Internship Project Title	RIO-125: HR Salary Dashboard - Train the Dataset and Predict Salary
Name of the Company	TCS iON
Name of the Industry Mentor	Rushikesh Meharwade
Name of the Institute	Mepco Schlenk Engineering College Sivakasi TamilNadu

Start Date	End Date	Total Effort (hrs.)	Project Environment	Tools used
05/09/2021	06/12/2021	130	Jupyter Notebook	Python 3

TABLE OF CONTENT

ACKNOWLEDGMENTS

I am conveying my sincere gratitude towards my industry mentor, Rushikesh Meharwade, and academic mentor, Dr. Muthu Kumar 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.

OBJECTIVE

The objective of this model is to make a salary prediction dashboard for human resource management. The model should be able to predict the salary of the person by inputting his details.

INTRODUCTION

The dataset that I have selected for the project and training of the model contains attributes age, workclass, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country attributes, and target salary. Throughout my entire process of the industry project, I overcame the milestones that have been defined by TCS iON. During the period of completing the first milestone, I cleaned and sanitized the dataset and had done some EDA. The next milestone was on training a model for predicting the salary of a person. For this, I went through some articles and videos to understand some of the concepts on model training and also for understanding various classification techniques. Then I developed some models for the second milestone. The models were on Logistic regression and Random forest. By the end of the second milestone, I tuned the logistic regression and predicted salary using a tuple of user-defined input. I have generated classification reports for the models I have used. At the end of the project, I have used Support Vector Machine(SVM) as another classification model and tuned it also.

The model that I have developed predicts the Salary and helps in creating a dashboard. the main objective of this model is to predict the salary of a person by providing the necessary details.

INTERNSHIP ACTIVITIES

- Completed the RIO – pre-assessment test.
- Gone through the project reference materials that have been available such as the welcome kit, day-wise plan, project reference material, etc.

- Watched the webinars and recorded lectures.
- Created a dataset that is suitable for this project.
- Cleaned and sanitized the dataset.
- Gone through many articles and videos to learn about classification models and training techniques.
- Trained the dataset to predict the salary for an HR by providing details of that particular person.
- Compared 3 different classification techniques i.e., SVM, logistic regression, random forest by using the scores and classification reports.
- Wrote activity reports and project interim reports.

APPROACH / METHODOLOGY

The approach I took for the internship project for completing the milestones is understanding the concepts of the requirements. Reading articles and watching videos helped in achieving knowledge about the requirements. Jupyter Notebook has been used for doing the programming. Google colab has also been used for much faster execution. A GitHub account has been created for publishing the codes.

The target salary in the dataset I have selected contains only two classes (<=50K and >50K). Hence, the model needs to predict the salary to be in one of these two classes. So, our model converges to a binary classification model. There are several methods to make a binary classification, which include SVM, logistic regression, random forest, etc. So, I have trained and tested my data using Logistic Regression, Random Forest, and SVM, and compared them.

ASSUMPTIONS

There was some errors or misvalues in the dataset. i.e. some troubles that incurred in the cleaning process. So I assumed this value ‘?’ to be irrelevant for the further process. Thus, I deleted all these rows that had ‘?’ value. But before deleting I checked how many are there containing this value. Then only I proceeded for deletion of the rows. Another issue was on the two columns ‘capital loss’ and ‘capital gain’. When I checked what are these attributes for an employee, I came to know that these columns incur only when that particular person has a salary. therefore, I removed both these columns. I assume that the ‘?’ value and the columns ‘capital loss’ and ‘capital gain’ do not affect the target classification.

