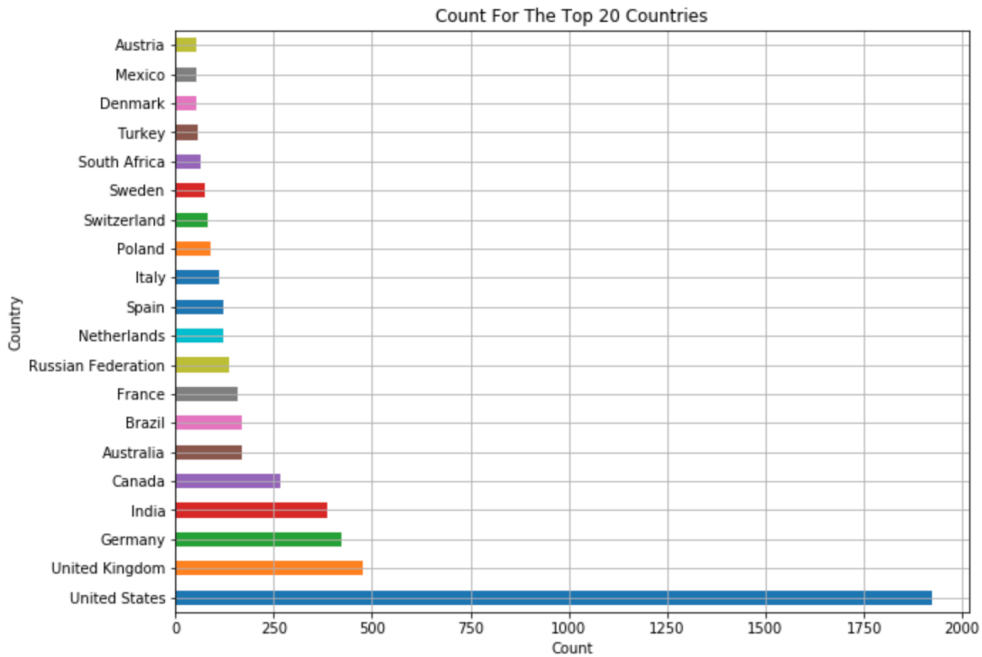
1. Linear regression
2. Gradient Boosting regression

We trained them on the features to predict salaries.

5.2 Results

These are the results that we obtained:

