PREDICTING DEVELOPER SALARIES

This project was done by:

1. Abdikarim Gedi
2. James Wainaina
3. Patrick Okore
4. Rosemary Mburu
5. Samuel Kyalo
6. Sharon Atieno

The link to the GitHub repository is: GitHub repository

Project Background

Salary negotiation can be a **critical stage** in the job search process, and job seekers often encounter various challenges during this phase like lack of information on salary trends.

This means that a job seeker might spend valuable time researching industry salary trends. Some might not be so lucky as the information might be non-existent.

Africa's Tech sector has become one of the **fastest-growing** tech ecosystems in the world with tech being one of the fastest growing sectors in Africa. This has led to a **rise in demand for jobs in the industry.** However, **unlike other parts of the world,** information on remuneration in these jobs remains hard to come by.

Existing resources such as Glassdoor and Brighter Monday have **limited information on salaries in Africa.**

Over the past few years, it has been observed that foreign companies enter the African Market, offering more competitive salaries compared to local companies resulting in the mass movement of experienced developers into these new roles.

This project seeks to **solve this problem** by **developing a platform that can predict developer salaries based on their personal information**, and also, providing a **comparison** between different incomes in different regions for similar roles.

Stakeholders:

- Jobseekers
- Employers
- Recruitment agencies

Business Understanding

Salary negotiation can be a critical stage in the job search process, and job seekers often encounter various challenges during this phase like lack of information on salary trends. This means that a jobseeker might spend valuable time researching industry salary trends. Some might not be so lucky as the information might be non-existent.

As the Tech labour market becomes more competitive, offering the right salary for new and current employees is crucial for employers as it means keeping or losing a valued resource. Thus it is imperative for them to offer fair and competitive compensation that is benchmarked to their industry.

Our project looks at coming up with a salary prediction model to help both job seekers and employers with the above challenges. We will focus on the tech industry (developers) and use data from stack overflow's annual developer survey.

Objectives

The **main objective** of this project is to come up with a salary prediction model that will:

1. Enable **Jobseekers** to ask for **competitive salaries** during contract negotiations.
2. Assist **employers** in offering **fair compensation** to their employees.
3. Assist **recruitment agencies** to offer **accurate salary estimates** to their clients.

These objectives will be achieved through the following **specific objectives:**

1. To select the **most important features** in the dataset to be used in Salary prediction.
2. To **describe how features** such as Professional experience and Education level **affect Annual compensation.**

3. To build a baseline model and iterate the baseline model by building multiple models, selecting the best model that meets the success criteria.
4. To **deploy the model** as an **online application.**

Success Metrics

The metrics to be used to measure the success of the model are:

For regression models:

- **Root Mean Square Error (RMSE)** which quantifies the average difference between the predicted values from the model and the actual values in the dataset.

For Classification models:

- **Accuracy score**. Calculates the ratio of correctly classified instances to the total number of instances across all classes.

An **Accuracy score** of 71**% on the test data** will be considered a success.

Data Understanding

The data used for this project is sourced from **Stack Overflow's annual developer survey for the year 2022**. The dataset consists of responses from developers around the world and contains information about various aspects of their professional lives, including their salaries.
It contains **73,268 responses** and **79 features**, i.e., 79 data points about each developer.
The target variable of interest is "ConvertedCompYearly," which represents the annual salary of each developer.

Data Preparation

The data preparation phase involves the following steps:

- Data Cleaning
- Data Preprocessing
- Exploratory Data Analysis
- Feature Engineering

1. Data Cleaning

The following steps will be performed during the data cleaning phase:

- **Handling missing values**: Dealing with missing values in the dataset by applying appropriate techniques such as imputation or removal, depending on the nature and significance of the missing data.
- **Addressing duplicate rows**: Identifying and removing any duplicate rows in the dataset to avoid bias and maintain data integrity.
- **Correcting column names**: Renaming columns with clumsy or unclear names to improve understanding and readability.