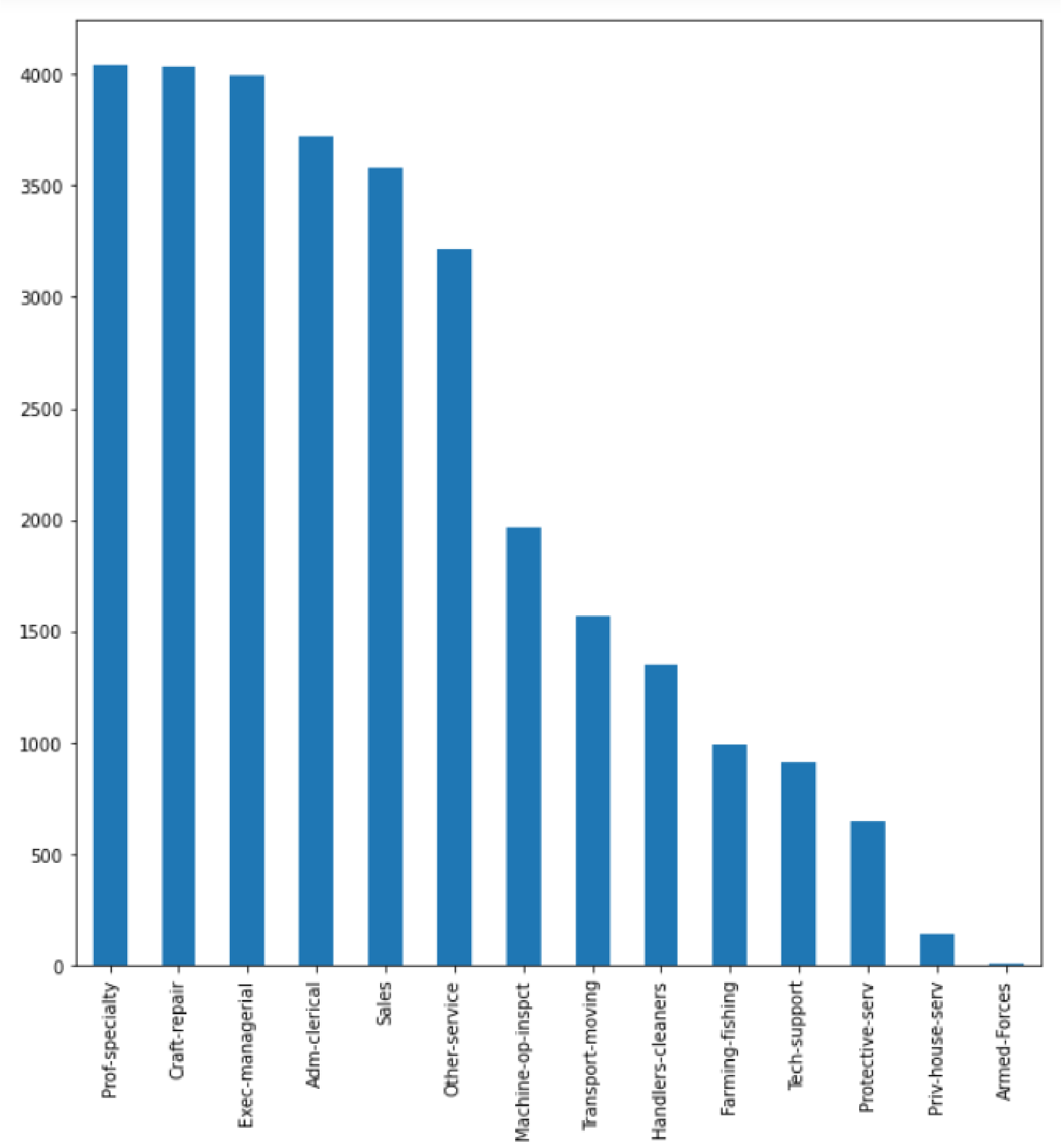
EXCEPTIONS

The exception incurred during the classification is that this project cannot predict an exact salary for an employee. The project would only classify his / her salary to be in one of the classes '>50K' or '<=50K'. Also, the dataset doesn't have data about the work experience of employees.

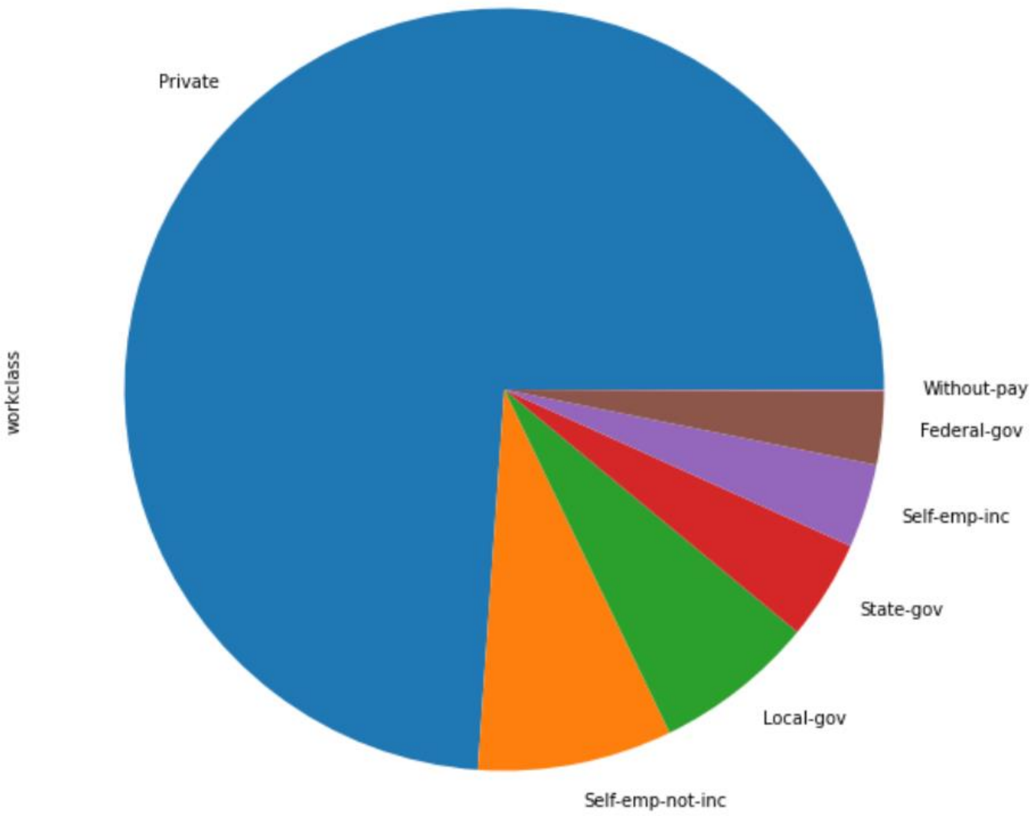
CHARTS, TABLES, DIAGRAMS

The following are the charts and diagrams that I have created as part of the EDA and Visualization.

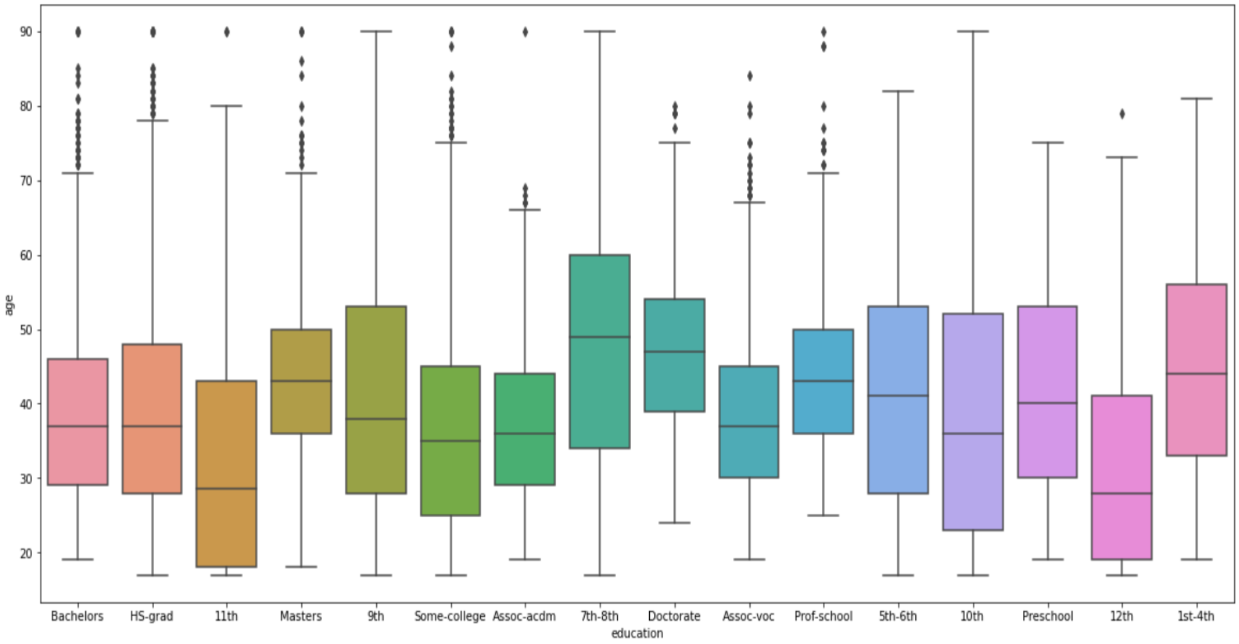
- This is a bar graph used for checking the variable 'occupation' in the dataset.



