|  |  |
| --- | --- |
| Internship Project Title | TCS iON RIO-125: HR Salary Dashboard - Train the Dataset and Predict Salary |
| Name of the Company | TCS iON |
| Name of the Industry Mentor | Debashis Roy |
| Name of the Institute | ICT Academy of Kerala |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Start Date | End Date | | Total Effort (hrs.) | | Project Environment | Tools used |
| 17-04-2023 | 16-07-2023 | | 125 | | Google colab | Python 3 |
| Milestone # | 3 | Milestone: | | Create data set, Clean and sanitize dataset, Preprocessing data set, Test and train the dataset, Build the classifier models and fit the data in the model, Dash board creation. | | |

**TABLE OF CONTENTS**

|  |
| --- |
| **Acknowledgement** |
| 1. **Objective** |
| 1. **Introduction** |
| 1. **Internship Activities**    1. Creating the dataset    2. Analyzing the dataset    3. Cleaning and sanitizing    4. Splitting the data set    5. Scaling    6. Modeling    7. Fine tuning 2. **Result & Conclusion** 3. **Enhancement scope** |
| 1. **Link to code and executable file** |

**ACKNOWLEDGEMENTS**

I am conveying my sincere gratitude towards my industry mentor, Debashis Roy, for helping me throughout this project till now and providing me this wonderful platform to complete this project. I am also thankful for answering my queries at every phase of the project. I also want to thank all my friends who helped me with valuable suggestions during this project.

**1. OBJECTIVE**

The objective of this project is to build a salary prediction dashboard for human resource management. This dashboard will use machine learning algorithms to predict the salary of job candidates based on factors like their experience, age, and qualifications. This information can help HR managers to make better decisions when hiring candidates for job positions.

**2. INTRODUCTION**

The human resources department often has a lot of job applicants to process and must choose the best candidates for each job. Candidates often consider salary when deciding whether to accept a job offer, so it's important for HR to offer competitive salaries.

In this project, we will use a dataset that includes information on over 32,000 job candidates, such as their experience and salary. This dataset is good for our analysis because it has a wide range of job profiles and salaries. We will use this data to build a salary prediction dashboard to help HR managers make better decisions about salaries for job candidates.

In the dataset I have chosen, the target salary only has two categories: less than or equal to 50K and greater than 50K. Therefore, the model needs to predict which of these two classes the salary belongs to. This means that our model will be a binary classification model. There are several methods we can use for binary classification, such as SVM, logistic regression, random forest, etc. I have trained and tested my data using Logistic Regression, KNN, Decision tree, SVM and Random Forest compared them to select the best model.

