Predictive Modeling And Analysis Of Software Engineers Salary Using Machine Learning

1st Md Abdullah Al Mahmud Pias
Department of electrical and computer science engineering
North South University
Dhaka, Bangladesh
abdullah.pias@northsouth.edu

2nd Marop Hossain
Department of electrical and computer science engineering
North South University
Dhaka, Bangladesh
marop.hossain@northsouth.edu

3rd Md. Jisan Ahmed
Department of electrical and computer science engineering
North South University
Dhaka, Bangladesh
jisan.ahmed@northsouth.edu

***Abstract*—Modern technology has contributed to modern society, and software developers play a significant part in developing and maintaining this technological world. But in this complex field, accurate salary prediction is crucial for job seekers and recruiters. But by the machine learning approach for predicting software engineering salaries effectively. By leveraging a data-set of historical salary information and various relevant features such as years of experience, education level, and geographical location, we employ a regression model to estimate software engineers' future salary or current possible salary. The methodology integrates data prepossessing, feature selection, model training, and evaluation to construct a robust prediction web framework. We consider several machine learning algorithms, including linear regression, decision trees, random forest, Support vectors, XGB, Save Model, and methods with extensive performance comparisons.**

Our research demonstrate the effectiveness of machine learning models in accurately forecasting software engineers' salaries by getting an highest accuracy of 76% and lowest Root mean squared error of 9.59 %. Also, this model empowers job seekers and employers to make more informed decisions regarding salary expectations. This research contributes to the field of software engineering by offering a valuable tool for salary negotiation and career planning, enhancing transparency and efficiency in the software job market.

***Index Terms*—Software engineer, Salary prediction, Machine Learning, Predictive modeling, Feature engineering, Model evaluation, Data preprocessing, Regression, Decision trees, Random forests,XGB.**

I. Introduction

The demand for software engineers is growing day by day with the improvement of technology. Currently, the software engineering field is the most challenging for a recruiter and a job seeker to predict salary. In most cases, a recruiter can offer a salary, and the job seeker thinks that it is not perfect for them; they feel that they deserve more. Also, the recruiter might offer more salary than a job seeker deserves. Because sometimes salary can vary by geographical location.In this article, we look into this problem and present a machine-learning approach to forecast the salary of software engineers by geographical location. The model described in this article will aid a prospective job seeker and recruiter in making a more informed decision while predicting the salary of a software engineer.

Identify applicable funding agency here. If none, delete this.

A. Software engineers salary in different top leading countries

The USA is the leading country in demand for software engineers, and this country tends to be higher than the demand for skilled software engineers in the technology sector. The median annual income for software developers, which includes software engineers, was $110,140 in May 2020, as reported by the US Labor Statistics. Also, the lowest 10% earned less than $63,250, while the wealthiest 10% made more than $164,590. [1] Also, it can vary by the different state. If we see that in San Francisco, software engineers get 31% higher salaries than the national average, whereas in Austin, it was only 14%. [2] Through the research, we also found that salary can vary based on economic conditions and skill base, like entry-level, mid-level, and senior-level [3], [4]. Also, we research in terms of India, [5] the Southeast Asian leading software engineers' labor market. Their entry-level software gets around $8000 to $10000 in terms of the USA, which is 90% less, where both have the same qualifications and skills. [6]

B. Applicability of Machine Learning in Predicting the salary of Software Engineers

A software engineer's salary is determined by some criteria, including their educational background, work experience, talents, and place of employment [7].It is not easy to effectively estimate their salaries using rule-based algorithms due to the complexity of these affecting variables. Using inductive machine learning techniques to extract salary predictions from the data-set is a more practical strategy. [8] For this reason, using a machine learning approach is a good fit for this specific application.

C. Research Goal

This research outlines several key objectives and contributions related to predicting the salary of software engineers

using machine learning approaches. Here's a summarized version:

1) Collect a large data-set of software engineer salaries and discover the most important features that can be utilized for predicting salary.
2) Using machine learning algorithms, predicted the salaries of software engineers with high accuracy.
3) To deploy the model as a web application on a local machine for ultimate deployment to end users.

The primary contributions of this research article include:

1) Collecting and carefully collecting a real-world data-set of software engineer salaries, which will be a great resource for the research.
2) Building an accurate predictive model for estimating salary, utilizing the collected data-set.
3) Evaluating the model's performance by comparing its predictions against unseen data, ensuring its reliability.
4) Deploying the model as a web application, making it accessible to end users for future use.

The rest of the article is structured as follows:

1) (Section II) Related Works:Reviews existing research in the field.
2) (Section III) Research Methodology:Details the steps involved in the research process.
3) (Section IV) Result:Discusses useful results derived from the research.
4) (Section V) Discussion: describes, analyzes, and interprets findings from the research.
5) (Section VI) Conclusion:The article ends.

II. Related Works

[11] Ignacio Martin and colleagues developed a machine learning model for predicting predicting IT Job salary. They collected a data set from a crawler with 3970 jobs with the feature of part-time and full-time employment and around 488 companies included. They test multiple models like the random forest, AdaBost, Vote3, and KNN and get the highest accuracy 85% in vote3 with precision of 0.837.

[12]Abu Samah and colleagues devloped a linear regression approach to predicting salaries with visualizations of job vacancies based on Malaysia job street. For this research, they collected data from ww.jobstreet.com using web scraping, where 22,250 jobs data with attributes like b title, salary, company, location, description, requirements, qualification, job type, career level, and years of experience. The author also made a web-based dashboard using Python for future uses. By functionality test, they get 96.58% accuracy in the linear regression machine learning model.

[13]Prof. D. M. Lothe and colleagues developed a linear regression model for predicting salary. They collected data from companies with job types like CFO, CEO, and Manager. Where they have features like degree, major, experience, and industry, they tested three different models: random forest, Ridge Regression, and Polynomial Transformation, and got the highest accuracy, 76% in Polynomial Transformation. And they mention it as a future work to add a web application with that model.