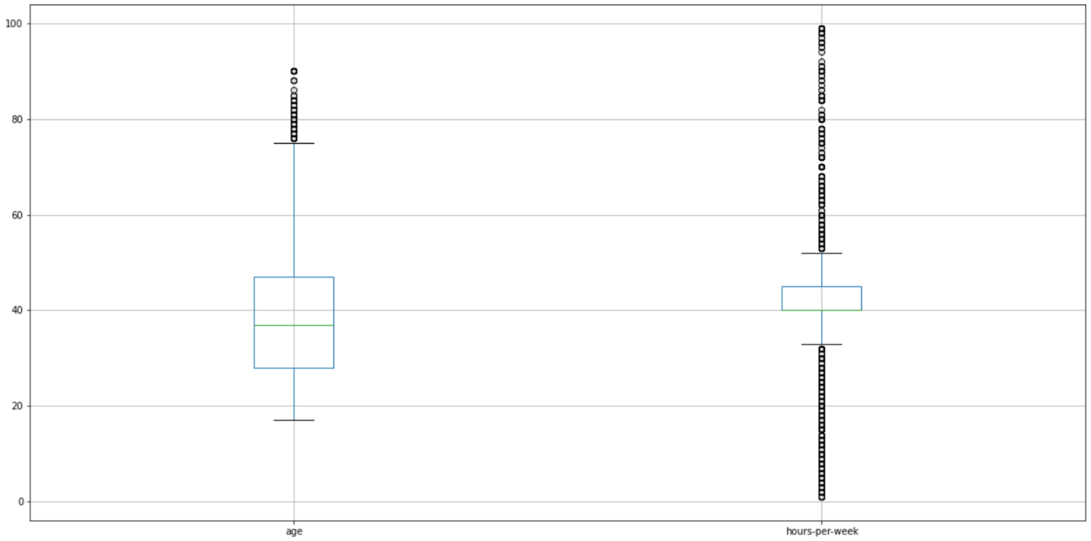
- This is a pie chart used for checking the variable ‘workclass’ in the dataset.



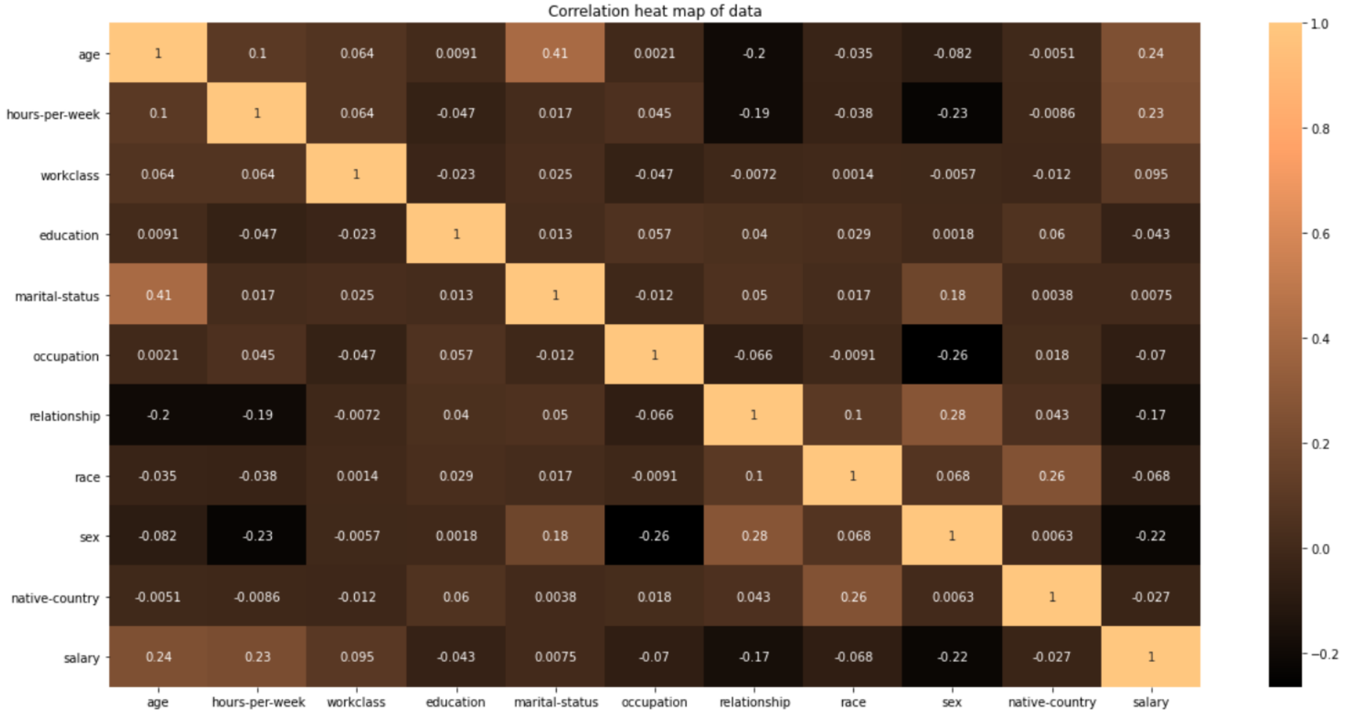
- This is a Box Plot used for checking the variables ‘education’ and ‘age’ in the dataset.



- This is a Box plot used for understanding the summary of the numeric variables ‘age’ and ‘hours-per-week’ variables.



- This is a correlation heatmap used for understanding the relationship between variables. this is done after converting all the categorical variables to numeric.



	age	hours-per-week	workclass	education	marital-status	occupation	relationship	race	sex	native-country	salary
age	1	0.1	0.064	0.0091	0.41	0.0021	-0.2	-0.035	-0.082	-0.0051	0.24
hours-per-week	0.1	1	0.064	-0.047	0.017	0.045	-0.19	-0.038	-0.23	-0.0086	0.23
workclass	0.064	0.064	1	-0.023	0.025	-0.047	-0.0072	0.0014	-0.0057	-0.012	0.095
education	0.0091	-0.047	-0.023	1	0.013	0.057	0.04	0.029	0.0018	0.06	-0.043
marital-status	0.41	0.017	0.025	0.013	1	-0.012	0.05	0.017	0.18	0.0038	0.0075
occupation	0.0021	0.045	-0.047	0.057	-0.012	1	-0.066	-0.0091	-0.26	0.018	-0.07
relationship	-0.2	-0.19	-0.0072	0.04	0.05	-0.066	1	0.1	0.28	0.043	-0.17
race	-0.035	-0.038	0.0014	0.029	0.017	-0.0091	0.1	1	0.068	0.26	-0.068
sex	-0.082	-0.23	-0.0057	0.0018	0.18	-0.26	0.28	0.068	1	0.0063	-0.22
native-country	-0.0051	-0.0086	-0.012	0.06	0.0038	0.018	0.043	0.26	0.0063	1	-0.027
salary	0.24	0.23	0.095	-0.043	0.0075	-0.07	-0.17	-0.068	-0.22	-0.027	1

- This is a table showing some of the synapsis in the classification reports of all three classifiers. these values are from the classification reports that were generated before the tuning process.