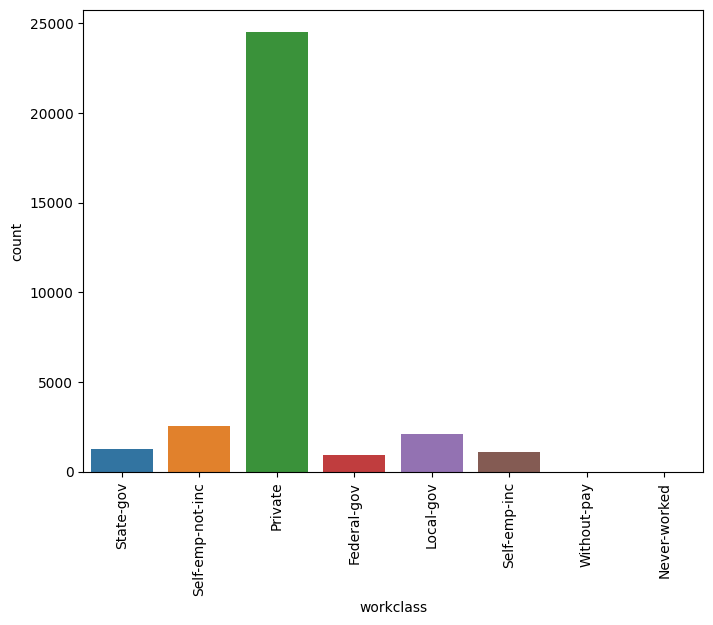
**3. INTERNSHIP ACTIVITIES**

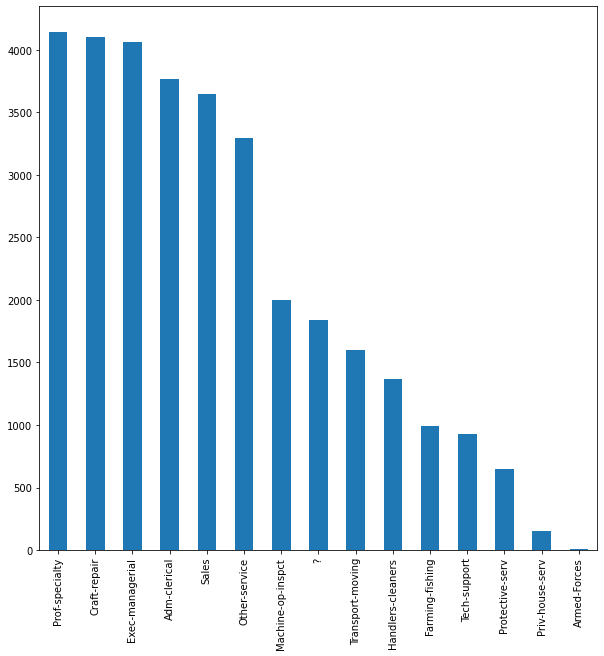
**3.1 Creating the dataset**

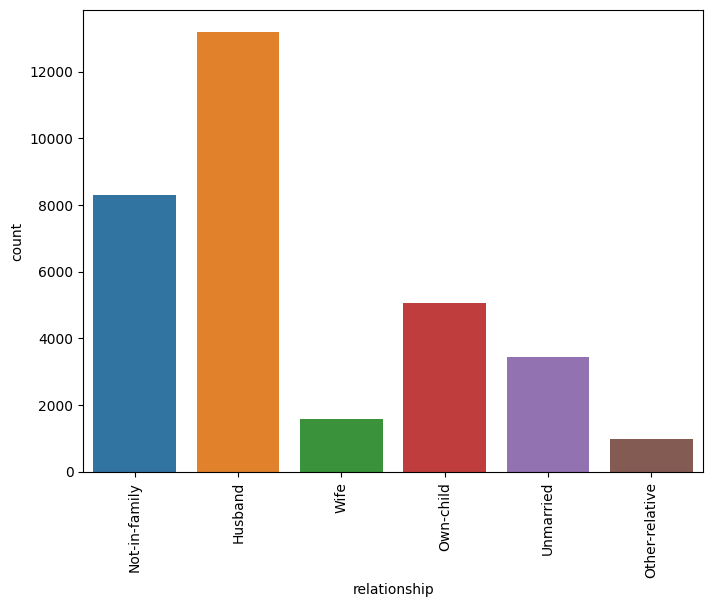
For this project, a dataset with 32562 rows and 14 columns has been selected. The columns in the dataset include Age, Work Class, Education, Education Number, Marital Status, Occupation, Relationship, Race, Sex, Capital Gain, Capital Loss, Hours per Week, Native Country, and Salary.