2. Data Preprocessing

The following preprocessing steps were applied:

- **Feature scaling**: Normalising numeric features to ensure that they are on a similar scale, preventing certain features from dominating the model.
- **Encoding categorical variables**: Converting categorical variables into a numerical representation suitable for modelling. The encoding techniques used were:
    1. Binary encoding. This was done to reduce the number of columns added to the dataset, when compared to one-hot encoding. It was performed on features that showed no inherent order among them.
    2. Ordinal encoding. This was performed on the features that showed an inherent order in them.
    3. Label encoding. This was performed on the target variable, “Annual Salary”
- **Handling outliers**: Identifying and addressing any outliers in the dataset that might significantly impact the model's performance.

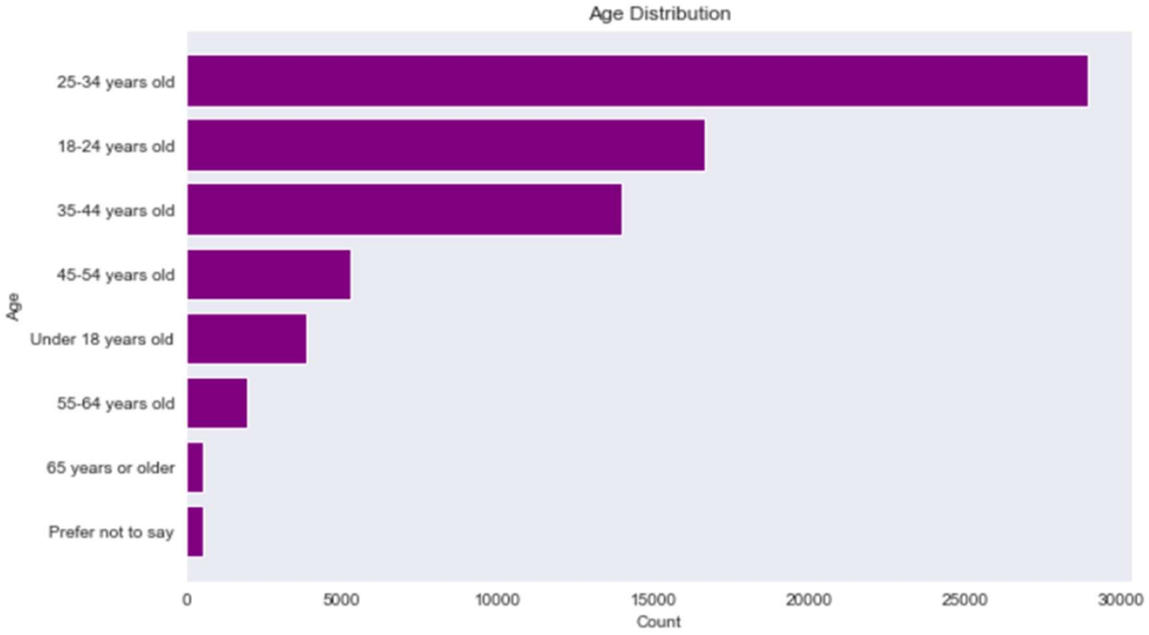
3. Exploratory Data Analysis

Exploratory Data Analysis (EDA) aims to gain insights into the dataset and understand the relationships between variables. The following techniques will be employed:

- Univariate analysis: Examining the distribution and statistics of individual variables to identify patterns, outliers, or data quality issues.
- Bivariate analysis: Analysing the relationships between pairs of variables to uncover correlations, dependencies, or potential predictive power.
- Multivariate analysis: Investigating interactions and dependencies among multiple variables using techniques such as correlation matrices, heatmaps, or scatter plots.