	Precision		Recall		F1 Score		Accuracy
	<=50K	>50K	<=50K	>50K	<=50K	>50K	
Logistic Regression	**0.79**	**0.56**	**0.94**	**0.23**	**0.86**	**0.33**	**0.76**
SVM	**0.84**	**0.68**	**0.93**	**0.45**	**0.88**	**0.54**	**0.81**
Random Forest	**0.84**	**0.65**	**0.9**	**0.52**	**0.87**	**0.58**	**0.8**

ALGORITHMS

The Algorithm used in the development of all the classifiers is as follows:

1. Start
2. Importing necessary libraries.
3. Vectorize character values.
4. Normalizing all the values.
5. Splitting the dataset to training and testing data.
6. Hyper-parameter tuning with the corresponding classification method.
7. Training the model.
8. Testing the model using test data.
9. End

CHALLENGES & OPPORTUNITIES

During this industry project, the challenges that I had faced was on visualizing each attribute’s relationships and developing a model for the classification. I only had a rough idea of these classification techniques from my academic background. But these daily activities made me much more knowledgeable on these techniques. So, I had the opportunities to build a good and conceptual knowledge of these models. Also, I learned about hyper-parameter tuning which I wasn’t aware of it earlier.

RISK Vs REWARD

In order, to predict a class according to the attributes that I had was not enough. This is because no previous work experience or any project experience was not available. Therefore, I had to use these available attributes to do the classification process.

REFLECTIONS ON THE INTERNSHIP

It was my first internship that I had done in my academic career. So, everything was new to me. the digital discussion room helped in connecting various people who are from different backgrounds and cultures. This helped me to develop a systematic approach to doing the project. The activity reports and interim reports helped me to analyze my process and doings. This helped in refurbishing some of the concepts throughout the project.

RECOMMENDATIONS

I felt that much more resources regarding the project can be made available in the project reference part. the reference materials were a bit small. I had to do some research out of the given project references.

OUTCOME / CONCLUSION

By looking back from the beginning stage of the project, I have cleaned and sanitized the data, i.e. data has been preprocessed. A logistic regression model has been trained and tested at the end of the milestone. A random forest classifier has also been implemented to understand the difference between certain models. Classification reports have been generated for both models. The parameters of the logistic regression model have been tuned for showing better performance. Also, for the tuned model, a classification report has been generated. After the second milestone, I started working on the support vector machine part and trained it for the classification part.

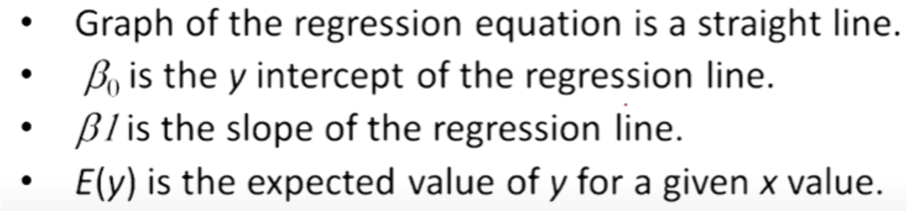
I was able to enhance my programming skills and even developed some methods to clean and identify errors from a dataset. I came to know more about various visualization libraries, classification models, and how to implement these classification models using python.

From these days of learning and implementing, I have grasped much knowledge about the following:

Linear Regression

Linear regression is perhaps one of the most well-understood and well-known algorithms in statistics and machine learning. Linear regression was borrowed from statistics to machine learning, which makes a statistical algorithm and machine learning algorithm. It is a linear model that assumes a linear relationship between the input variables (x) and the single output variable (y). The equation that describes how y is related to x and an error term is called the regression model. The simple linear regression model is

$$E(y) = \beta_0 + \beta_1 x$$



- Graph of the regression equation is a straight line.
- β_0 is the *y* intercept of the regression line.
- $\beta 1$ is the slope of the regression line.
- *E*(*y*) is the expected value of *y* for a given *x* value.

Linear regression helps in finding the best line of fit through the data by searching for the regression coefficient (B1) that helps in minimizing the total error of the model.

Now we can find or estimate the values for the parameters beta1 and beta2 using the equation for calculating the error. Then we define a model by minimizing the residual error.

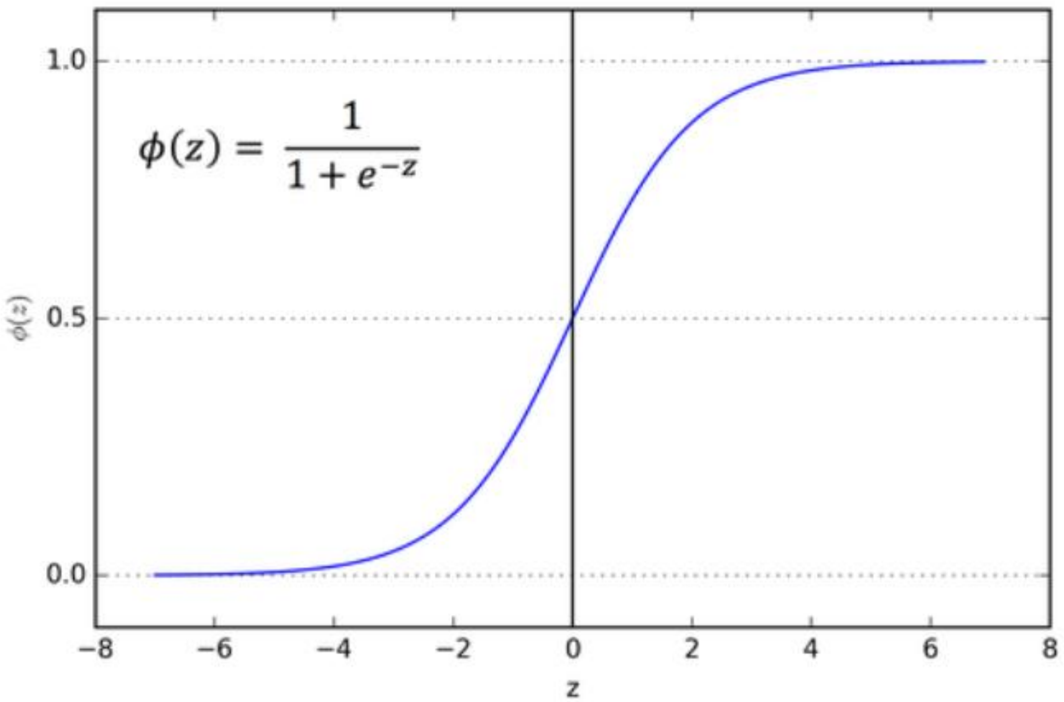
$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

The gradient descent technique is an optimization algorithm that helps in finding the values of parameters of a function to minimize the cost function.

