# PROJECT REPORT

On Predictive Analysis for the Reduction of Downtime for AMS and Makino Machines





**Submitted by**

**Padala venkata sai**

**TABLE OF CONTENTS**

**1** **PROJECT CHARTER PURPOSE**

[**2**](#_heading=h.2bn6wsx) **PROJECT EXECUTIVE SUMMARY**

[**3**](#_heading=h.qsh70q) **PROJECT OVERVIEW 3**

[**4**](#_heading=h.3as4poj) **PROJECT SCOPE 4**

[4.1](#_heading=h.1pxezwc) Goals and Objectives 4

[4.2](#_heading=h.49x2ik5) Project Deliverables 4

[4.3](#_heading=h.2p2csry) Deliverables Out of Scope 4

[**5**](#_heading=h.3o7alnk) **PROJECT CONDITIONS 5**

[5.1](#_heading=h.23ckvvd) Project Assumptions 5

[5.2](#_heading=h.ihv636) Project Issues 5

[5.3](#_heading=h.32hioqz) Project Risks 5

[5.4](#_heading=h.1hmsyys) Project Constraints 5

[**6**](#_heading=h.41mghml) **Project Structure Approach 6**

**7 Project report 7**

**8 RESULT AND CONCLUSION 8**

**9****PROJECT REFERENCES 9**

**1.PROJECT CHARTER PURPOSE**

* Defines the scope, objectives, and overall approach for the project.
* Serves as a critical element for initiating, planning, executing, controlling, and assessing the project.
* Acts as a single point of reference for project goals, objectives, scope, organization, estimates, work plan, and budget.
* Functions as a contract between the Project Team and Project Sponsors, stating deliverables, budget, time constraints, risks, resources, and standards agreed upon.

**2.** **PROJECT EXECUTIVE SUMMARY**

* Project goals
* Objectives
* Data prepossessing
* Scope
* Assumptions
* Risks
* Costs
* Timeline
* Approach
* Organization

**3.PROJECT OVERVIEW**

* The objective of this initiative is to enhance operational efficacy and decrease expenses by minimizing downtime on AMS and Makino machines. This will be achieved through a comprehensive approach encompassing data gathering and analysis, engagement with key stakeholders, creation of a strategic plan, successful execution, ongoing monitoring and assessment, and a commitment to ongoing improvement.

**4.PROJECT SCOPE**

* Comprehensive Data Collection and Analysis: Systematic gathering and examination of machine uptime and downtime data to identify areas for improvement.
* Strategy Development: Development of strategies aimed at reducing downtime through improvements in preventive maintenance processes, equipment upgrades, and operator training.
* Implementation: Successful execution of the strategies developed, including the implementation of process changes, equipment upgrades, and operator training.

4.1 **GOAL AND OBJECTIVE**

**GOAL:-**

* Reducing the downtime and improving the efficiency of machines

**OBJECTIVE**

* To gather data on machine uptime and downtime, including the causes of downtime.
* To identify and develop strategies for reducing downtime.
* To implement the strategies developed. and monitor the success of the strategies implemented
  1. **Project Deliverables**

**Milestone**

* + Identifying Constraints and design the project architecture, explore various public forums to collect relevant data, Data Preparation.
  + EDA and Descriptive Analytics, Model Building for Association (Fuzzy Algorithm) and Recommendation
  + Model Evaluation, tuning and insights, Deployment
  + Show case and review, Final Presentation and documentation, Handover and KT.

**Deliverable**

* Deliverable 1.1—Identifying Constraints and design the project architecture.
* Deliverable 1.2—Explore various public forums to collect relevant data.
* Deliverable 1.3— Data Preparation
* Deliverable 2.1— EDA and Descriptive Analytics
* Deliverable 2.2— Model Building for Association (Fuzzy Algorithm) and Recommendation
* Deliverable 3.1— Model Evaluation, tuning and insights.
* Deliverable 3. 2— Deployment
* Deliverable4.1 – show case and review
* Deliverable4.2 – Final Presentation and documentation
* Deliverable4.3 – Handover and KT
  1. **Deliverables Out of Scope**
  + **Designs:** This likely refers to any design work unrelated to the project's primary objectives, such as graphic design or industrial design that is not directly related to the project's goals of minimizing downtime on specific machines.
  + **Web Application:** The project explicitly states that creating a web application is out of scope. This indicates that the project does not involve the development or deployment of a web application as part of its objectives.
  + **MONOTORING AND EVALUATION**

1. **PROJECT CONDITIONS**

**5.1 Project Assumptions**

● Work on data which is extracted from public sources.

● Can create a web API by using flask or streamlit.

● Robust Tested: Application should be tested for noise data also.

* 1. **Project Issues**

Priority Criteria

1 − High-priority/critical-path issue; requires immediate follow-up and resolution.

2 − Medium-priority issue; requires follow-up before completion of next project milestone.

3 − Low-priority issue; to be resolved prior to project completion.

4 − Closed issue.

# Date Priority Owner Description Status & Resolution

1 High

2 High

**5.3 Project Risks**

# Risk Area Likelihood Risk Owner Project Impact-Mitigation Plan

1 [Project Risk] [High/Medium/Low] Technical issues with the equipment upgrades, which could result in additional downtime.

2 [Project Risk] [High/Medium/Low] Inaccurate data which could lead to incorrect conclusions and ineffective strategies.

**5.4 Project Constraints**

● Technical constraints, such as limited access to certain equipment or data sources.

● While reducing the downtime of the machines it should not impact on the profits of the organization.