[14]Krishna Gopal and colleagues developed a machine learning salary prediction system for school students. They experimented on the student dataset using ten cross-validation. As features, they use student gender, courses, program enrolled, grade, and salary. As a training model, they used a classification algorithm Decision tree, Naive bias, and KNN and got the highest accuracy of 98% in classification.

[15]Ashty Kamal Mohamed Saeed and colleagues developed a machine learning computer engineering salary prediction system based in India. They have collected data from the Zonedo website, which have 11 variable and 2002 observation. As a feature of the dataset, they use Designation, city, Gender, age, Degree, Specialization, college GPA, graduation year, and English. They work on three algorithms: Naive bias, SVM, and Random forest. They got less than 50% accuracy for every algorithm, And the highest accuracy is 41%, which was obtained for Naive Bias classification.

[16]Yasser and colleagues developed a machine learning salary prediction system. They have collected survey data from the Saudi Arabia Labour market. As a feature of the dataset, they use estimated mean annual salary across economic activities and significant occupational groups. They worked on a Linear regression model and got an accuracy of 94%. Also, they can propose a framework for determining the yearly salary ranges for various economic activity categories, organization sizes, and professions.

III. Methodology

The approach adopted in this work is outlined in Figure 1:

A. Data Acquisition

We have collected our dataset from stack overflow official website. Stack Overflow is a professional community for developers. Every year since 2011, they conduct a developer survey and release the collected data open-source on the web. Developers from almost 180 countries participate in this survey [17]. Stack overflow then collects all the data and then publish them as a csv file in their official website every year at once. The participants are mostly from the US, India, UK and EMEA regions. The majority of the survey respondents had a background of developer/ coding experience. For our project, we have selected the 2020 Stack overflow survey dataset. Almost 65000 developers from all over the world have participated in this survey [18]. The initial dataset has 64461 instances and 61 features. But we have selected total five attribute including the target class. The dataset that we have used to generate results can be found in [19] description of selected attributes from data set in given in table 1.

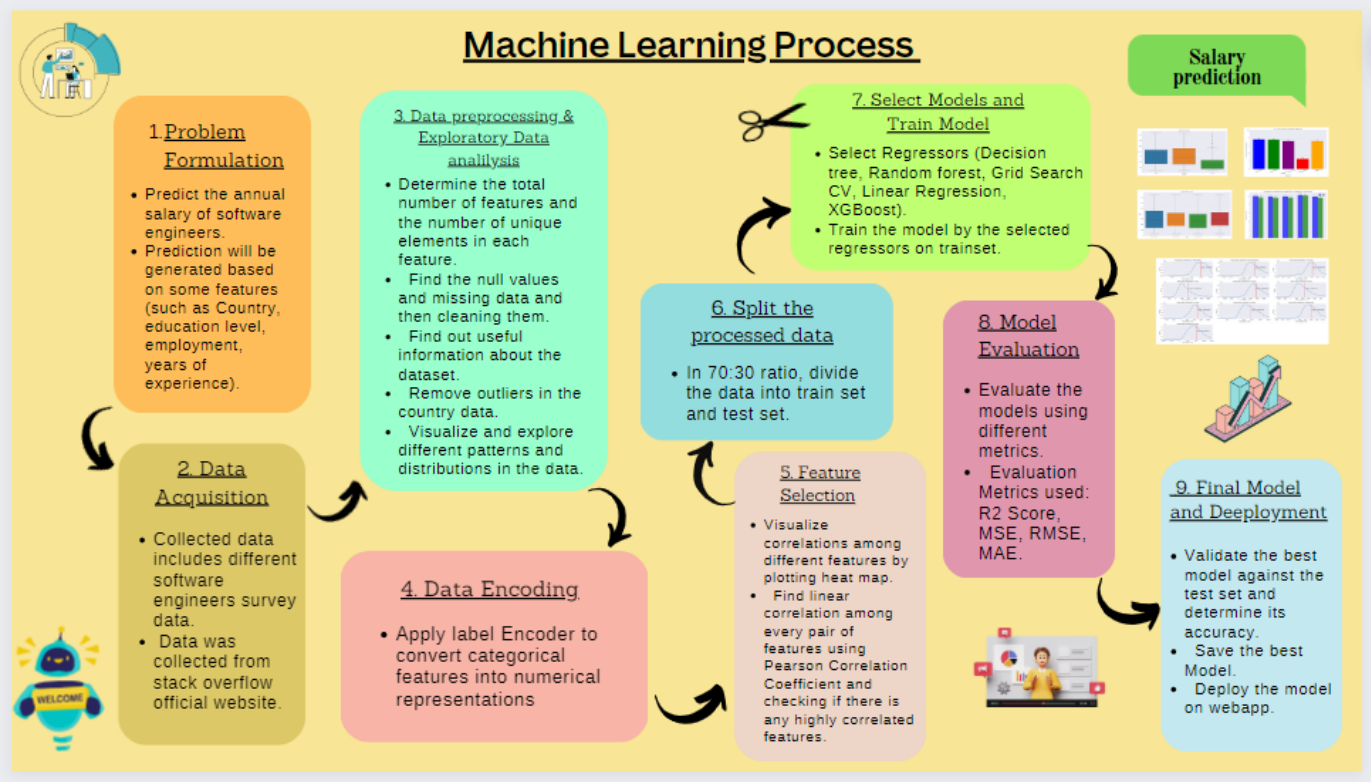

Fig. 1. Flow Chart Of Research Methodology

Description of selected attributes.	
Variable Name	**Description**
Country	Name of the countries
Edlevel	Education lavel of each employees
Employment	Employment status of them
YearsCodePro	Years of experience of them
ConvertedComp	Annual salary of the empoyees in USD

TABLE I
DESCRIPTION OF SELECTED ATTRIBUTES.

Number of unique values for all the features before prepossessing.	
Feature Name	**Unique Values**
Country	9
Employment	7
YearsCodePro	52
Country	183

TABLE II
NUMBER OF UNIQUE VALUES FOR ALL THE FEATURES BEFORE PREPOSSESSING.