Logistic regression

This type of regression is used when the dependent variable is categorical. The function used here is a sigmoid function.



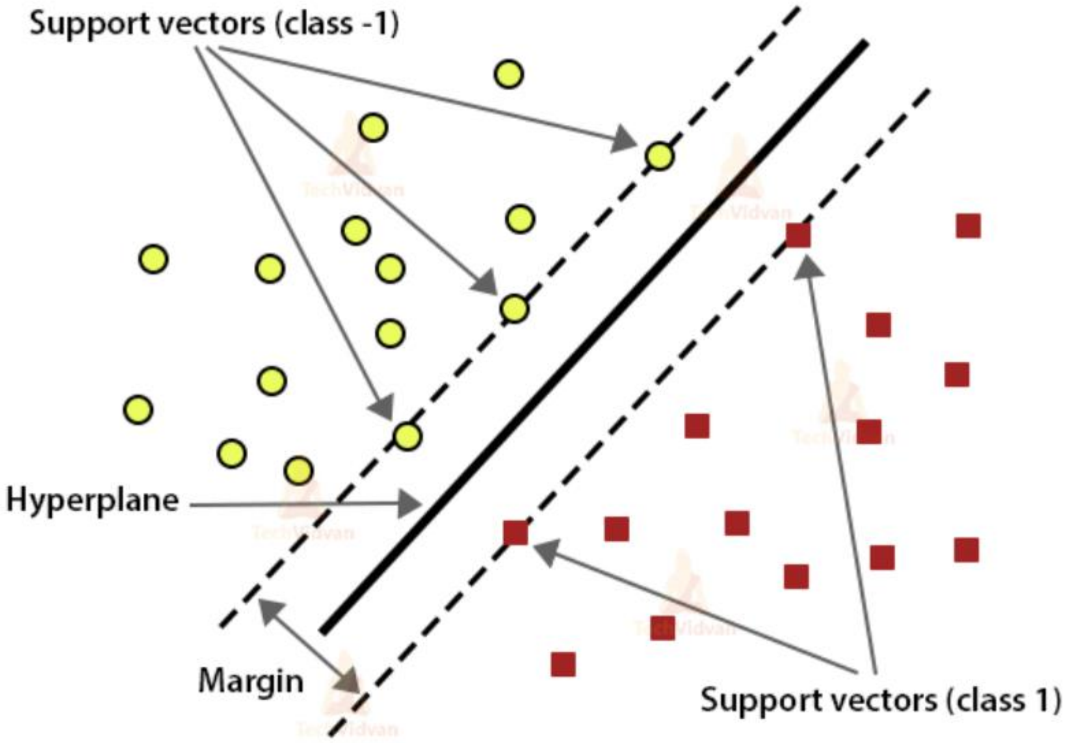
If ‘z’ tends to infinity, y tends to 1 and if ‘z’ tends to negative infinity, y tends to 0. So the outputs for a logistic regression will be 0 and 1. By using the function mentioned in the figure, we can define an odds ratio as

$$\text{odds} = \frac{P(y=1|x_1, x_2, \ldots, x_p)}{P(y=0|x_1, x_2, \ldots, x_p)} = \frac{P(y=1|x_1, x_2, \ldots, x_p)}{1 - P(y=1|x_1, x_2, \ldots, x_p)}$$

Support Vector Machine

In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.



In SVM, the best hyperplane is the one that maximizes the margins from both tags. The loss function that helps maximize the margin is hinge loss.

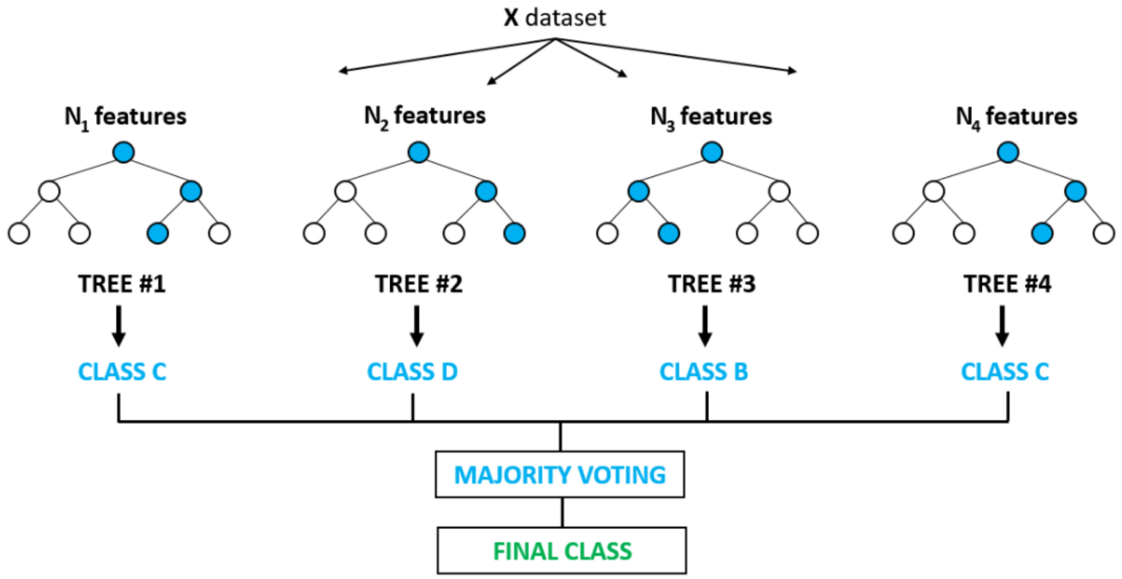
$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

Random Forest

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. It is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Random forest adds additional randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.