**6.Project Structure Approach**

Project Initiation

↓

Data Collection

↓

Data Analysis

↓

Model Building

↓

Model Evaluation

↓

Model Deployment

**7 .PROJECT CODE**

**Step 1:-**  IMPORTING THE LIBRARIES

**# For Data Processing**

# Numerical operations in Python

# Data manipulation and analysis

**# For Visualization**

# Plotting library

# Data visualization library based on matplotlib

**# For Modelling**

# Splitting data into train and test sets

# Standardizing features by removing the mean and scaling to unit variance

# Evaluation metrics for classification models

# Random Forest classifier model

# Library for encoding categorical variables

# Cross-validation techniques

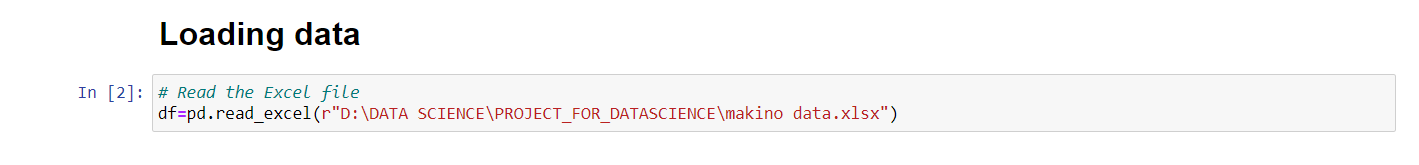
# Warning control

# Suppress warnings

****

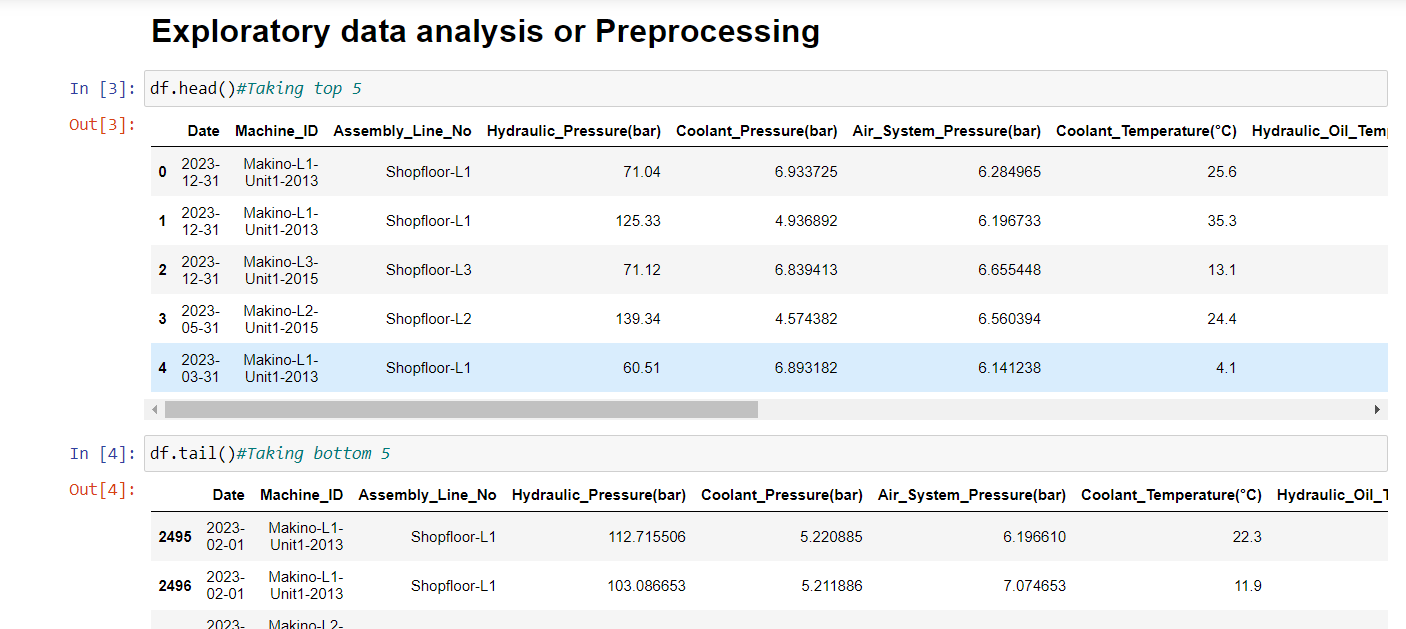
**STEP 2:-**

READING THE DATA SET

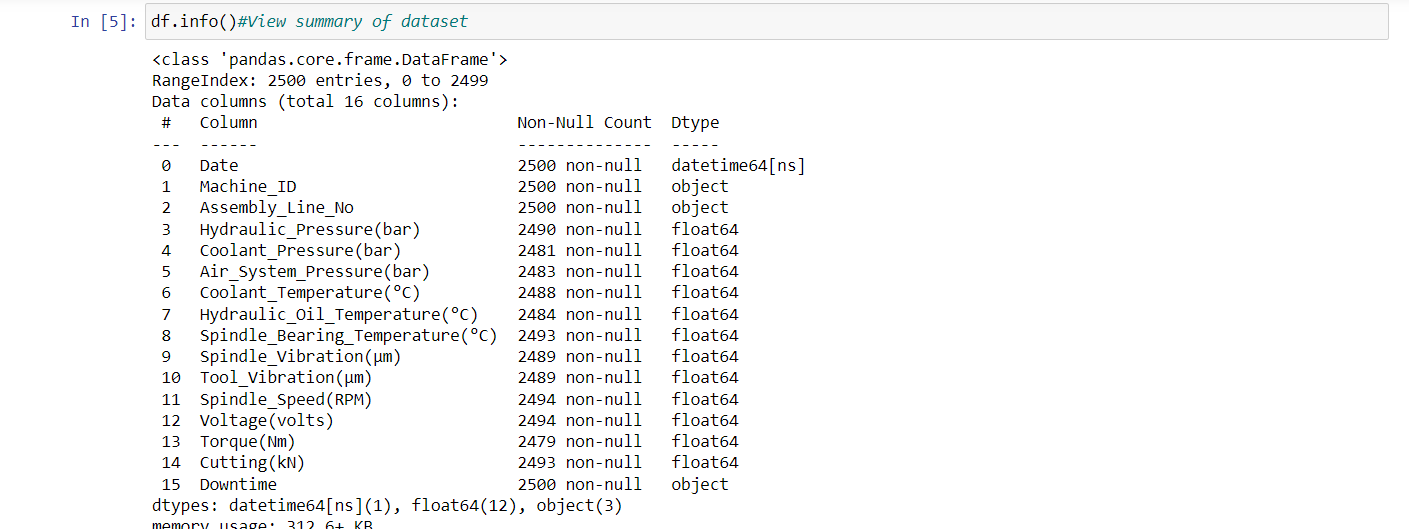
****

**STEP 3:-**  DATA PREPROCESSING :-

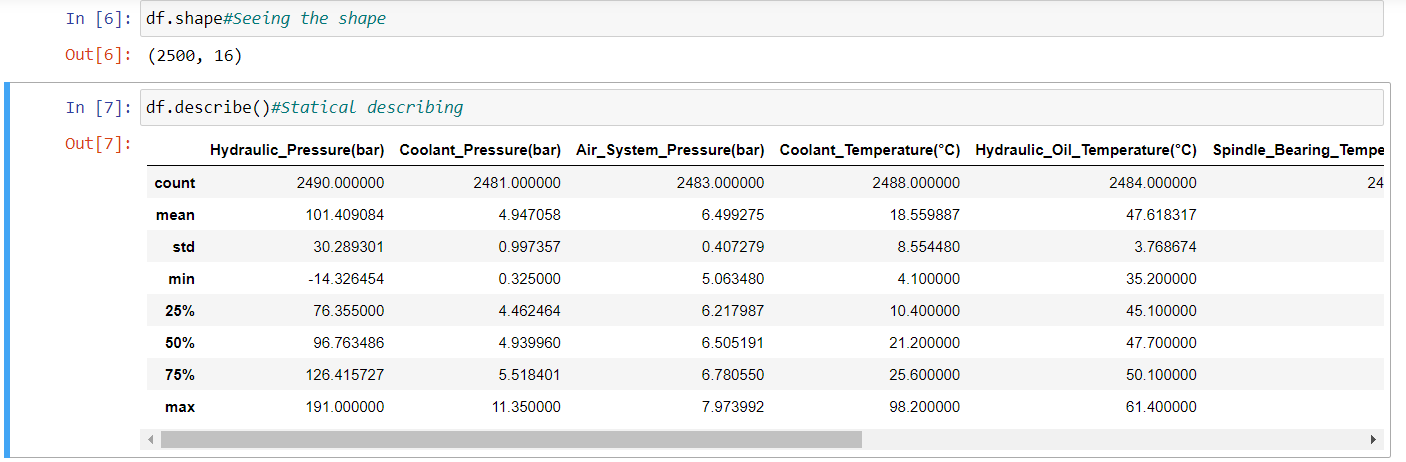
**1.**taking top 10 and bottom 10



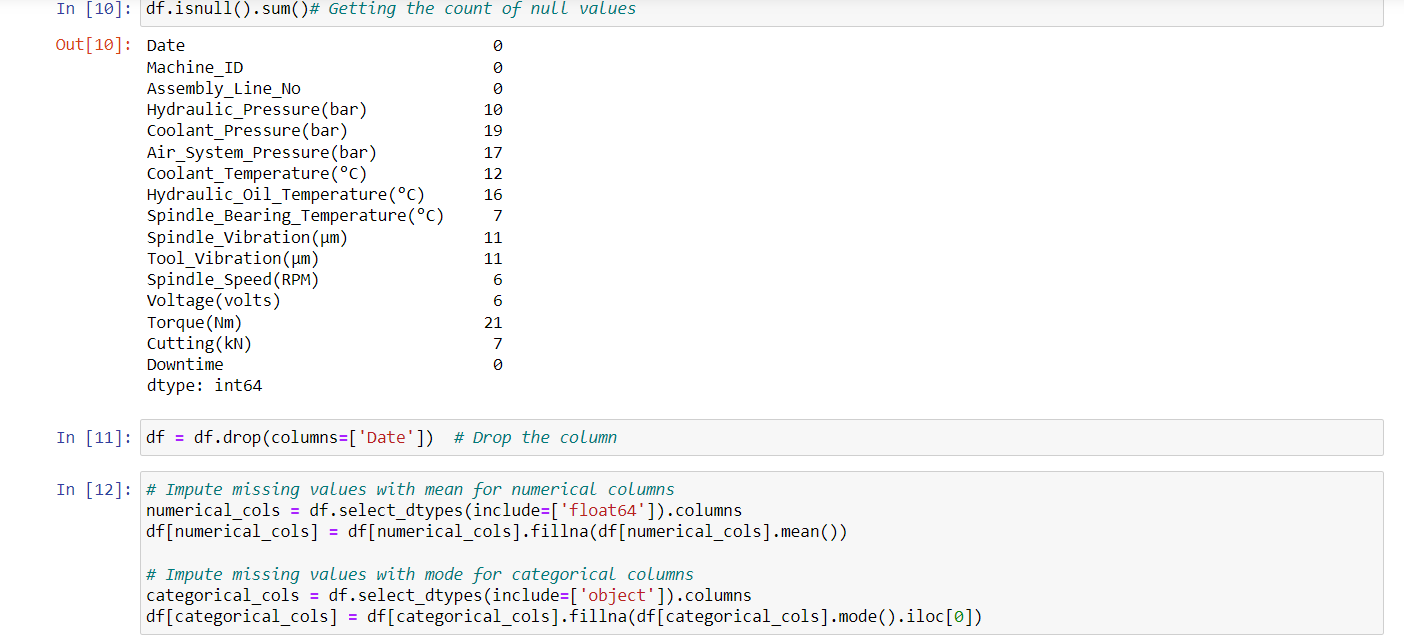
**2.**summary of the dataset



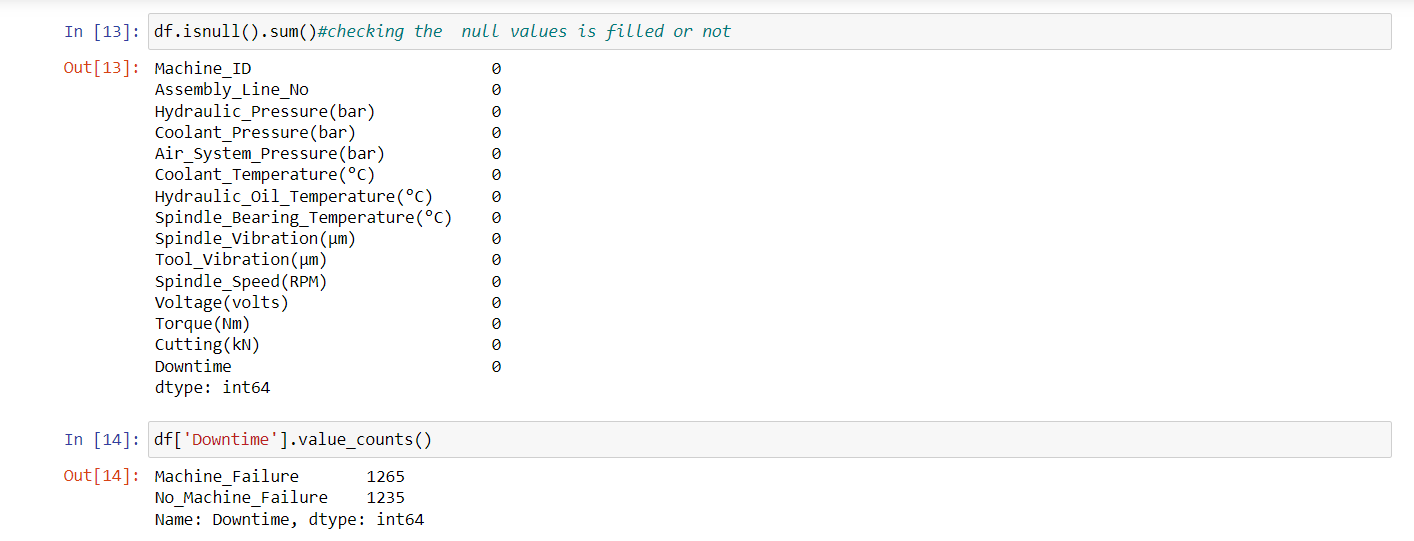