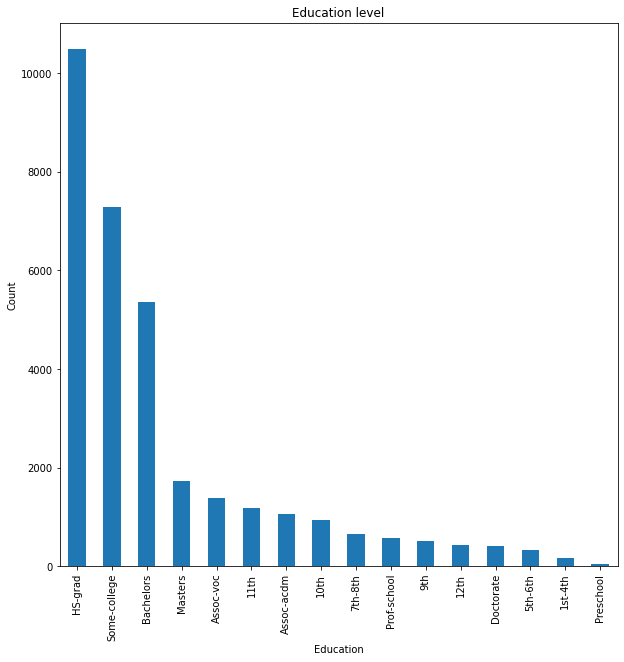
**3.2 Analyzing the dataset**

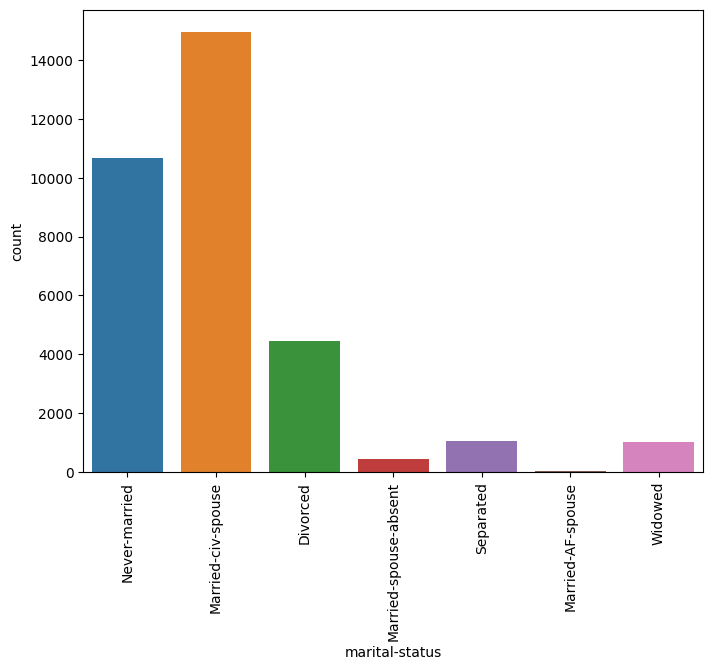
During the analysis of the dataset, it was observed that the columns "Capital-loss" and "Capital-gain" do not have any significant role in predicting the salary. It was also discovered that these columns only contain values when the individual has a salary. Therefore, these columns can be dropped from the dataset. Similarly, the column "Education number" can also be dropped as it is a numerical representation of the "Education" column. Exploratory data analysis was performed on various features of the dataset. The results of this analysis are presented below.

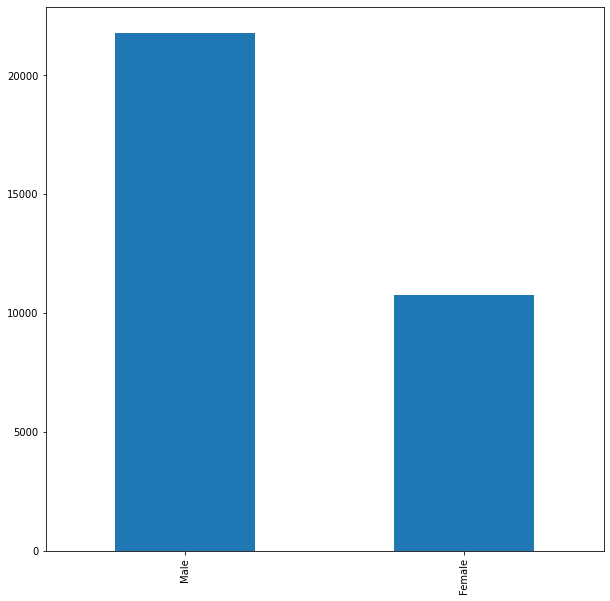












**3.3 Cleaning and sanitize the dataset**

In the data set, no null values are present, but ‘?’ present in some columns and are replace with respective mode of each column. Then checked for the outliers before processing, and in our data, outliers are found in some of the columns like age and hours-per-week and are removed. The categorical data has been converted to a numerical form in this step using label encoding.

**3.4 Splitting the data**

The dataset has been split into two parts before scaling the data.

**3.5 Scaling of data**

To improve the performance of many machine learning algorithms, it is often beneficial to scale numerical input variables to a standard range. There are two widely used techniques for scaling numerical data prior to modeling: normalization and standardization.

**Normalization** scales each input variable separately to the range of 0-1, which is the range for floating-point values where we have the most precision.

**Standardization** scales each input variable separately by centering the data (i.e., subtracting the mean) and then dividing by the standard deviation. This shifts the distribution of the data to have a mean of zero and a standard deviation of one. In this dataset, standardization was used as the scaling technique.

**3.6 Modeling**

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data. A classification model tries to draw some conclusions from the input values given for training. It will predict the class labels/categories for the new data.