Project Development

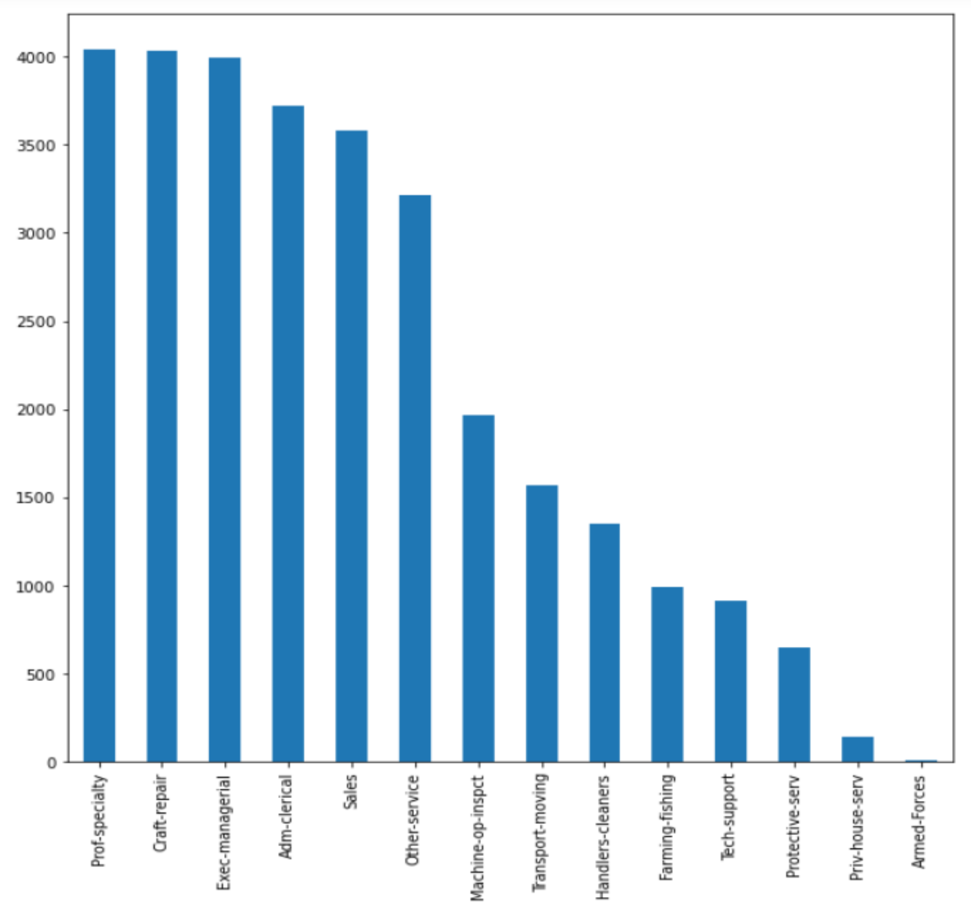
Now at the end phase of the project, I have cleaned, sanitized, visualized, done some EDA, and preprocessed the dataset enabling it for training. I have also trained and tested the model using logistic regression, SVM, Random Forest. Then I generated a classification report for each of the classifiers. Then I have tuned the parameters of the models, SVM and Logistic Regression, and chose the best ones. And again, I have tested and printed the classification report. Hence it was evident that the SVM model had more accuracy than the other 2. So I chose the SVM classifier over others. In the end, I have tested the model against a user-defined data tuple also.

Project Inference

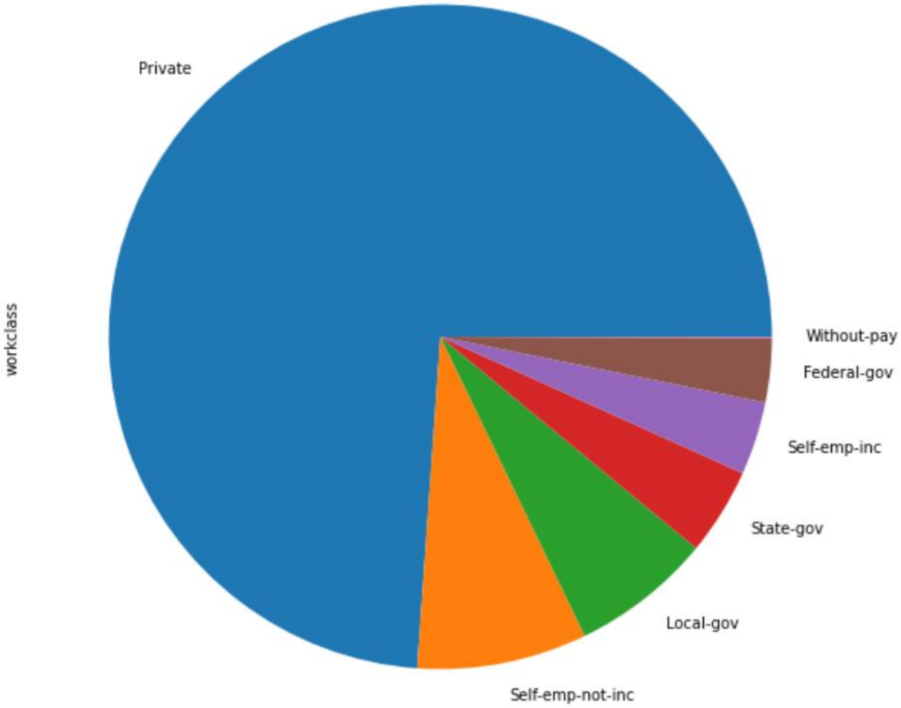
After hyper-parameter tuning and cross-validation of Logistic regression and SVM, the classification reports that I have generated are as follows.