**3.**shape of the dataset and statical describing



**4.** Checking the null values and droping the data column because no use and here iam not droping null values instead of that imputeing the missing values with mean and mode for numerical columns

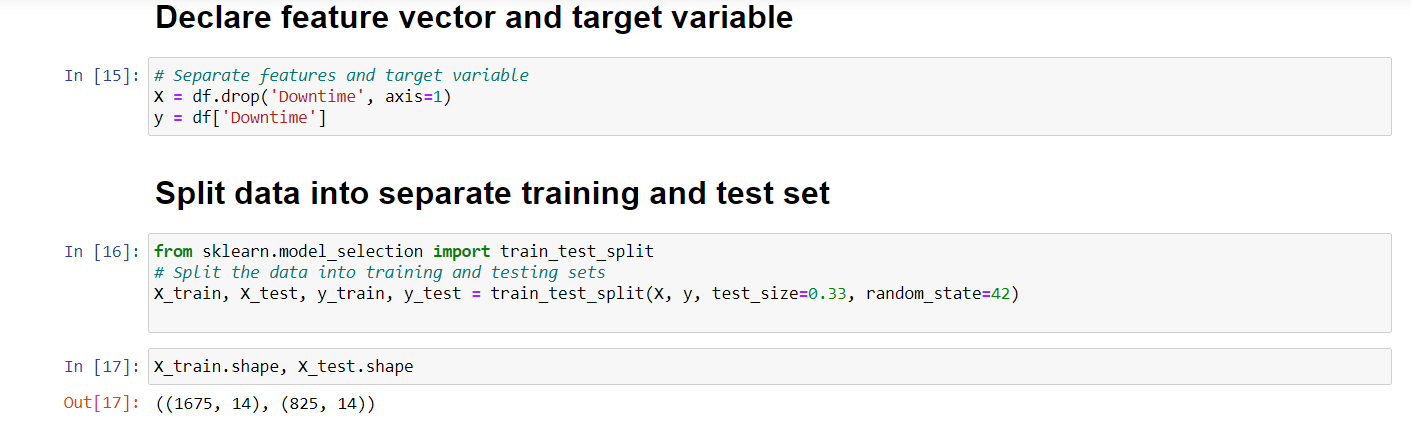


**5.** and iam checking the null values are filled or not filled and also checking the values count of Down time

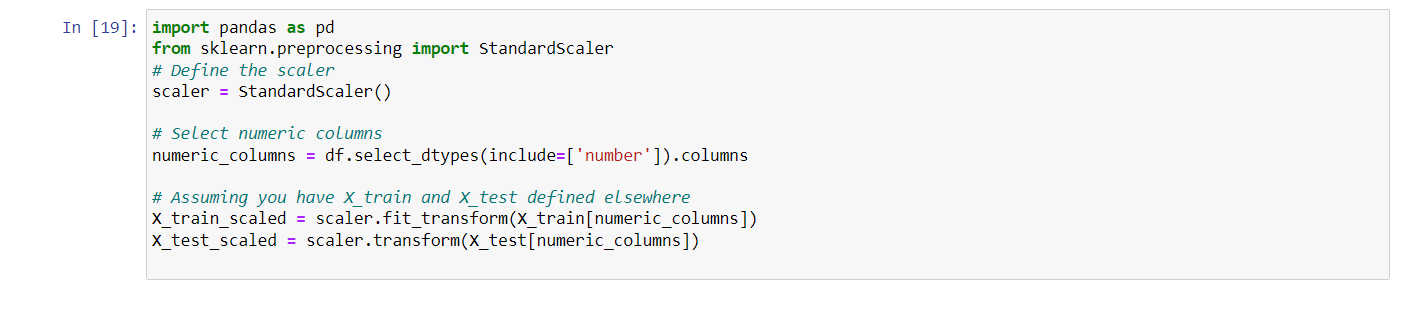


**STEP:-4**

**6.**Declaring feature vector and targeting variable to predict and splitting the data into separate training and test sets and seeing the X\_train,X\_test.shape