**A. Logistic regression classifier**

Logistic regression is a machine learning algorithm used for classification tasks. It models the probabilities of possible outcomes of a single trial using a logistic function, making it useful for understanding how independent variables influence a single outcome variable. However, logistic regression only works when the predicted variable is binary, assumes that all predictors are independent of each other, and requires the data to be free of missing values.

**B. K-Nearest Neighbours**

**Neighbors-based classification is a type of machine learning algorithm that falls under the category of lazy learning. It does not attempt to construct an internal model but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbors of each point. This algorithm is simple to implement and robust to noisy training data, making it effective when the training data is large. However, determining the value of k can be challenging, and the computation cost is high as it needs to compute the distance of each instance to all the training samples.**

**C. Decision Tree classifier**

A decision tree is a machine learning algorithm that generates a sequence of rules based on the input data's attributes and corresponding classes. It is a popular algorithm because it is easy to understand and visualize, requires minimal data preparation, and can handle both numerical and categorical data. However, decision trees have some limitations. They can create overly complex trees that may not generalize well to new data. Additionally, small variations in the data can result in completely different decision trees being generated, making them unstable.

**D.** [**Support vector machine**](https://analyticsindiamag.com/understanding-the-basics-of-svm-with-example-and-python-implementation/)

The text describes a Support Vector Machine (SVM) algorithm, which is a machine learning algorithm used for classification and regression analysis. SVM represents the training data as points in space, separating them into categories with a clear gap that is as wide as possible. This approach is effective in high dimensional spaces and is memory efficient as it uses a subset of training points in the decision function. When presented with new examples, the algorithm maps them into the same space and predicts the category they belong to based on which side of the gap they fall on. However, the algorithm does not directly provide probability estimates, which require an expensive five-fold cross-validation to calculate.

**E. Random forest classifier**

The Random Forest classifier is a meta-estimator that fits multiple decision trees on various sub-samples of datasets and uses an average to improve the predictive accuracy of the model while controlling over-fitting. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacement. This reduces over-fitting, and in most cases, Random Forest classifiers are more accurate than decision trees.

However, one disadvantage of the Random Forest classifier is that it may not be suitable for real-time prediction, as it can be computationally expensive to build the forest and classify new data. Additionally, implementing a Random Forest classifier may be more difficult than implementing simpler models due to the complexity of the algorithm and its many hyper parameters that need to be tuned.

Five different models are tried and compared the results.

|  |  |
| --- | --- |
| Model | Accuracy |
| Logistic Regression | 75.84 |
| KNN | 79.59 |
| Decision Tree | 76.14 |
| SVM | 79.13 |
| Random Forest | 79.74 |

When we compared the results, it is clear that Random Forest have better accuracy than other classifiers. So, we just used hyper parameter tuning to check whether the performance is improved or not.

**3.7 Fine Tuning**

Fine tuning machine learning predictive models is crucial step to improve accuracy of the forecasted results. Most performance variation can be attributed to just a few hyper parameters; the tunability of an algorithm, hyper parameter, or interacting hyper parameters is a measure of how much performance can be gained by tuning it. In machine learning, a hyper parameter is a parameter whose value is used to control the learning process. The values of other parameters are derived via training. Here I have done fine tuning of Random forest modeling and got accuracy above 82%. Hence the accuracy is improved and decided to use Random Forest classifier as model for HR salary prediction.

**4. RESULTS & CONCLUSION**

The dataset for HR salary prediction has 14 features, but only 11 of them were found to be relevant for making accurate predictions. After experimenting with various machine learning models, it was observed that Random Forest Classifier performed the best with the highest accuracy. The Random Forest model was fine-tuned to further improve its performance, and it was ultimately chosen as the most suitable model for this particular dataset. A website is hosted using these features to predict salary.

**5. ENHANCEMENT SCOPE**

This industry project has a broad scope. It involves predicting the salary of an individual using their resume or CV. Some of the Natural language processing techniques will help in developing this application.

**6. LINK TO CODE AND EXECUTABLE FILE**

* Link to the code:

[https://colab.research.google.com/drive/17Fwo6c4L1TANeOpr2696UqNn2nvVzRcH#scrollTo=DoGGIVD7dFd\_](https://colab.research.google.com/drive/17Fwo6c4L1TANeOpr2696UqNn2nvVzRcH%23scrollTo=DoGGIVD7dFd_)

* Executable file:

<https://github.com/Zandubai/TCS_iON-RIO-125.git>