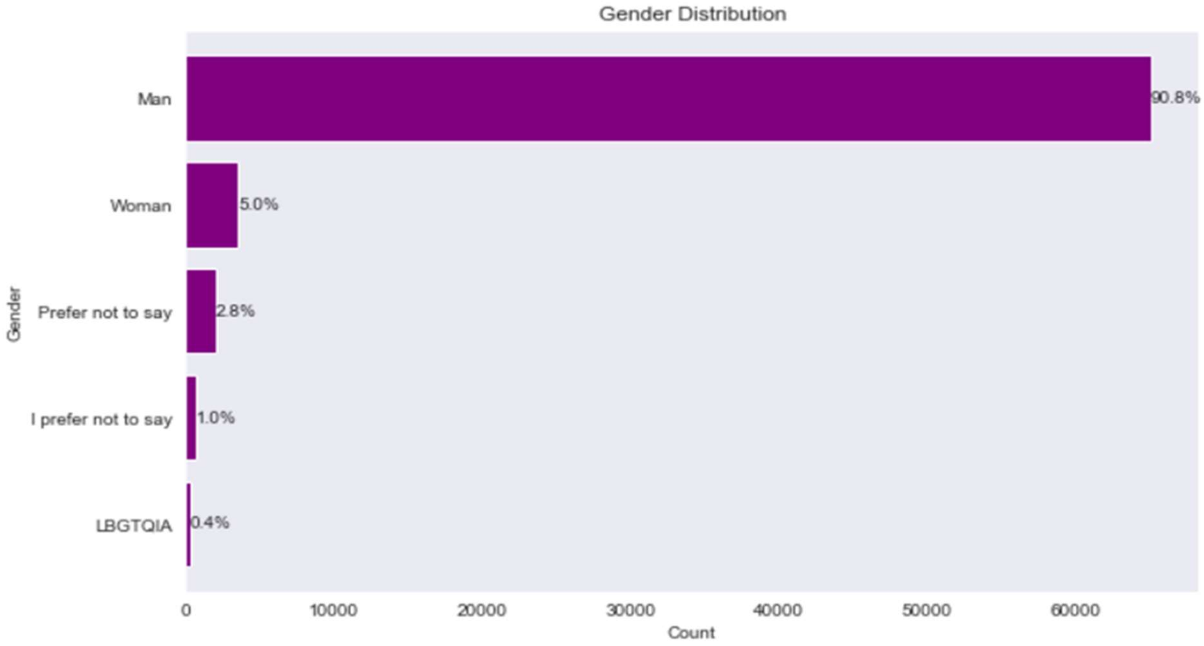
The following insights were obtained from the dataset:

1. **Distribution of respondents by age:**



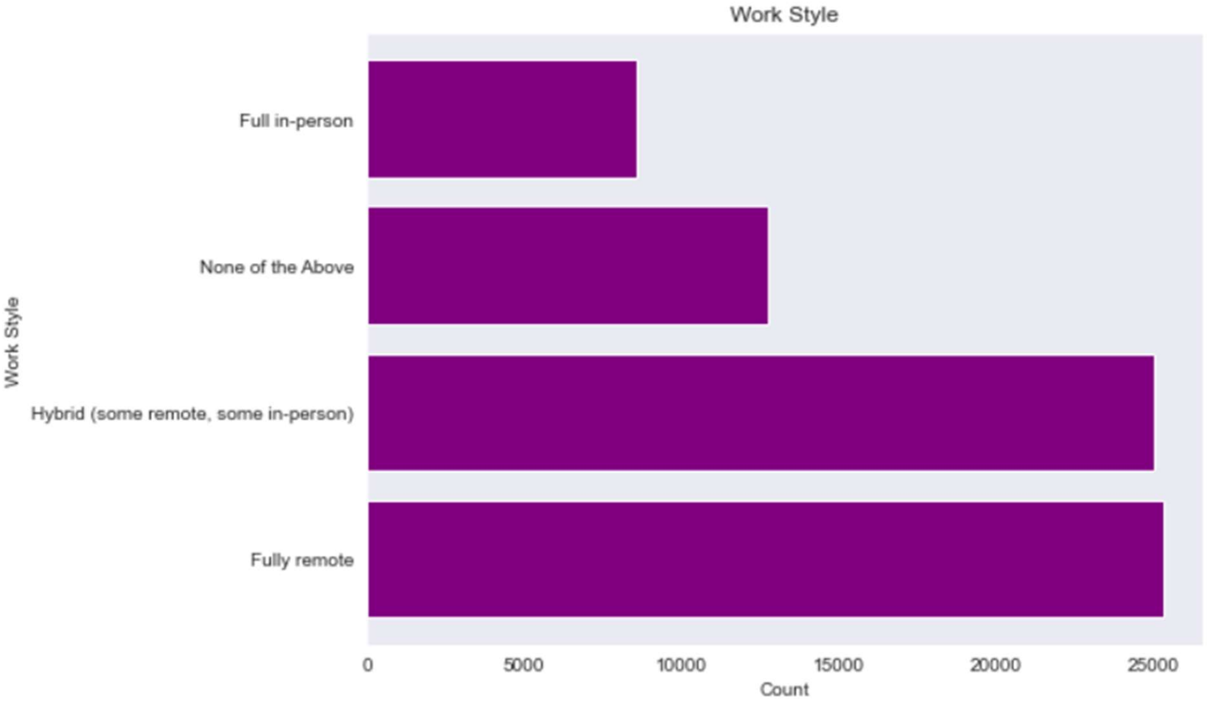
This showed that most of the respondents were below 35 years of age.