**7.**here using scalar standard scale and selecting the numerical columns and then assuming x train and x test



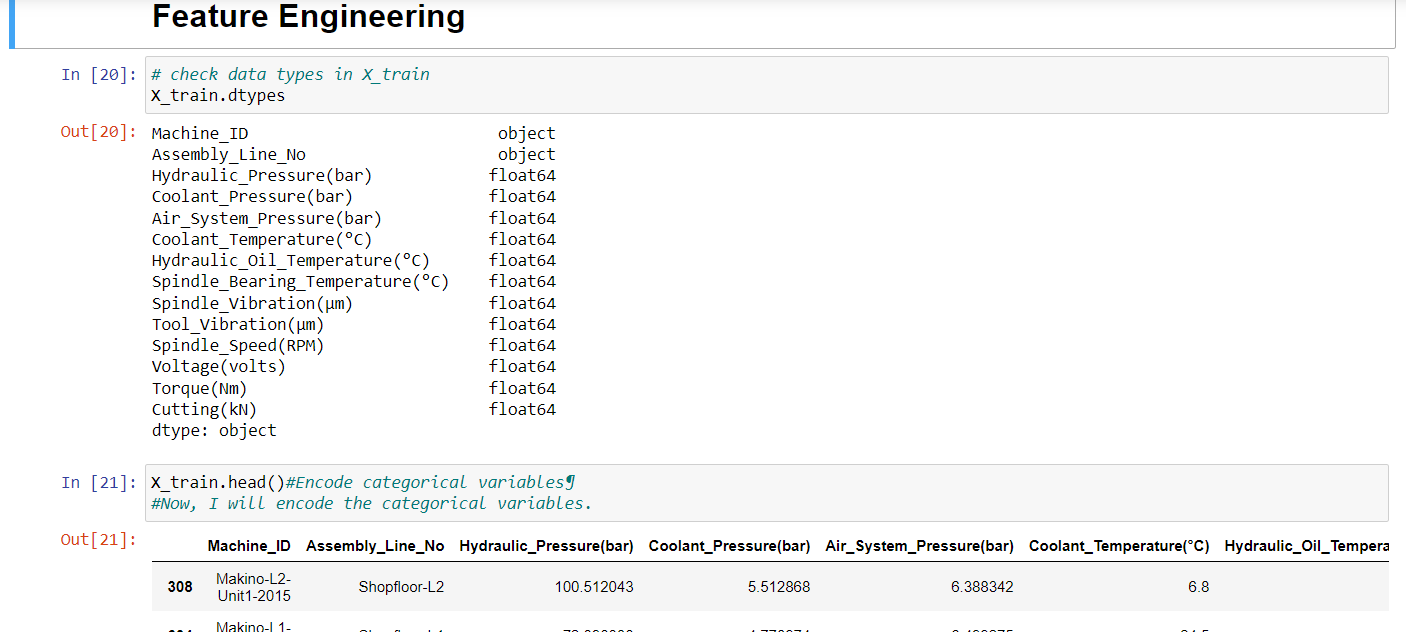
**STEP :-5**

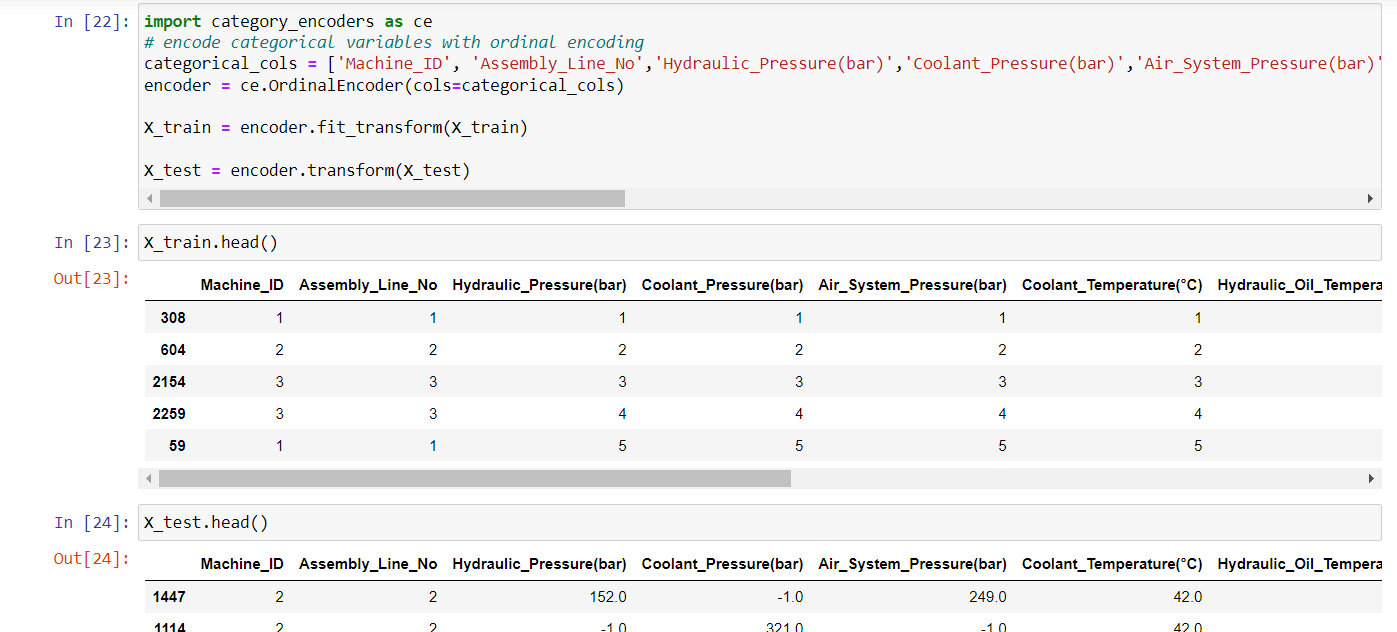
**8.** **Feature Engineering**

is the process of transforming raw data into useful features that help us to understand our model better and increase its predictive power. I will carry out feature engineering on different types of variables.

First, I will check the data types of variables again.

here iam checking the data types and encoded categorical variable now I will encode the categorical variable.



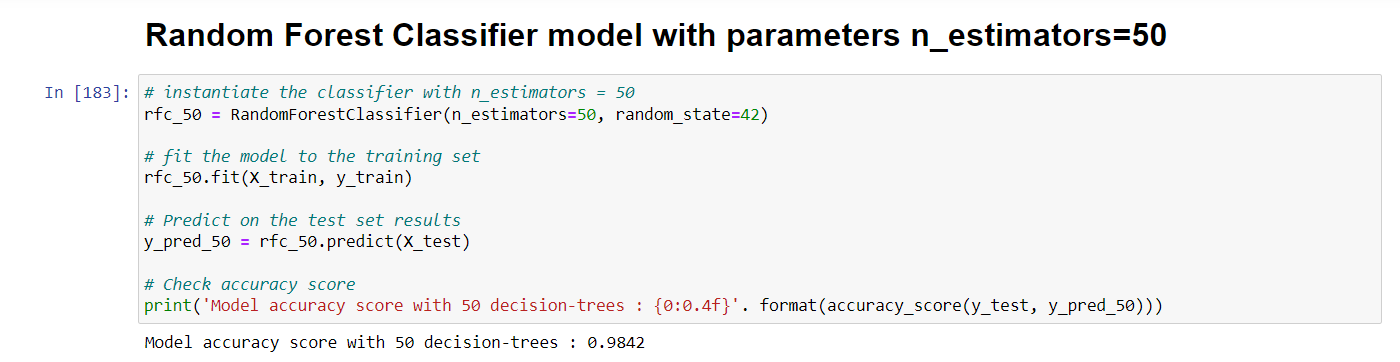
**9.** and I had imported categorical variable with ordinal encoding and fitting the transform xtrain and x test and see top 5

