**Detailed introduction of the project**

The project targeted to predict the salary of the employees based on certain factors. The problem statement is based on the pretext that while switching jobs, the employees certainly go forward with the idea of compensation growth, but how much percentage increase they should be expecting or asking should also be realistic enough. Many organizations put up the questions regarding the expected compensation. Consequently, lack of knowledge of their skills’ market value in the current scenario would prevent them in asking an unrealistic expected compensation growth, which can certainly discourage the human resources to move forward with a candidate that has a better clarity on their future prospects.

Also, there could be another situation, where a qualified candidate has multiple job offers and wants to solely make the decision based on the compensation structure. It again becomes crucial for the candidate to come up with the better decision-making temperament to not regret the decision later, after joining the company.

In both the above scenarios, Machine Learning could play a signification role in providing an interface to facilitate that decision-making platform to solve the problem in hand. Machine Learn model integrated on a user interface could run the regressive algorithms to provide an answer in the form of predicted salary figure of continuous nature based on the inputted aspects of the candidates, such as: job experience, job title, gender.

The project intended to use supervised algorithm as the labelled data has been collected in the initial phase of a Machine Learning model development life cycle. There will be input features [X] and output feature [y]. And the goal would be to train a Machine Learning model with such data beforehand. In the testing phase the test data with hidden output variable would be fed to the model. And by comparing the predicted output variable values with actual output variable values would yield the metrics score.

We are intended to utilize certain Regression Machine Learning algorithm candidates to predict the output variable. Comparing the metrics score of these algorithms on our model, we will ascertain the final algorithm to move forward with, to predict based on live user data.

**Data set description**

The initial raw data has been collected through distributing the Google form link to the potential users who are currently employed in various domains, such that to be able to train the model on data pertaining to wide variety of job titles and varied job experiences.

The form is directly configured to Google sheet to store the responses given by the audience. This sheet, is then directly downloaded to the data folder in the project structure created specific to store different files, folders and dependencies.

Google Form used to collect data from the different employees can be found at the below http server address:

<https://docs.google.com/forms/d/e/1FAIpQLSdHulNst5L1dgtQT12JHq3o1XlB0OTkPgXtnDWg4QQuXOtghg/viewform?vc=0&c=0&w=1>

**Input Features (Variables):**

* Experience
* Job Title
* Gender

**Output Feature (Variable):**

* Salary

**Nature of Features;**

* Experience is **continuous**
* Job Title is continuous at first but processed later to be captured into **categories**
* Gender is **categorical** (Male or Female)

**Propose a machine learning model**

Due to the possibility of the inclusion of outliers in salary which cannot be ignored/removed, it is very clear that such a machine learning algorithm is desired that can be least impacted by such anomalies. Hence, the best candidate that comes as an obvious answer is:

**Random Forest**

Initially, we have built the model using Random Forest. The metrics score has been recorded and subject to be used for comparison when we try out other models like Logistic Regression, etc.

**We have run our first phase of model development that includes:**

* Importing libraries
* Data loading
* Exploratory Data Analysis
* Processing and Cleaning of data
* Visualization of data based on various factors like:
  + Salary inequalities based on Gender, Job Title and Experience
* Dividing the data in hand into training and testing data using train test splitting mechanism
* Creating the instance of a model (Random Forest with n\_estimators = 100)
* Fitting the training data into the model
* Feeding the testing data into the model
* Recording the metrics like Mean Absolute Error, Mean Squared Error and Root Mean Squared Error

**Pending next phase work includes:**

Tuning the model to get possibly better metrics

Running other model candidates and comparing the metrics

In case a better model is observed, declaring it the final algorithm to be used in our model

**Pending Non-Machine Learning work includes:**

Creation of UI based on HTML and CSS (and possibly ReactJS)

Creation of Flask API to move data from UI to machine learning model to general prediction and send it back to be displayed on the UI application

Final run of the complete app on the local host

Pushing the code to a GitHub repository

Deploying the code to Heroku app server

Publishing the final http URL server specifications

**Preliminary results**

**Mean Absolute Error:** 0.12176136363636529

**Root Mean Squared Error:** 1.6153470285753906

**Code repository can be found at:** [**https://github.com/Simran-IBM/SalaryInsights**](https://github.com/Simran-IBM/SalaryInsights)