B. Data Pre-Processing

The data-set may not be always in good format for operating machine learning algorithms using it. Data pre-processing is very crucial to ensure that machine learning algorithms can be effectively used using the data-set. We have identified a few problems upon analyzing our initial data-set. First of all, we selected 5 attributes from the data-set that are much relevant for predicting the results. So we eliminated the other attributes and only selected them. Each of the attributes now contain different unique values. It is shown in table 2.

Now we investigate the null values that each attributes have. From table 3, we can see that the missing rate of values are highest is salary attribute. The other features also contain missing values. So, we needed to pre-process the data to remove null entries in our data frame.

Total entries of each features before prepossessing.	
Feature Name	**Total Entries**
Country	64072
EdLevel	57431
Employment	63854
YearsCodePro	46349
Salary	34756

TABLE III
TOTAL ENTRIES OF EACH FEATURES BEFORE PREPOSSESSING.

So we filtered the Data Frame to exclude rows where the Salary column has a null (NaN) value using the operation df[df[”Salary”].notnull()]. Thus the Salary attribute doesn’t contain any null value in the data frame. It has now 34756 entries in total. But we can see that the other attributes still have some null values in table 4. So we use df.dropna() to drop the rows containing any NaN values in any column of

the DataFrame. Now we don't have any null entries in our dataframe and each of the attributes contains 34025 entries in total.

Total entries of each features after removing null values in Salary.	
Feature Name	**Total Entries**
Country	34756
EdLevel	34188
Employment	34717
YearsCodePro	34621
Salary	34756

TABLE IV
TOTAL ENTRIES OF EACH FEATURES AFTER REMOVING NULL VALUES IN SALARY.

Another challenge was to resolve the inconsistency in the values of feature Country. From table 5, we can see that the value counts of United States are highest. India is the second highest in this case and then United Kingdom and so on. But if we look at the bottom part of table 5, we can see that countries like Yemen, Malawi, Burkina Faso etc. have only one software engineers participating in this survey. This will affect the performance of our model a lot.

Before processing the feature Country.	
Country	**Value Counts**
United States	8082
India	2563
United Kingdom	2551
Germany	2206
Canada	1293
...	...
Brunei Darussalam	1
Guinea	1
Burkina Faso	1
Malawi	1
Yemen	1

TABLE V
BEFORE PROCESSING THE FEATURE COUNTRY.

To resolve this inconsistency, we have assigned a cutoff region with value 200. That means the countries that have more than 200 software engineers, we will only consider them. The associated rows of the other countries are mapped to a new entry caller Others. Surprisingly, Others includes 4369 rows which is the second highest value Count in Country attribute shown in table 6. Now, each of the attributes including our target class contain 34025 entries in total.

After plotting the Salary vs Country in a boxplot (shown in figure 2), we have seen that only United states has outliers when salary is greater than a certain range and all the other countries contains salaries less than that. Thus, we have assigned a certain range of $5000 to $1500000 and the rows associated with the salaries outside of this range will be removed from out dataframe (shown in figure 3). Again we have dropped the rows that have feature Country as 'Others' as we are not concerned about them.

After processing the feature Country.	
Country	**Value Counts**
United States	8082
Other	4369
India	2563
United Kingdom	2551
Germany	2206
Canada	1293
Brazil	1139
France	1103
Netherlands	798
Poland	789
Australia	755
Spain	744
Italy	660
Russian Federation	595
Sweden	573
Turkey	378
Switzerland	342
Israel	323
Pakistan	318
Romania	313
Mexico	312
Czech Republic	307
Ukraine	296
Austria	296
South Africa	281
Ireland	280
Iran	270
Norway	270
Belgium	255
Denmark	242
Portugal	237
Argentina	233
Hungary	227
Finland	224
New Zealand	201
Greece	200

TABLE VI
AFTER PROCESSING THE FEATURE COUNTRY.

Now, it's time to clean feature YearsCodePro. Already, we have removed all the null entries. After cleaning Salary and Country features, YearsCodePro has now total 50 unique values. Most of them are integer and two of them are object datatype. They are 'More than 50 years' and 'Less than 1 year'. We have converted the first one to 50 and second one to 0.5 as integer number. Now YearsCodePro has all the values in integer format.

Again, we clean feature Edlevel. It has total 9 unique values. In this part, we are considering 'Bachelor's degree', 'Master's degree', 'Professional degree' and 'Other doctoral'. 'Professional degree' and 'Other doctoral' are combined and converted to 'Post grad'. Finally, the other values that the feature Edlevel have are converted to 'Less than a Bachelors'. Now it has four unique values and they are 'Bachelor's degree', 'Master's degree', 'Less than a Bachelors', 'Post grad'.

Furthermore, we have cleaned the feature Employment. It has total 3 unique values now after processing Salary, Country, YearsCodePro, Edlevel. We have converted 'Employed full-