Throughout the project till the second milestone, I have done some EDA and visualizing. They are as follows:

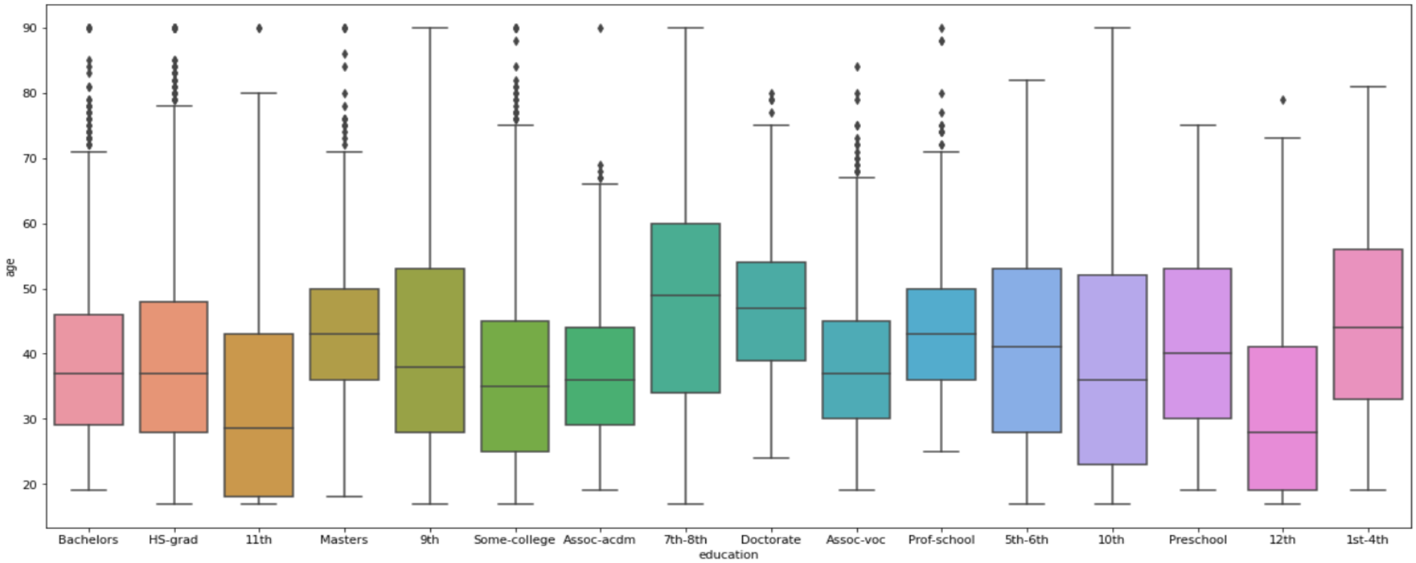
- This is a bar graph used for checking the variable ‘occupation’ in the dataset.



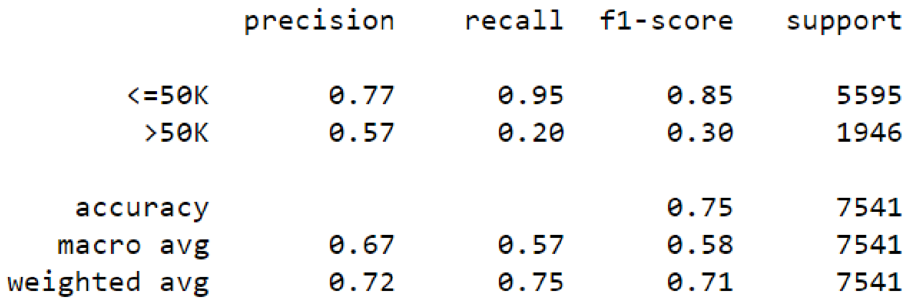
- This is a pie chart used for checking the variable ‘workclass’ in the dataset.



- This is a Box Plot used for checking the variables ‘education’ and ‘age’ in the dataset.

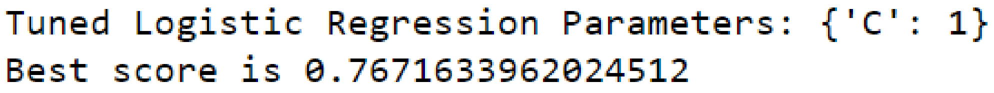


- This is a classification report of the logistic regression model before hyperparameter tuning. The accuracy score is almost 75.5%.



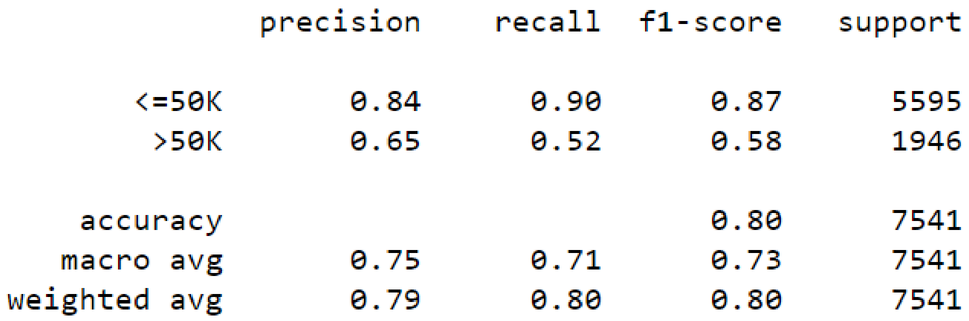
```
              precision    recall  f1-score   support

       <=50K       0.77      0.95      0.85      5595
        >50K       0.57      0.20      0.30      1946

    accuracy                           0.75      7541
   macro avg       0.67      0.57      0.58      7541
weighted avg       0.72      0.75      0.71      7541
```

The next figure shows the best accuracy score of the tuned logistic regression model with the used parameter, which is 76.7% So some improvement has been shown when the model is gone through the tuning process.



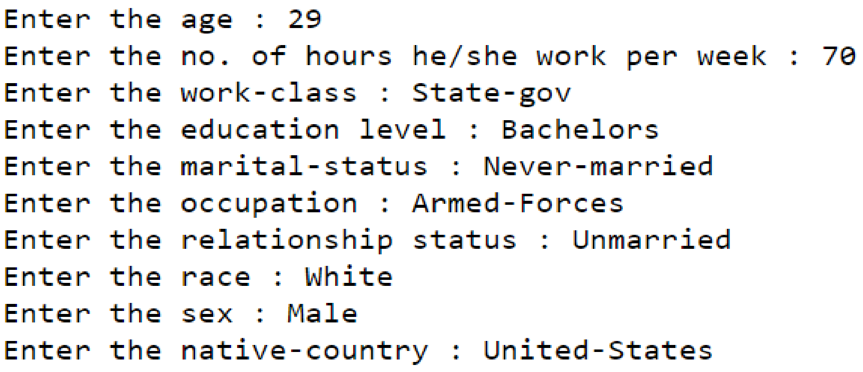
```
Tuned Logistic Regression Parameters: {'C': 1}
Best score is 0.7671633962024512
```

- This is a classification report of the random forest classifier which shows a good accuracy score of 80% and also has an increase in the precision compared to the non tuned logistic regression model.