2. **Distribution of respondents by gender:**



This showed that the majority of respondents were male, giving a general overview of the gender distribution in the tech industry.

3. **Distribution of respondents by work style:**



This showed that most tech jobs are performed either fully remotely or in a hybrid manner.

4. **Feature Engineering**

The following techniques were considered:

- **Creating new features**:

A new feature, “**continents**” was created from the ‘Country’ column. This was **used to categorize the salaries** during the data cleaning phase.

The null values in the salary column were replaced with the median salary according to the continents where the respondents were from.

This ensured that the values were filled **accounting for the salary imbalance between different regions.**

- **Renaming the columns:**

The columns in the dataset were renamed to names that accurately described them.

- **Dimensionality reduction:**

Employing techniques like Principal Component Analysis (PCA) to reduce the number of features while preserving relevant information and minimising multicollinearity.

Description of features selected for prediction of salaries

Feature	Description
Code Certifications	This describes the certifications that the developer has. The categories include: ● "Udemy" ● "Coursera" ● "Codeacademy" ● "Pluralsight" ● "Edex" ● "Udacity" ● "Skillsoft" ● "Other" ● "None of the Above " - (Describes users with no additional certifications.)
Education Level	This describes the highest level of education the developer has attained. The categories include: ● Bachelor's degree ● Master's degree ● Some college/university study without earning a degree ● Secondary school

	● Associate degree ● Other doctoral degrees (Ph.D., Ed.D., etc.) ● Primary/elementary school ● Something else ● Professional degree
Employment status	This describes the current status of employment of the developer. The options include: ● Employed, full time. ● Employed, part time. ● Student, full time ● Student, part time ● Independent contractor/freelancer/self employed ● Not employed, looking for work ● Not employed and not looking for work ● Retired ● I prefer not to say.
Influence	This category describes the level of influence that the developer has in their organization. It is used as an indication of their rank in the organization that they work for. The options include: ● I have little or no influence. ● I have some influence. ● I have a great deal of influence
Country	This is the country that the developer is from. It includes **180 countries.**
Professional Experience	This is a numerical column indicating the **years of experience** that the developer has.
Work Style (Remote vs	This describes the mode of work. The categories include:

Onsite)	● Fully remote ● Hybrid ● Full in person ● None of the above.
Developer Type	Describes the type of the developer. It has 30 categories and examples include: “Front end developer”, “Back-end developer”, “data engineer”, etc
Coding language	Contains a list of 42 coding languages. Examples include: “Java”, “R”, “Python” etc
Developer Description	This category describes the level of skill of the developer. The Categories include: ● I am a developer by profession. ● I am learning to code. ● I am not primarily a developer, but I write code sometimes as part of my work. ● I code primarily as a hobby. ● I used to be a developer by profession, but no longer am
Annual Salary	This is a description of the annual salary of the developer. They were categorized into 7 different bands: ● On average $5,000 ● $5,000 - $15,000 ● $15,000 - $30,000 ● $30,000 - $50,000 ● $50,000 - $100,000 ● $100,000-$200,000 ● Over $200,000

5. Modelling

In the modelling phase, we aimed to develop a salary prediction model using various algorithms.