**STEP :-6**

**10.**Model accuracy score with 10 decision-trees : 0.9247

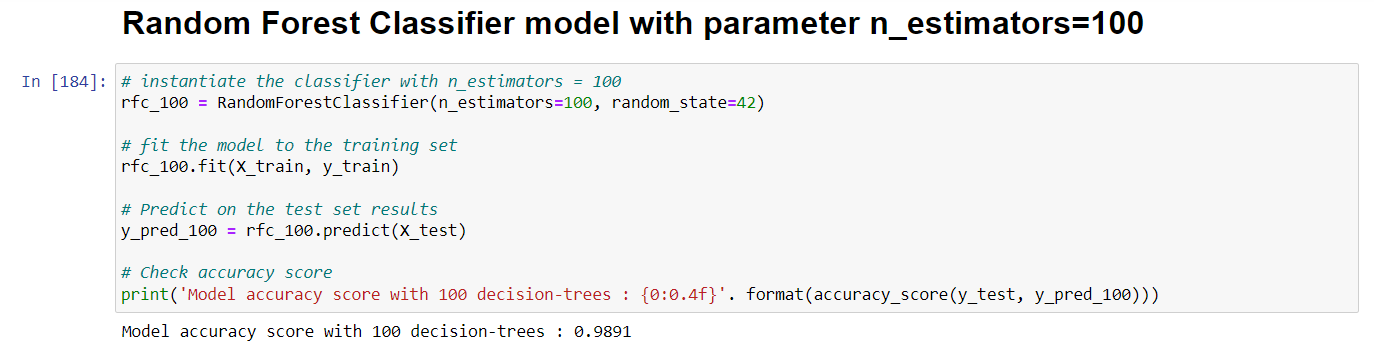
Here, **y\_test** are the true class labels and **y\_pred** are the predicted class labels in the test-set.

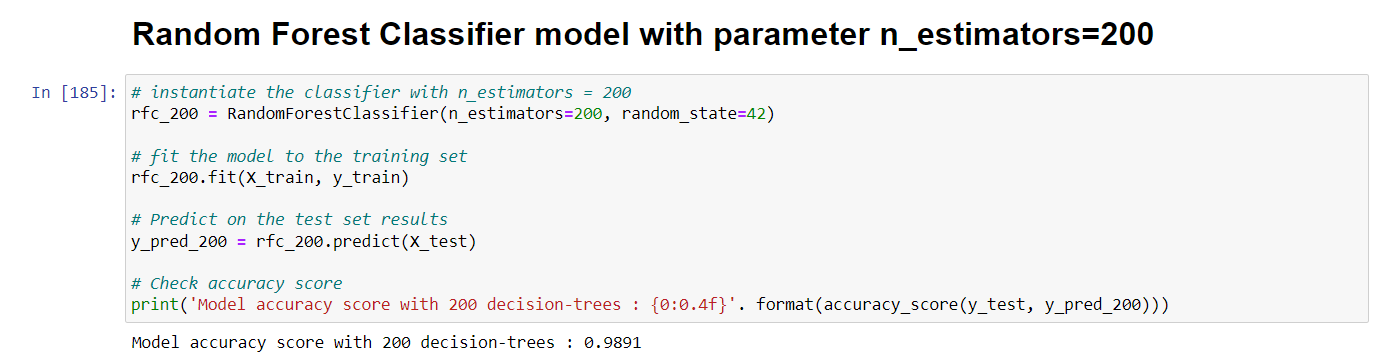
Here, I have build the Random Forest Classifier model with default parameter of n\_estimators = 50. So, I have used 10 decision-trees to build the model. Now, I will increase the number of decision-trees and see its effect on accuracy.



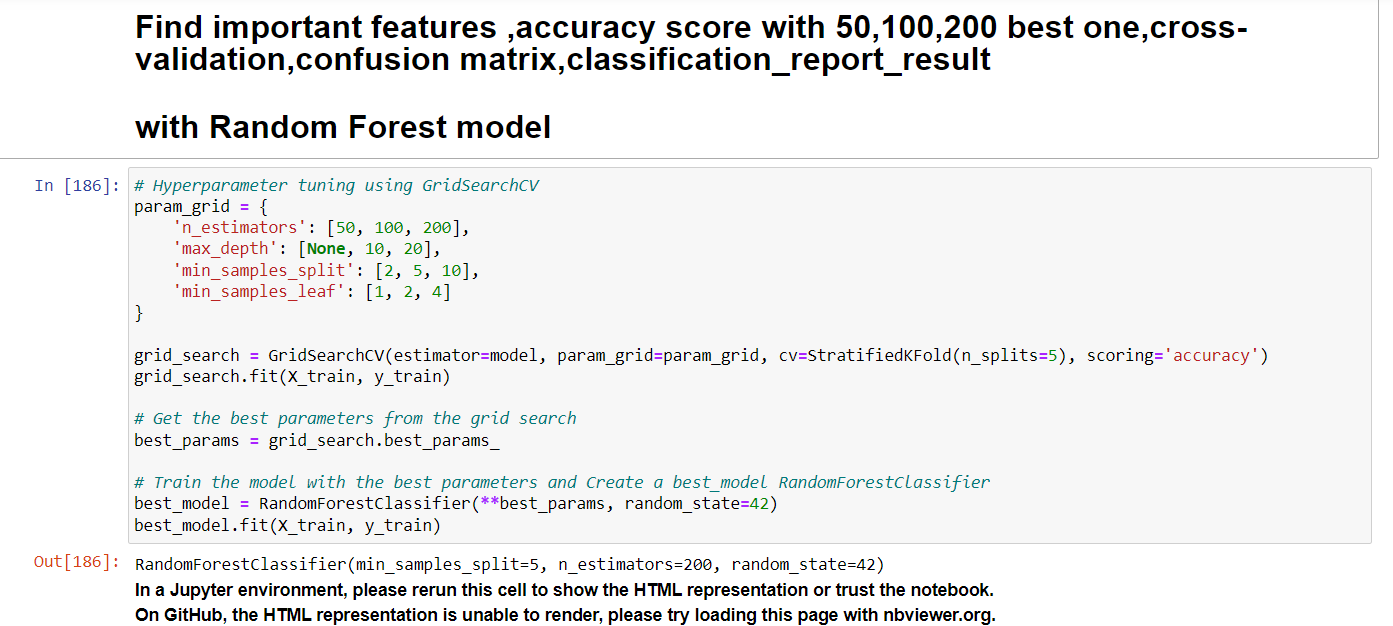
1**1.**Model accuracy score with 100 decision-trees : 0.9457