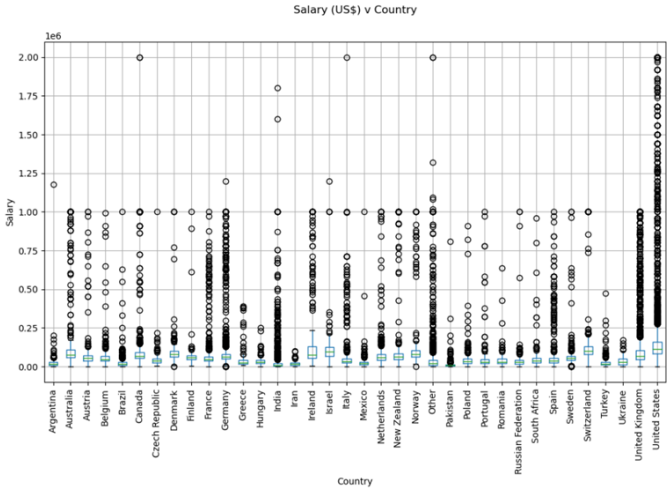

Fig. 2. Box-plot of Salary (US$) vs Country before removing outliers

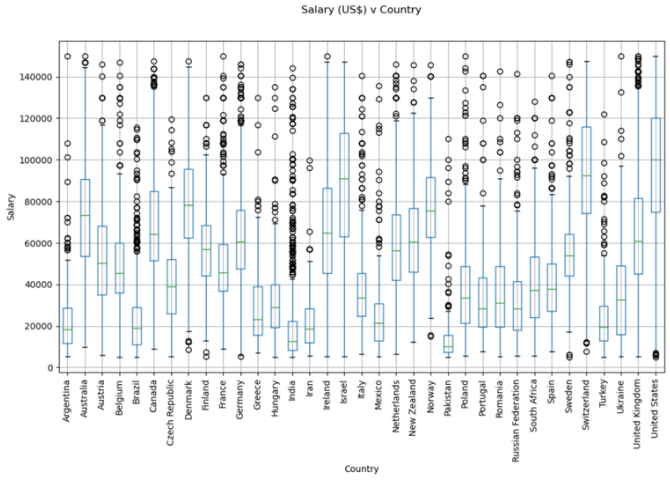

Fig. 3. Box-plot of Salary (US$) vs Country after removing outliers.

time' to 'Full time', 'Employed part-time' to 'Part time' and 'Independent contractor, freelancer, or self-employed' to 'No employment'.

After data pre-processing and cleaning all the features, the total entries in each features including the target class are 25024 shown in table 7.

Total entries of each features after preprocessing	
Feature Name	**Total Entries**
Country	25024
EdLevel	25024
Employment	25024
YearsCodePro	25024
Salary	25024

TABLE VII
TOTAL ENTRIES OF EACH FEATURES AFTER PREPROCESSING

C. Exploratory Data Analysis

This box plot in (figure 4) shows the distribution of the salaries among the four education levels: less than a bachelor's degree, bachelor's degree, master's degree, and post-graduate degree. The box plot shows the median salary (the middle line in the box), the 25th and 75th percentiles (the edges of the box), and the minimum and maximum salaries (the whiskers). The median salary is highest for post-graduate degrees, followed by master's degrees, bachelor's degrees, and less than a bachelor's degree. The interquartile range is also widest for post-graduate degrees, followed by master's degrees, bachelor's degrees, and less than a bachelor's degree. This suggests that there is more variability in salaries among those with post-graduate degrees than among those with other education levels.

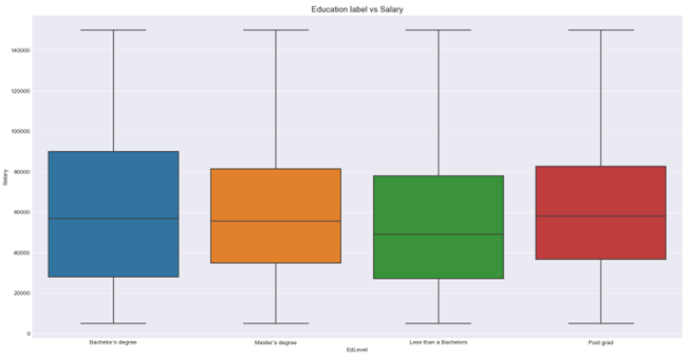

Fig. 4. Distribution of the salaries among the four education levels.

This box plot in (figure 5) compares employment status and salary. In this particular box plot, the employment status categories are "No employment," "Part time," and "Full time." The salary is represented on the vertical axis. Comparing the boxes across employment categories reveals the overall salary distribution trends. In this case, the box for "Full time" employment is positioned considerably higher than the boxes for "No employment" and "Part time," indicating that full-time employees generally earn higher salaries. The wider spread of the "Full time" box suggests a greater range of salaries among full-time employees compared to those who are not employed or employed part-time.

This beelow graph in (figure 6) shows the yearly salary distribution in eight different countries: United States, United Kingdom, Germany, India, Canada, Brazil, France, Poland, Netherlands, and Spain. Each graph shows the distribution of salaries in that country, with the x-axis representing the yearly salary in USD and the y-axis representing the number of people earning that salary. The mean salary for each country is also indicated. The salary distribution in the United States is the most skewed to the right of all the countries shown in the graph. The mean salary in the United States is also the highest of all the countries shown, at $97,402.40. Canada has the second highest mean salary, at $69,305. The United Kingdom has the third highest mean salary, at $65,644 and

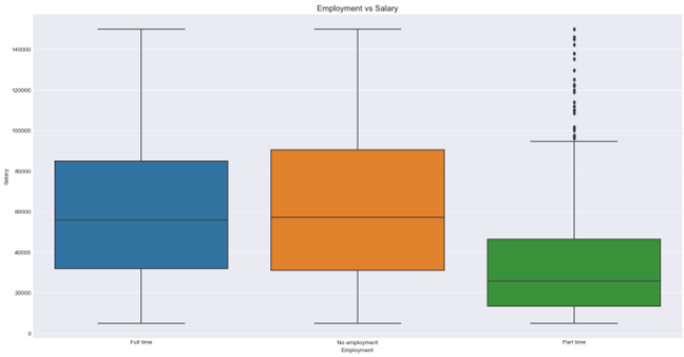

Fig. 5. Distribution of the salaries among the four education levelComparing employment status and salarys.

India has the lowest mean salary of the eight countries, at $19,112.

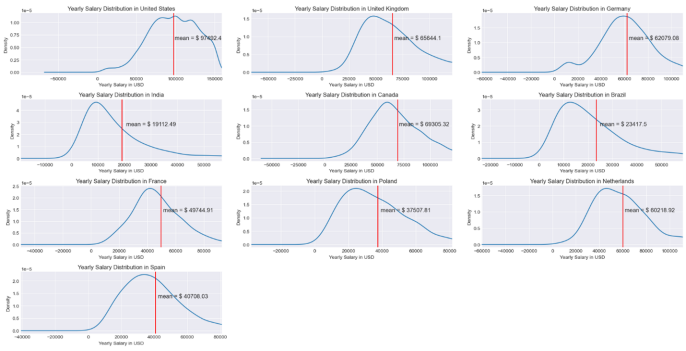

Fig. 6. Yearly salary distribution in eight different countries.