Initially, we attempted to use **regression algorithms**, but due to the target variable not fitting well, we decided to change the problem statement to **a classification problem.**
The goal was to predict the salary range based on the developer's skills.

This approach was deemed viable as salaries vary depending on different factors such as organization size and their nature. This meant that one could earn different salaries in different organizations' with the same skill set.

Providing a salary range is therefore a more accurate mode of approach.

We evaluated several classification algorithms, including Logistic Regression, K-Nearest Neighbours (KNN), Random Forest Classifier, and XG Boost.

I). Logistic Regression

Logistic regression from **scikit-learn** is a supervised learning algorithm which in this project has been used for multiclass classification of salaries. It predicts the probability of each class using a logistic function and selects the class with the highest probability.

We started with Logistic Regression and evaluated its performance. The initial results were as follows:

Training scores:

- Precision Score: 0.440
- Recall Score: 0.442
- Accuracy Score: 0.442
- F1 Score: 0.418

Test scores:

- Precision Score: 0.438
- Recall Score: 0.440
- Accuracy Score: 0.440
- F1 Score: 0.416

After hyperparameter tuning, the updated results were:

Training scores:

- Precision Score: 0.442
- Recall Score: 0.446
- Accuracy Score: 0.446
- F1 Score: 0.429

Test scores:

- Precision Score: 0.445
- Recall Score: 0.448
- Accuracy Score: 0.448
- F1 Score: 0.431

II). K-Nearest Neighbours (KNN)

K-Nearest Neighbours (KNN) from scikit-learn is a supervised learning algorithm which in this project has been used for multiclass classification of salaries. It assigns a new data point to the class based on the majority vote of its K nearest neighbours in the feature space.

The K-Nearest Neighbours algorithm was applied and observed the following results:
Training scores:

- Precision Score: 0.724
- Recall Score: 0.724
- Accuracy Score: 0.724
- F1 Score: 0.720

Test scores:

- Precision Score: 0.585
- Recall Score: 0.592
- Accuracy Score: 0.592
- F1 Score: 0.585

After hyperparameter tuning, the results were:

Training scores:

- Precision Score: 0.980
- Recall Score: 0.980
- Accuracy Score: 0.980
- F1 Score: 0.980

Test scores:

- Precision Score: 0.758
- Recall Score: 0.766
- Accuracy Score: 0.766
- F1 Score: 0.759

III). Random Forest Classifier

Random Forest Classifier from **scikit-learn** is an ensemble learning algorithm which has been used for multiclass classification of salaries. It **combines multiple decision trees** to make predictions by **averaging the results** of individual trees.

We attempted Random Forest Classifier with default parameters and tuned hyperparameters. The initial and tuned results were as follows:

Training scores (default parameters):

- Precision Score: 0.984
- Recall Score: 0.984
- Accuracy Score: 0.984

- F1 Score: 0.984

Test scores (default parameters):

- Precision Score: 0.753
- Recall Score: 0.753
- Accuracy Score: 0.753
- F1 Score: 0.753

After hyperparameter tuning, the updated results were:

Training scores:

- Precision Score: 0.857
- Recall Score: 0.857
- Accuracy Score: 0.857
- F1 Score: 0.856

Test scores:

- Precision Score: 0.717
- Recall Score: 0.714
- Accuracy Score: 0.714
- F1 Score: 0.714

IV). XG Boost

XGBoost (Extreme Gradient Boosting) is an **ensemble learning algorithm** which has been used for multiclass classification of salaries. It combines the predictions of multiple weak decision tree models in a sequential manner, utilising gradient boosting to minimize the loss function and improve predictive accuracy.