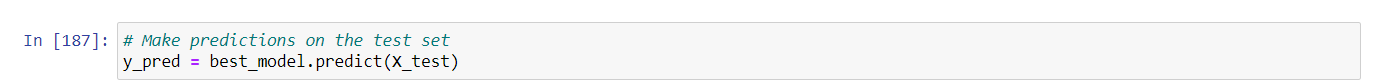
The model accuracy score with 100 decision-trees is 0.9247 but the same with 100 decision-trees is 0.9457. So, as expected accuracy increases with number of decision-trees in the model.





**12.**here below it hyperparameter tunning using grid serch that takes best parameters for grid serch and train the model with the best parameters and create a best model random forest

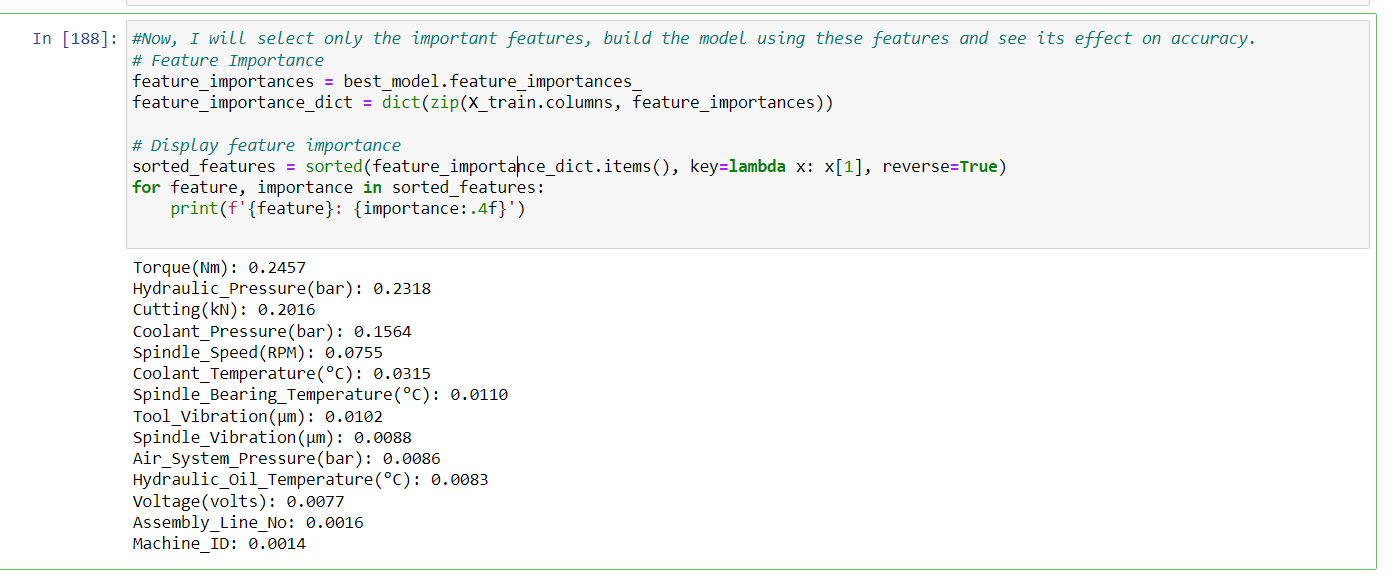




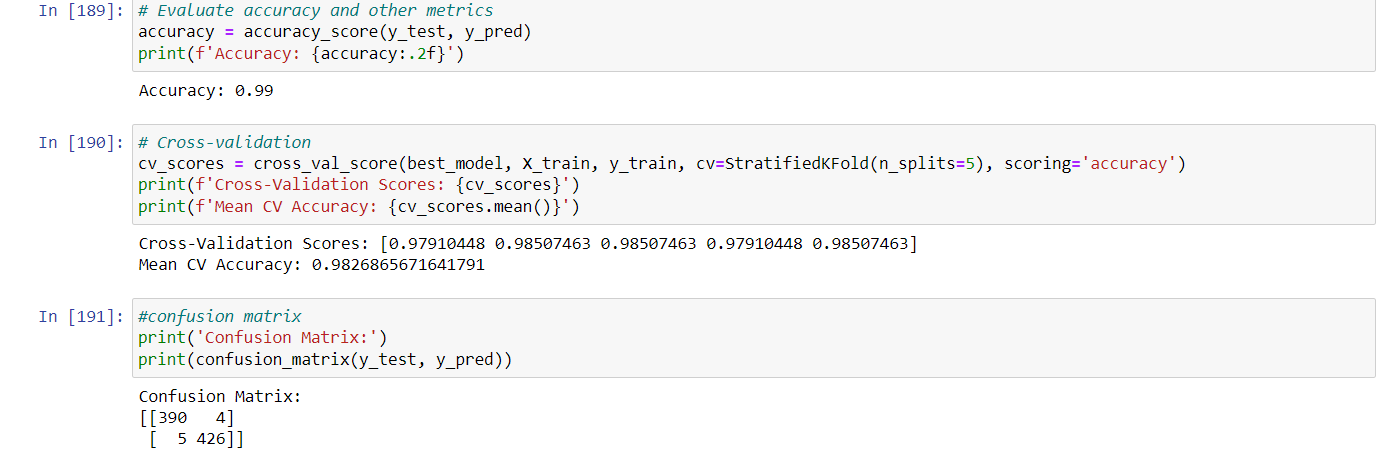
