**Report Project named**

**“Salary Insights and Estimation”**

**Under the category “Software & IT”**

**By Group 12**

**Submitted**

**For**

**The fulfillment of the award of the**

**Post Graduate Diploma**

**in**

**Data Science**

**Submitted By:**

**Abin Thomas**

**Chhayank Datta**

**Prince Munjal**

**Simran Preet Singh Raina**

**Dated: 14th June, 2020 Sundeep Reddy Koluguri**

**1. Introduction to 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.

**2. Project problem**

As mentioned above in the introduction, the problem statement arises when a candidate, successful in his/her interview endeavor has to decide the deserving compensation structure based on his/her skills in the market. Also, the employer also needs to find out the appropriate salary structure to be offered to the candidate as per the experience and other factors of the successful candidate. If there could be a system, that can be accessed by employee and employer, where based on certain criteria/aspects of the candidate, salary insight can be estimated, it would be a solution to such problem.

**3. Key contributions**

**Dataset collection: Abin Thomas, Chhayank Datta**

**Front-end and back-end web app: Prince Munjal**

**Machine Learning model: All team members**

**4. Machine learning**

* + **Data set resource**

Data set in the form of excel sheet (xlsx) file was utilized wherein data pertaining to different features/criteria were collected from the peers. Also, due to the insufficient number of entries from the peers, a bigger data set was utilized later, by downloading it from the below Kaggle site link:

<https://www.kaggle.com/filco306/salaries-across-industries-crowdsourced/metadata>

It helped in providing a big chunk of data that can provide better salary estimates as disperse data was on our disposal.

* + **Data set pre-processing**

Preprocessing is an integral part of any machine learning lifecycle, wherein data is exported and cleaned. Missing values and outliers are cured. And

Pre-processing was done in the following manner:

* + - Certain redundant or futile feature columns were removed.
    - Job profile categories were created, and using the keywords that could possibly be present in the dataset for job designations, these were put under the appropriate and relevant categories. This was done to give a generic job profiles for number of job designations that could be very diverse. Such diversity if not contained in certain meaningful categories would give rise to uncontrolled estimates.
    - Years of experience was a given a non-fractional number of as certain years were in fractions.
    - Various data analysis aspects were performed based on:
      * Gender Vs Salary
      * Salary with respect to Gender, Experience, Job Profile
    - Later for standardization purpose, based on experience; salaries were normalized, removing the gender discrepancies and helping to easily evaluate machine learning model outcomes.
    - Label encoding for the feature values that can be fed into a regression ML model.
  + **Finding the best machine learning model for the problem objective**

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**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Initially, we have built the model using Random Forest and the metrics score has been recorded with initial values for n\_estimators as 100 random statue value as 0.

**Following Machine Learning lifecycle has been executed:**

* + - 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
    - 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

**Below Software Engineering work implemented:**

* + - Creation of UI based on 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 web application on AWS.
    - Publishing the final http URL server specifications
  + **Training and testing the model**

For training and testing purposes, train\_test\_split library is used.

Training and testing ratio being: 80:20

* + **Model Evaluation**

Model evaluation is done based on the difference (or error) between predicted and actual salaries. Model evaluation is calculated based on the below parameters:

**Mean Absolute Error**

**Mean Squared Error**

**Root Mean Squared Error**

Preliminary results:

**Mean Absolute Error:** 0.12176136363636529

**Root Mean Squared Error:** 1.6153470285753906

**5. Analysis of results**

After deploying the model on the servers and validating the salaries predicted or estimated by the model, it was clear that there is overfitting issue in the ML model. The model is very generalized as per the training and testing data for the regression data set.

To **avoid overfitting** a **regression model**, we drew a random sample that is large enough to handle all of the terms that we could expect to include in our **model**. This process requires that we investigate similar studies before we collect data. Hence, the data was again drawn randomly (random sample) from the population sample. And it solved the issue of overfitting.

**Code repository can be found for:**

**ML model:** [**https://github.com/Simran-IBM/SalaryInsights**](https://github.com/Simran-IBM/SalaryInsights)

**Web application Back End:** [**https://github.com/princemunjal/api-salary-insights**](https://github.com/princemunjal/api-salary-insights)

**Web application Front End:** [**https://github.com/princemunjal/web-salary-insights**](https://github.com/princemunjal/web-salary-insights)

**App URL:** [**https://master.d19zdppwuiif9i.amplifyapp.com/**](https://master.d19zdppwuiif9i.amplifyapp.com/)