Finally, we applied the XG Boost algorithm and obtained the following metrics:

Training scores:

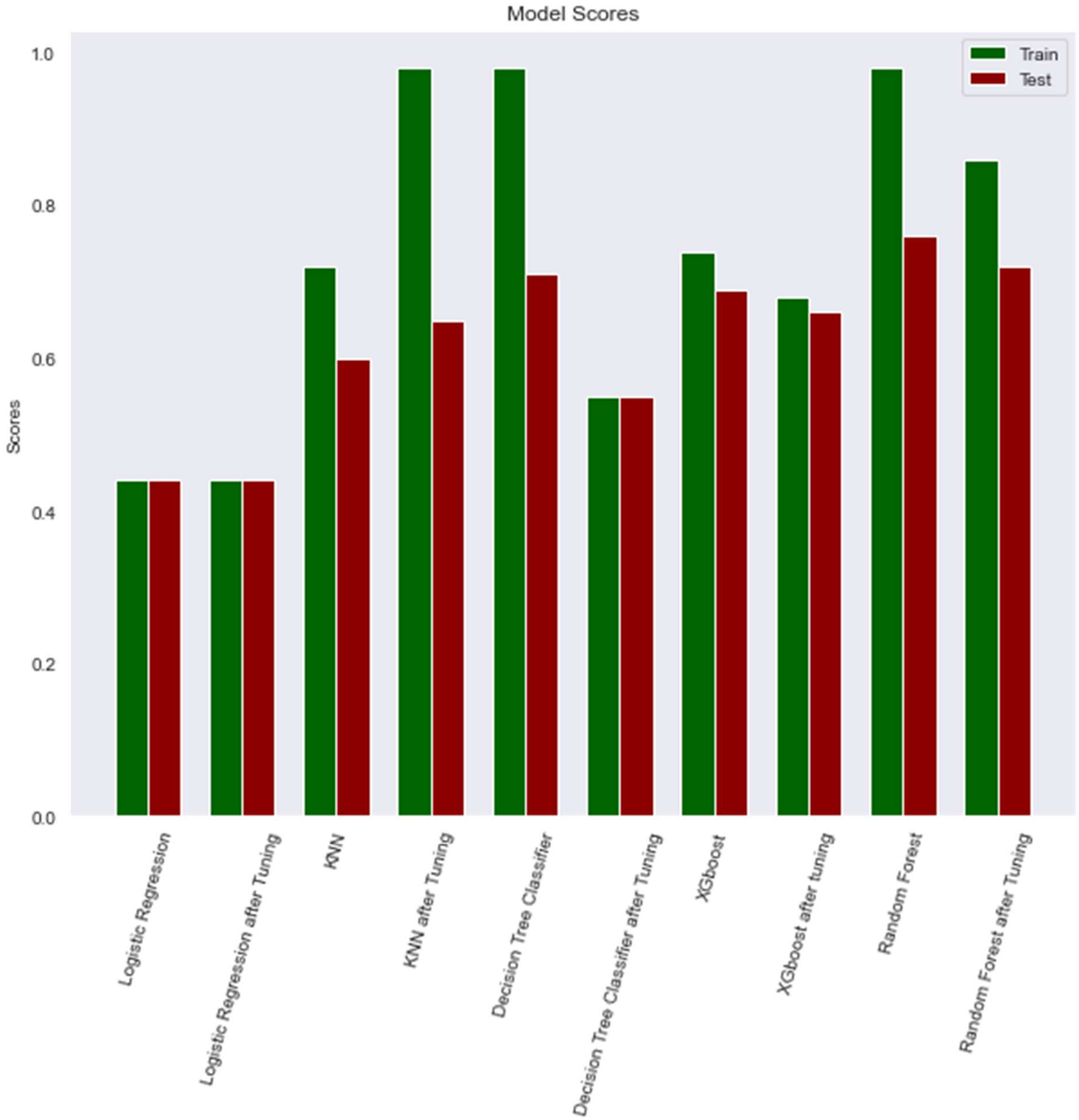
- Precision Score: 0.741
- Recall Score: 0.741
- Accuracy Score: 0.741
- F1 Score: 0.740

Test scores:

- Precision Score: 0.689
- Recall Score: 0.689
- Accuracy Score: 0.689
- F1 Score: 0.688

Based on the results, XG Boost showed the most promising performance, with the least gap between training and test metrics, indicating less overfitting. Therefore, it was considered the best candidate for hyperparameter tuning.