D. Data Encoding

After pre-processing, we have a number to categorical values such as Country, EdLevel and Employment in our dataset. To be able to train our model, we created a numerical representation of these categorical values. We used label encoder [20] from scikit-learn [21] to get a numerical representation of the features. Label encoding means converting categorical features into numerical values.

Thus in the Country feature all the values are converted to corresponding numbers. In Feature Edlevel, 'Bachelor's degree' is converted to 0, 'Master's degree' is converted to 1, 'Post grad' is converted to 2 and 'Less than a Bachelors' is converted to 3. Similarly, in feature Employment, 'Full time' is converted to 0, 'Part time' is converted to 1, 'No employment' is converted to 2. Thus all the categorical values in converted into numerical formal.

E. Feature Selection:

In (figure 7), we visualize the correlations among different features by plotting heat map. If there are strongly positive or strongly negative correlations between the feature, then we basically drop then and take only one. We used the Pearson correlation co-efficient [22] to measure the strength and direction of a linear relationship between two features.As there is no strongly correlation between the features we have selected, we don't need to drop any of them (shown in Figure 7),

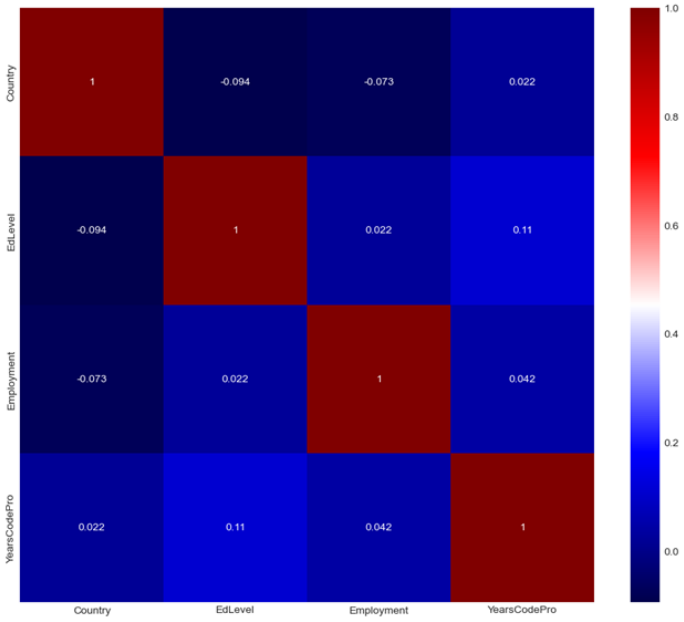

Fig. 7. Visualize correlations among different features by plotting heat map.

F. Data Splitting

It is very important to split the dataset for a machine learning project. We may assess how effectively a model responds to new data by segmenting the data. Also the performance of our regressors are also increased while segmenting the data. In our project we have splitted our dataset. Our train set contains 70% of the data and test set contains rest 30% of the available data. Before splitting, our dataset contains total 25024 entries (shown in table 7). After splitting, our train set contains 17516 rows and teest set contains 7508 rows.

G. Regressors Used

In our project, we have used total 5 regressors. They are Decision Tree Regressor, Random Forest Regressor, Grid SearchCV Regressor, Linear Regression and XGBoost Regresssor.

1) Decision Tree: Decision Tree [23] is a supervised machine learning technique that builds regression or classification models. It uses a tree-like structure where the leaf nodes are the outcomes.All the other nodes except the leaf nodes are called decision nodes where further splits are made depending on yes/no questions. How a decision tree splits the data is often determined by entropy or Gini index. In this work, we used

Gini index to split the data in decision tree. Following equation show the formula for Gini index equation 1:

$$\text{Gini Index(S)} = 1 - \sum_{i=1}^{n} p_i^2 \tag{1}$$

where S is the subset of the training data and pi is the probability of the class.

2) Random Forest: Random Forest [23] is an ensemble learning method that uses multiple decision trees in order to create classification or regression model. Random forest consists of a large number of decision trees that work as an ensemble. Each individual trees predict the value of the target class, and their predictions are combined in order to get more accurate prediction. Random forest is a supervised machine learning algorithm that can be used to solve both classification and regression problems.

3) Grid SearchCV: Grid SearchCV [24] is similar like decision tree regressor. In decision tree, it is hard to find the best hyperparameters. So we first define some parameters and using grid search cv algorithms we can find the best suited parameters for our model. In this case, we use the max_depth and random_state as our parameters. FInally, we apply Decision Tree Regressor on our dataset after finding the best Hyper parameters.

4) Linear Regression: Linear Regression [25] is a regression model that assumes a linear relationship between the features (X) and the target class (y). With this model, target class is derived from the linear combination of the input variables. Based on number of variables, it can be a univariate linear regression or multivariate linear regression.A linear regression line has an Equation 2 of the following form:

$$\text{Y} = A + BX \tag{2}$$

Here,Y is the predicted dependent variable and X is the independent variable. The slope of the line is B and a is the intercept.

5) XG Boost: XGBoost [26] (Extreme Gradient Boosting) is a powerful and efficient machine learning algorithm known for its speed and performance in supervised learning tasks. It belongs to the ensemble learning category, specifically the gradient boosting method.

IV. Result and Findings

A. Performance of the models and selecting the best one

We have trained our dataset with 5 different repressors. We have used some metrics to determine which regressor performs the best. They are RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) and R-squared score.
R2 score measures how close the data fit with the regression line. It is closely related to Mean Squared Error. Following Equation 3 is used to calculate R2 score:

$$\text{R}^2 = 1 - \frac{SSR}{SST} \tag{3}$$

Here, RSS = sum of squares of residuals, and TSS = total sum of squares.

Root Mean Squared Error is the standard deviation of the prediction errors. It is calculated using the following Equation 4:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{a}_i)^2} \tag{4}$$

Here, pi is the predicted value and ai is the actual value.