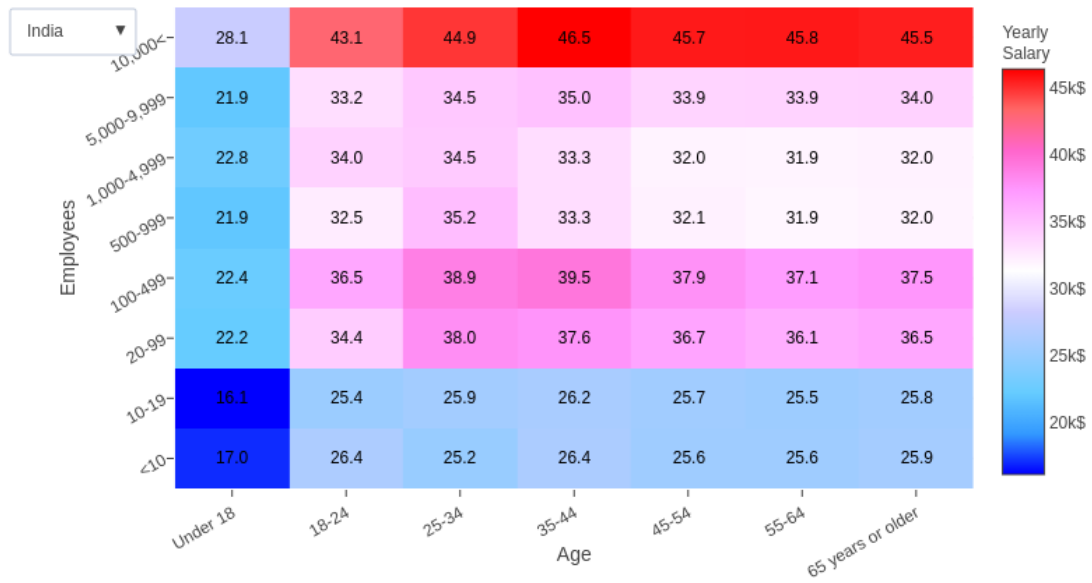




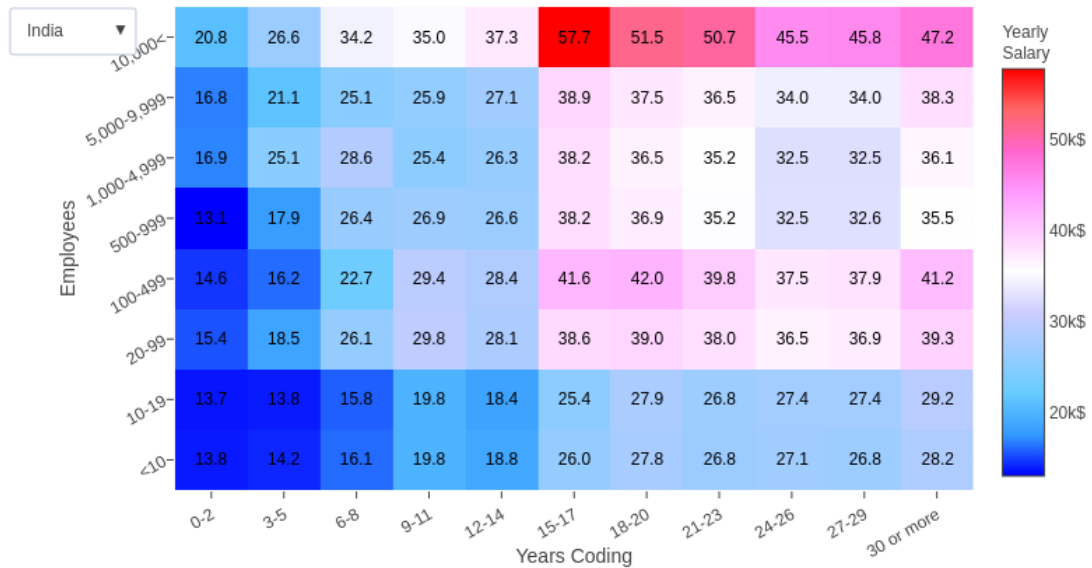
Here are the results that the models achieved:

| Model Used | Error Percentage (on test set) (%) |
|------------------------------|------------------------------------|
| Linear regression | 63.9% |
| Gradient Boosting regression | 41.8% |

Median Salary Of A Data Scientist In India



Median Salary Of A Data Scientist In India



Chapter 6

Conclusion

6.1 Conclusion

We can observe that the most popular jobs for programmers is in web development as - front-end developer, backend developer and full-stack developer.

We can also observe that a large majority of software developers around the world earn between 10,000 USD to 25,000 USD.

The data also shows that the top 5 countries that produce software developers are the United States, United Kingdom, Germany, India and Canada.

Finally, the data also points towards the gender disparity in the software industry - where there are at least 5 times more male software developers than female developers.

We then created a salary prediction model using this data that tries to predict the salary of an individual based on trends seen in the industry. We used 2 algorithms - linear regression and gradient boosting. The first model that used linear regression performed significantly worse than the second model that used gradient boosting. The second model gave 22.1% more accurate results as compared to the first.

This was expected because gradient boosting is a more sophisticated algorithm than linear regression.

6.2 Further Steps

- We can try to further explore the impact of other types of algorithms on the predictions.
- We can use Random Forest algorithm, Support Vector Machines and neural network based models.

References

- [1] Educative.io, [What is Matplotlib](#), 2017
- [2] Plotly documentation, [Getting Started](#), 2015
- [3] Seaborn documentation, [Statistical Data Visualisation](#), 2009
- [4] Educative.io, [What is Pandas in Python](#), 2017
- [5] Numpy documentation, [What is Numpy](#), 2008
- [6] Codecademy, [What is Scikit-Learn](#), 2016
- [7] Explained.ai, [How to explain gradient boosting](#), 2017
- [8] Yale University, [Linear Regression explanation](#), 2014
- [9] Kaggle, [StackOverflow 2019 Dataset](#), 2020
- [10] Yale University, [Scatterplot](#), 2014
- [11] Yale University, [Correlation Coefficient](#), 2013
- [12] Mason and Baxter, [Boosting Algorithms as Gradient Descent in Function Space](#), 1999
- [13] Brenner, M. H., & Lockwood, H. C, Journal of Applied Psychology, [Salary as a predictor of salary: A 20-year study.](#), 1965
- [14] Social Work Research and Abstracts, [Gender and salary inequity: statistical interaction effects](#), 1982
- [15] Social Behavior and Personality: an international journal, [THE MULTIFACTORIAL ACHIEVEMENT SCALE AS A PREDICTOR OF SALARY GRO...](#), 1995

-
- [16] Karla H. Hamlen, William A. Hamlen, [Faculty salary as a predictor of student outgoing salaries from MBA programs](#), 2015
- [17] Fazio, Kilpatrick and Baker, Stanford University, [Predicting Compensation for Job Seekers](#), 2016
- [18] Martin, Mariello and Battiti, [Salary Prediction in the IT Job Market with Few High-Dimensional Samples: A Spanish Case Study](#), 2018
- [19] Rupashri Barik, JIS College of Engineering, [Salary Prediction Using Regression Techniques](#), 2020

Appendix

Here are the system requirements:

- **Processor:** Intel CORE i3 - 3225 3.3 GHz
- **Memory:** 4GB RAM
- **Graphics Card:** Intel HD 4000
- **OS:** Windows 10 64 bit
- **Storage:** 1GB Available HD Space

Here are the software requirements:

- **Jupyter Notebook**
- **Python:** Version 3.9.0
- **Pandas:** 1.2.3
- **Scikit-learn:** 0.23
- **Matplotlib:** 3.4.0
- **Plotly:** 4.14.3
- **Numpy:** 1.20.3
- **Seaborn:** 0.11.1