6. Model Evaluation



From analysis of the graph above, The tuned random forest model was selected. The scores for this model were:

Accuracy Score of 0.857 on the training set and 0.714 on the test set. This model met the success metrics of the project as it scored above 70% in accuracy.

7. Conclusion

As the tech industry in Africa continues to grow, it creates the need to have a reliable platform that can accurately predict salaries of developers based on their different skill sets and experience. This would assist job seekers to ask for competitive salaries, provide employers with an accurate source of data to use in their salary offers, and recruiters to accurately advise their clients on remuneration expectations.

This project utilizes machine learning models to predict these salaries.

The output of these models can be accessed through a web application by following this link

8. Limitations

1. **Imbalanced salary distribution:** Based on our visualizations, we witnessed a disproportionate number of individuals in certain salary ranges, this influenced biases towards predicting the majority class and made our model struggle to accurately predict the less-represented salary ranges.

2. **Overfitting**: Random Forest Classifier has the potential to overfit the training data due to its high model complexity, especially when there are many input features compared to the number of observations(high dimensionality). Overfitting resulted in high training accuracy but lower generalization performance on unseen data, leading to inaccurate salary range predictions.

3. For a student with zero experience the model would always suggest a range of 5,000 regardless of other variables, this would have been addressed if we had more data on students who were earning.

4. To avoid the increase in columns, thus increase in dimensionality caused by binary encoding, Categorical columns were assigned a unique numerical value to each category of a feature, disregarding any order it may have had. The numeric values can be misinterpreted by algorithms as having some sort of hierarchy/order in them, therefore, reducing the performance of the model.

9. Next Steps

Based on the analysis and results obtained from the modelling phase, the following steps can be taken to improve the project:

Further Refinement of the Random Forest Model:

- The Random Forest Model showed the most promising performance among the evaluated algorithms. To enhance the model's predictive power, we recommend conducting further refinement by fine-tuning the hyperparameters using techniques like grid search or random search. This can help optimize the model's performance and potentially improve its accuracy in predicting salary ranges.

Continuous Data Collection and Model Updating:

- To ensure the model remains relevant and accurate, it is recommended to continuously collect new data from developers in the industry. This will allow for regular updates and retraining of the model, incorporating the latest trends and changes in the tech job market. By keeping the model up to date, it can provide more accurate salary predictions and adapt to evolving industry dynamics.

Incorporate Additional Features:

- Consider expanding the feature set by incorporating additional relevant variables. Factors such as location, industry, years of experience, and educational background may influence salary ranges and should be considered in future iterations of the model. Gathering more comprehensive data and including a wider range of features can enhance the model's predictive capabilities and provide more meaningful insights.

User-Friendly Interface and Visualization:

- When deploying the model as an online application, focus on creating a user-friendly interface that allows job seekers, employers, and recruitment agencies to easily input their desired variables and receive the predicted salary range.

Additionally, consider incorporating visualizations to help users understand the factors influencing salary predictions and provide a clear overview of the distribution of salary ranges in different regions.

Collaboration with Local Tech Companies:

- To improve the accuracy and relevance of the salary predictions, consider collaborating with local tech companies in Africa. By partnering with companies and obtaining salary data directly from them, the model can be fine-tuned to the specific regional context, taking into account local salary trends and market dynamics.

By implementing these recommendations, the salary prediction model can continue to evolve and provide valuable insights to job seekers, employers, and recruitment agencies in the African tech industry.