Mean Absolute Error is a measure of average distance between the real data and the predicted data. It is calculated by using the following Equation 5:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|p_i - \hat{a}_i| \tag{5}$$

We have trained all our models using our dataset and calculated the R-squared score, RMSE, MAE for each of them. After training we got the metrics values that are mentioned above and also tested all our models by giving input data ['United States', 'Master's degree', 'Full time', '15']. Each of the models predicted the salary according to the given input data. After training all the models and testing them individually, we selected our best model which is XgBoost [26] that has 72% R-Squared in train set and 65% in test set shown in table table 8. The Linear regression [25] model gives us the lowest performance for both train and test set (shown in figure 8). After selecting the best model, we have used pickle [27] that converts a Python object into a character stream to save our selected model..

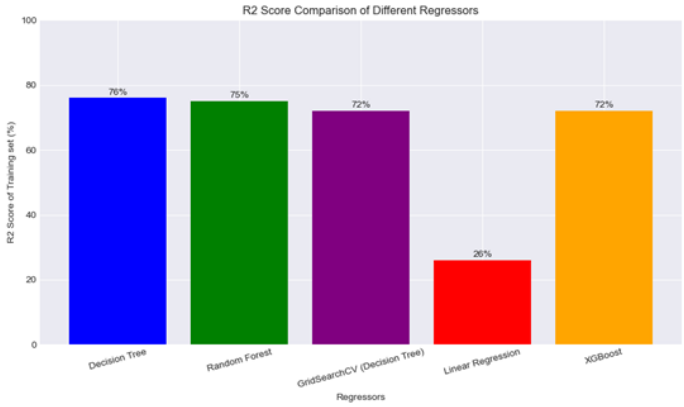

Fig. 8. R2 Score Comparison of Different Regressors in the Training Set

Models	R2 Score(%)		Log RMSE(%)		Lof MAE(%)	
	Train Set	Test Set	Train Set	Test Set	Train Set	Test Set
Decision Tree	76	58	9.81	10.09	9.43	9.75
Random Forest	75	63	9.83	10.2	9.50	9.71
Gird SearchCV	72	63	9.91	10.03	9.59	9.70
Linear Regression	26	26	10.37	10.38	10.17	10.18
XGBost	72	65	9.59	10.00	9.56	9.67

TABLE VIII
PERFORMANCE OF CLASSIFIERS THAT WE HAVE USED FOR BOTH TRAIN AND TEST SET

This graph (figure 9) shows a bar graph comparing the R-squared scores of various regression models, including Linear Regression, Decision Tree, Random Forest, GridSearchCV (Decision Tree), and XGBoost. The R-squared score is a statistical measure that indicates how well a regression model fits the data. A higher R-squared score generally indicates a better fit. In this case, the Decision Tree model has the highest R-squared score, with a value of 76% for the training set. This suggests that the Decision Tree model is able to explain the most variance in the data compared to the other models. The XGBoost and GridSearchCV (Decision Tree) models have similar R-squared scores of 72%, while the linear regression model has a slightly lower score of 60%. The Linear Regression model has the lowest R-squared score, with a value of 26%. This suggests that the Linear Regression model is the least effective at explaining the variance in the data.

Fig. 9. R2 Score Comparison of Different Regressors in the Testing Set

This graph(figure 9) shows a bar graph comparing the R-squared scores of various regression models, including Linear Regression, Decision Tree, Random Forest, GridSearchCV (Decision Tree), and XGBoost. The R-squared score is a statistical measure that indicates how well a regression model fits the data. A higher R-squared score generally indicates a better fit. In this case, the XGBoost model has the highest R-squared score, with a value of 65% for the testing set. This suggests that the XGBoost model is able to explain the most variance in the data compared to the other models. The Random Forest and GridSearchCV (Decision Tree) models have similar R-squared scores of 63%, while the Decision Tree model has a slightly lower score of 58%. The Linear Regression model has the lowest R-squared score, with a value of 26%. This suggests that the Linear Regression model is the least effective at explaining the variance in the data.

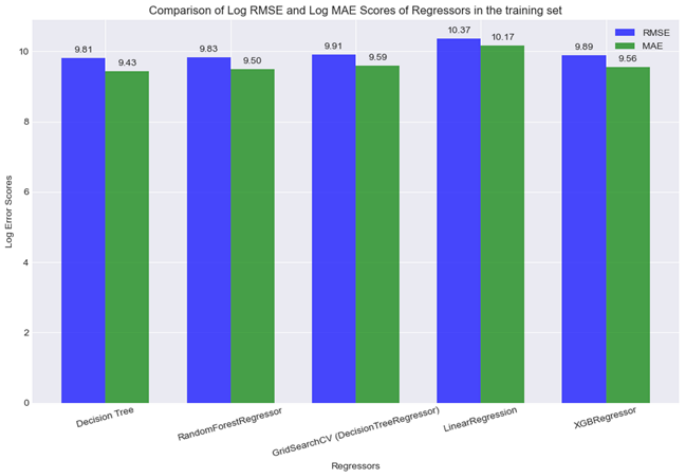

Fig. 10. Comparison of Log RMSE and Log MAE Scores for Different Regressiors in the Training Set

This graph (figure 10) compares the scores of regressors in the training set between the root mean squared error (RMSE) and the mean absolute error (MAE). The RMSE scores are higher than the MAE scores, which means that the RMSE is a more sensitive measure of error than the MAE. This is because the RMSE squares the errors before taking the average, which means that large errors are given more weight than small errors. The MAE, on the other hand, simply takes the average of the absolute values of the errors, which means that all errors are given equal weight. The graph shows the results for five different regressors: LinearRegression, XGBRegressor, DecisionTreeRegressor, RandomForestRegressor, and GridSearchCV (DecisionTreeRegressor). The LinearRegression regressor has the highest RMSE score of 10.37, which means that it is the most sensitive to errors. The GridSearchCV (Decision Tree Regressor) has the lowest RMSE score of 9.83, which means that it is the least sensitive to errors.