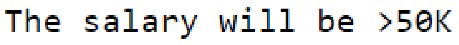


```
              precision    recall  f1-score   support

       <=50K       0.84      0.90      0.87      5595
        >50K       0.65      0.52      0.58      1946

    accuracy                           0.80      7541
   macro avg       0.75      0.71      0.73      7541
weighted avg       0.79      0.80      0.80      7541
```

- This is the input given for achieving a prediction using the logistic regression model. The predicted output is given as the next figure.

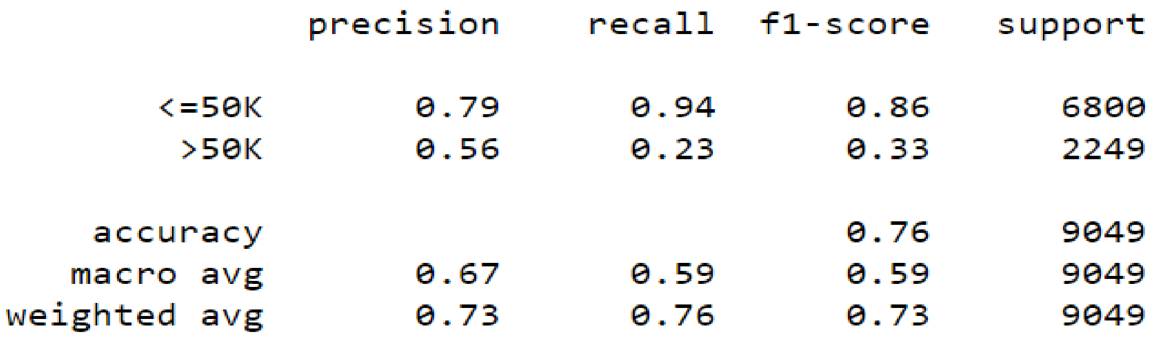


```
Enter the age : 29
Enter the no. of hours he/she work per week : 70
Enter the work-class : State-gov
Enter the education level : Bachelors
Enter the marital-status : Never-married
Enter the occupation : Armed-Forces
Enter the relationship status : Unmarried
Enter the race : White
Enter the sex : Male
Enter the native-country : United-States
```



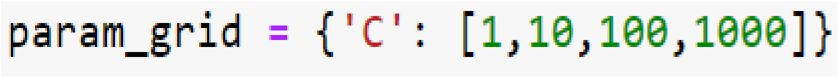
```
The salary will be >50K
```

- This is a classification report of Logistic regression after the tuning with some parameters.



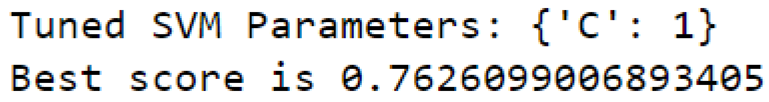
```
              precision    recall  f1-score   support

       <=50K       0.79      0.94      0.86      6800
        >50K       0.56      0.23      0.33      2249

    accuracy                           0.76      9049
   macro avg       0.67      0.59      0.59      9049
weighted avg       0.73      0.76      0.73      9049
```

This figure is the parameter used for the classification.



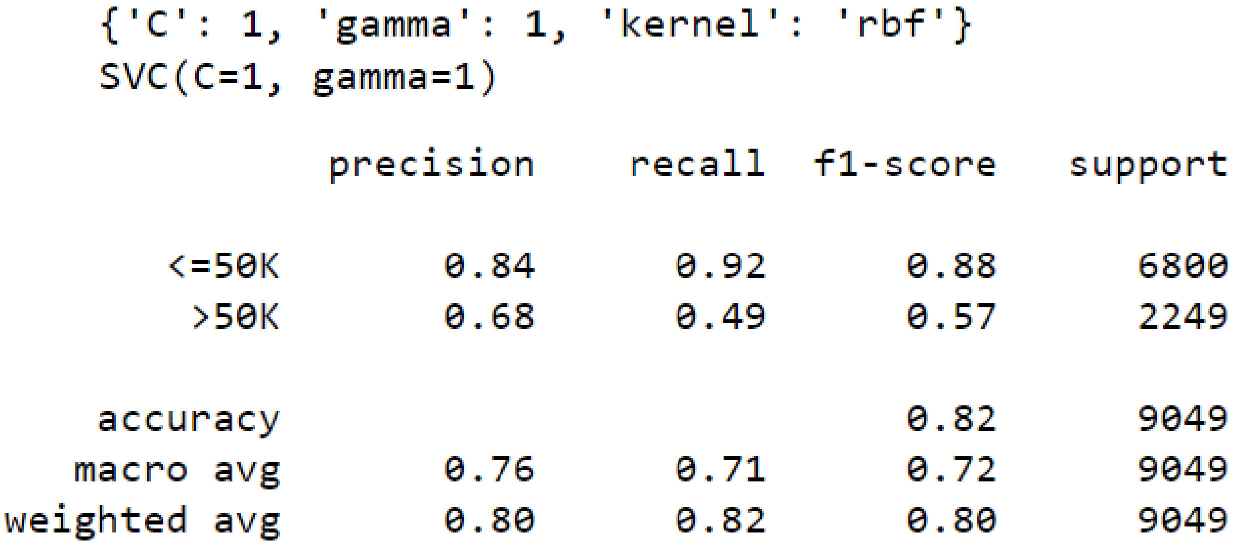
```
param_grid = {'C': [1,10,100,1000]}
```

- This is the score of SVM after the tuning with some parameters.



```
Tuned SVM Parameters: {'C': 1}
Best score is 0.7626099006893405
```

The next figure shows the parameters having the best score and classification report used in the cross-validation process.



```
{'C': 1, 'gamma': 1, 'kernel': 'rbf'}
SVC(C=1, gamma=1)

              precision    recall  f1-score   support

       <=50K       0.84      0.92      0.88      6800
        >50K       0.68      0.49      0.57      2249

    accuracy                           0.82      9049
   macro avg       0.76      0.71      0.72      9049
weighted avg       0.80      0.82      0.80      9049
```

To conclude, HR Salary Dashboard works like a very useful tool for predicting the salary of new employees. Including an attribute like work experience or internship experience might be useful for predicting the salary in an efficient manner.

ENHANCEMENT SCOPE

This industry project has a wide scope. Using the resume or CV of an individual, one can actually predict the salary. Some of the Natural language processing techniques will help in developing this application.

LINK TO CODE AND EXECUTABLE FILE

Link to the colab file:
https://colab.research.google.com/drive/1bv2emmxIlnsemMEFFEdmNNGqXtOu1G4g#scrollTo=TlSvT73XCP3E

Link to the GitHub repository:

https://github.com/VishnuVenkat18/HR-Salary