This graph (figure 11) compares the scores of regressors in the training set between the root mean squared error (RMSE) and the mean absolute error (MAE). The RMSE scores are higher than the MAE scores, which means that the RMSE is a more sensitive measure of error than the MAE. This

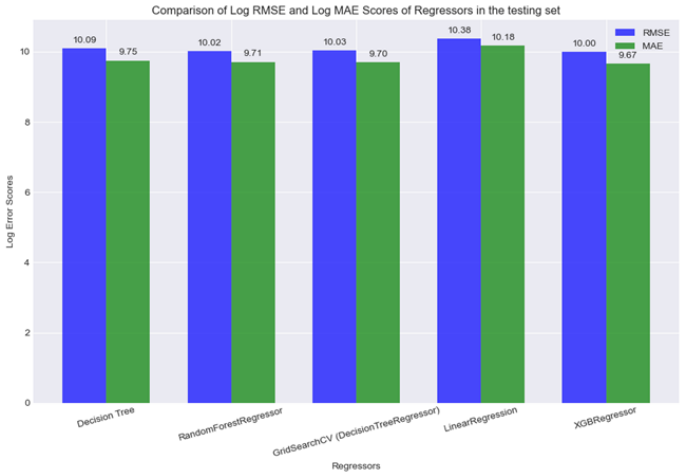

Fig. 11. Comparison of Log RMSE and Log MAE Scores for Different Regressiors in the Testing Set

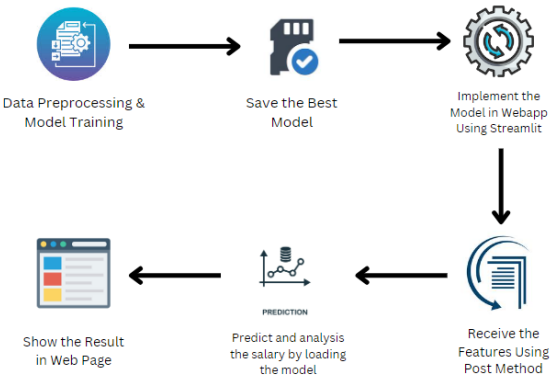

Fig. 12. Architectural diagram of the deployment model.

is because the RMSE squares the errors before taking the average, which means that large errors are given more weight than small errors. The MAE, on the other hand, simply takes the average of the absolute values of the errors, which means that all errors are given equal weight.

The graph shows (figure 11) the results for five different regressors: LinearRegression, XGBRegressor, DecisionTreeRegressor, RandomForestRegressor, and GridSearchCV (DecisionTreeRegressor). The LinearRegression regressor has the highest RMSE score of 10.38, which means that it is the most sensitive to errors. The XGBRegressor has the lowest RMSE score of 10.00, which means that it is the least sensitive to errors.

B. Deployment in Webapp

The Deployment architechture of our model in webapp is given (figure 12). For deploying our model in webapp, we have used streamlit. It is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

Our Webapp has two different options, predict and explore (shown in figure 13). User can predict the annual salary by providing the input data using the predict option and also can explore the salary with other input features using the explore options.

1) Prediction of the salary: In this part, user will first select the features which are country, Education level, Employment and years of experience. After selecting, they just need to click the Calculate salary button. Then the estimated yearly salary

Fig. 13. Software Engineers Salary Prediction

based on the given features will be shown below (shown in figure 13).

After scrolling down the explore page, another bar graph will appear which basically shows the mean salary based on the countries. It is highest in United states, then Switzerland, Israel and so on (shown in figure 14). Also there is a graph which shows the mean salary based on years of experiences of the software engineers (shown in figure 15).

V. Discussion

Our findings significantly impact the software engineering field, making predictions using machine learning. We have successfully built a strong predictive model that considers various characteristics such as education, experience, talents, geographical location, and others. The model's precision and dependability are shown in its capacity to generate accurate salary predictions, which improves honesty and equity in salary discussions. Key outcomes of our work include:

A. High Accuracy

In terms of software engineer salary prediction, our testing model frequently shows great accuracy. Its efficacy is shown by its 76% accuracy and lowest Log RMSE of 9.59 %.

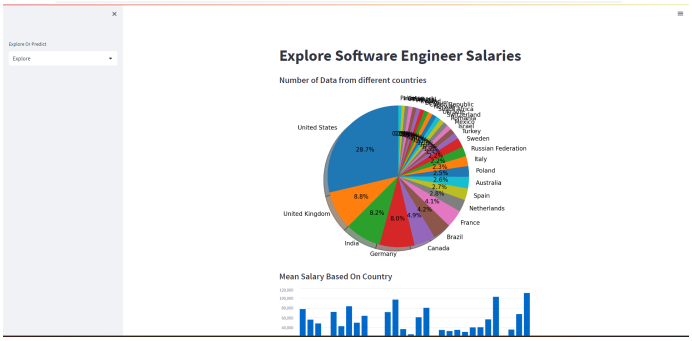

Fig. 14. Explore Software Engineers Salary By Geographic Location

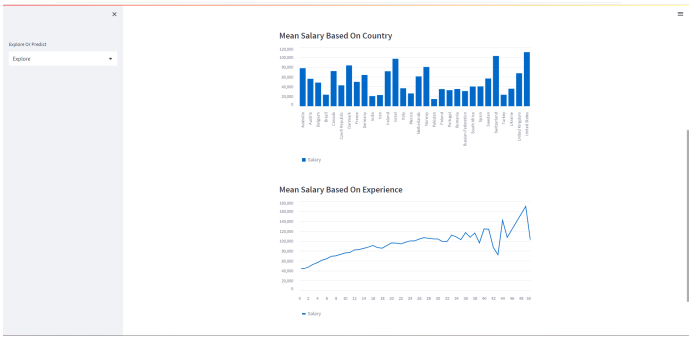

Fig. 15. Explore Software Engineers Salary By Geographic Location

B. Fairness

We have executed significant efforts to address concerns about bias and discrimination, ensuring that our predictions are objective and non-discriminatory. This commitment to fairness distinguishes our work.

C. User Friendly Deployment

The deployment of our testing model as a web application on a local computer enables software developers and employers to easily access and make predictions, speeding up the compensation-creating process.

D. Compression with other notable work

Our research develops on and varies from previous significant research in software engineer pay prediction using machine learning. A comparison of our findings to the work of these noticeable studies shows many significant contributions:

1) Accuracy and Transparency: Despite some previous research that might have had imperfect or unclear models, our work focuses on both fairness and accuracy. This combination defines our study as an effective tool for bargaining on salary.

2) Deployment and User Accessibility: Even though some earlier research may have produced accurate models, they frequently missed the deployment-related practicality. Our study closes this gap by offering a web application that is simple to use and accessible to employers and software engineers equally.

3) Inclusion of Non-Monetary Factors: In predicting salaries, a lot of previous research have mostly concentrated on financial factors. On the other hand, we take non-financial aspects into account, providing a more comprehensive understanding of the benefits plan.

Our research makes us unique in that it is transparent, accurate, practical, and dedicated to fairness. We demonstrate our work's distinctive contributions and ability to significantly improve software engineer salary practices by contrasting it with other significant research. This study establishes a standard for ethical and fact-based salary conversations in the software engineering sector as well as being a valuable tool for salary prediction.

VI. Conclusion

This research analyzed the use of machine learning in predicting software engineer salaries, focusing on application to the changing employment market. We found significant factors in software engineer salary, including education, experience, skills, geographical location, and more, by combining a wide data-set and employing modern data analysis tools. Our findings offer information on the dynamic relationship of these variables in making salary predictions.

We discovered that our predictive model, which is based on Decision Tree, Random Forest, Grid SearchCV, Linear Regression, and XGBost, has a remarkable accuracy rate of 76% after careful testing with various machine learning algorithms, making it a highly effective tool for forecasting software engineer salaries.

Also, our dedication to honesty and equity has guided us in dealing with prejudice and discrimination issues, ensuring that our predictions are objective and fair. We also evaluated the impact of non-monetary aspects in compensation calculation, such as remote work opportunities, business culture, and benefits.

Our testing model's deployment as a web application on a local workstation [28] is an essential phase in making this technology available to end users. This easy-to-use interface will make pay discussions and compensation conversations easier for software engineers and employers.

In the future, we are interested in expanding the scope of this research by collecting more data sets with a more significant number of essential variables, which will improve the precision of our testing models and their capacity to react to changing job market dynamics. As the technology industry develops, the findings of this study will be important in directing salary choices and ensuring that software engineers receive sufficient salary for their knowledge and contributions.

References

[1] US Bureau of labour statistics, Available Online: https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm
[2] Inded Software Engineers Salary Index. Available Online: https://www.indeed.com/career/software-engineer/salaries
[3] Talent.com Salary Index, Available Online: https://www.talent.com/salary?job=software+engineer

[4] Levels FYI Salary Index Available Online:https://www.levels.fyi/t/software-engineer?countryId=20
[5] Pay Scale Software Engineers Salary Index. Available Online:https://www.payscale.com/research/IN/Job=Software-Engineer/Salary
[6] Indeed Software Engineers Salary Index. Available Online:https://in.indeed.com/career/software-engineer/salaries
[7] Career Foundry Software engineers salary depend on. Available online: https://careerfoundry.com/en/blog/web-development/software-engineer-salary/
[8] Career Foundry Software engineers salary depend on. Available online: https://medium.com/analytics-vidhya/machine-learning-project-1-predict-salary-using-simple-linear-regression-d83c498d4e05
[9] Career Foundry Software engineers salary depend on. Available online: https://careerfoundry.com/en/blog/web-development/software-engineer-salary/
[10] Career Foundry Software engineers salary depend on. Available online: https://careerfoundry.com/en/blog/web-development/software-engineer-salary/
[11] Martín, Nacho,Mariello, Andrea , Battiti, Roberto , Hernández, José. (2018). Salary Prediction in the IT Job Market with Few High-Dimensional Samples: A Spanish Case Study.
[12] Abu Samah, Khyrina Airin Fariza , Wirakarnain, Nurqueen Hamzah, Raseeda , Moketar, nor aiza , Riza, Lala , Othman, Zainab. (2022). A linear regression approach to predicting salaries with visualizations of job vacancies: a case study of Jobstreet Malaysia.
[13] Prof. D. M. Lothe, Prakash Tiwari, Nikhil Patil, Sanjana Patil, Vishwajeet Patil (2021). Salary Prediction Using Machine Learning.
[14] Salary Prediction Using Machine Learning, Available Online: https://ijirt.org/master/publishedpaper/IJIRT151548-PAPER.pdf
[15] Saeed, Ashty, Abdullah, Pavel ,Tahir, Avin. (2023). Salary Prediction for Computer Engineering Positions in India.
[16] Matbouli, Y.T.; Alghamdi, S.M. Statistical Machine Learning Regression Models for Salary Prediction Featuring Economy Wide Activities and Occupations
[17] Available Online:https://survey.stackoverflow.co/2023/
[18] Available Online:https://insights.stackoverflow.com/survey/2020
[19] Available Online:https://insights.stackoverflow.com/survey
[20] LabelEncoder. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html
[21] LabelEncoder. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html
[22] Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In Noise Reduction in Speech Processing; Springer:
[23] Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In Noise Reduction in Speech Processing;
[24] . Liaw, A.;Wiener, M. Classification and regression by randomForest. R News 2002, 2, 18–22
[25] Available Online:https://www.w3schools.com/python/python-ml-grid-search.asp
[26] Montgomery, D.C.; Peck, E.A.; Vining, G.G. Introduction to Linear Regression Analysis; JohnWiley , Sons: Hoboken, NJ, USA, 2021
[27] Available Online:https://www.geeksforgeeks.org/xgboost/
[28] Available Online:https://docs.python.org/3/library/pickle.html
[29] Available Online:https://streamlit.io/