**STEP:-7**

**13. Find important features with Random Forest model**

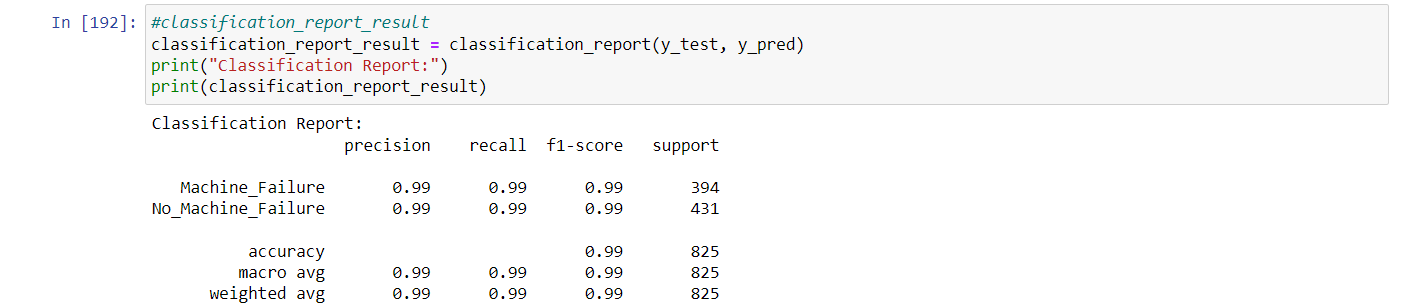
Until now, I have used all the features given in the model. Now, I will select only the important features, build the model using these features and see its effect on accuracy.

Now, I will use the feature importance variable to see feature importance scores. 

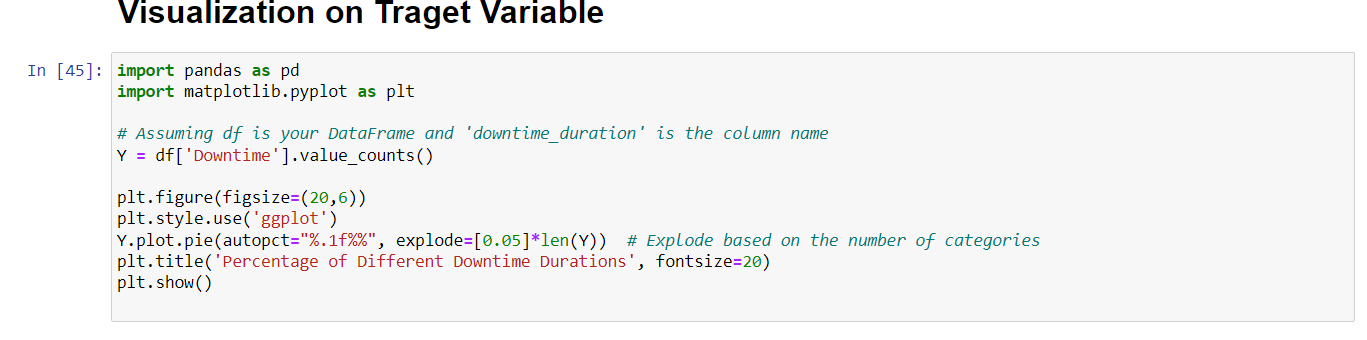
**14.**Once again checking the accuracy of the metrics and crossvalidation score and also confusion matrix

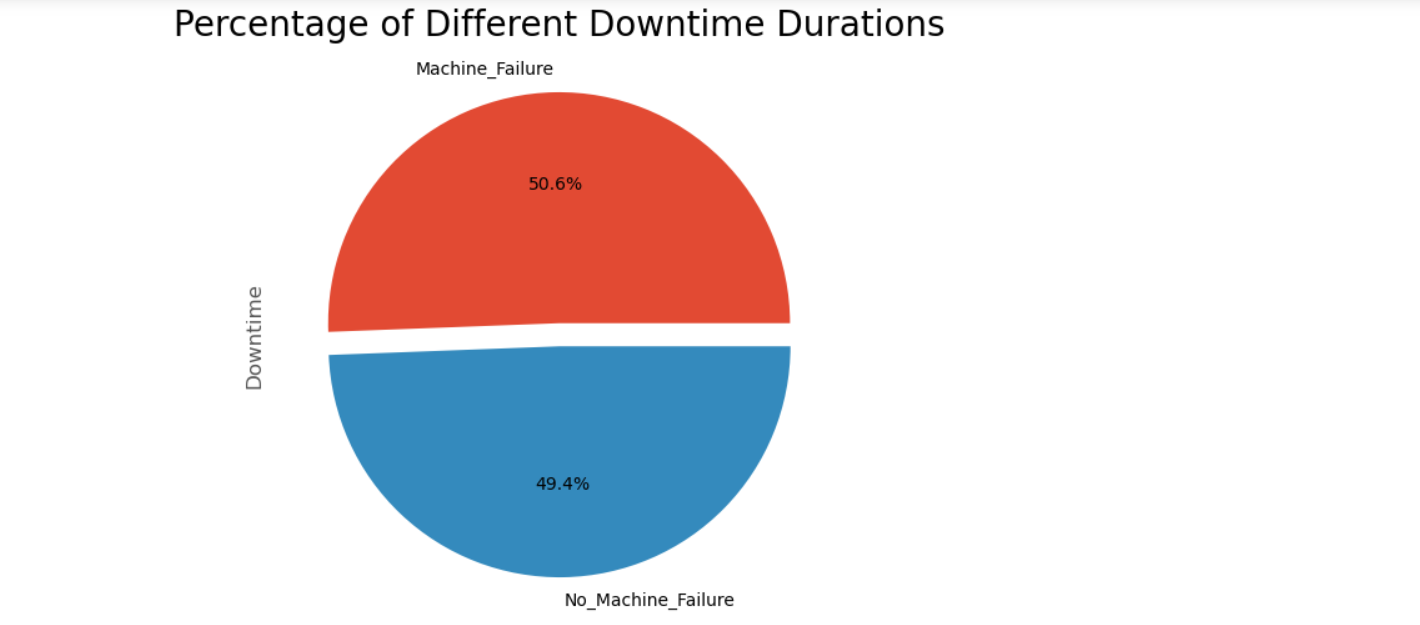


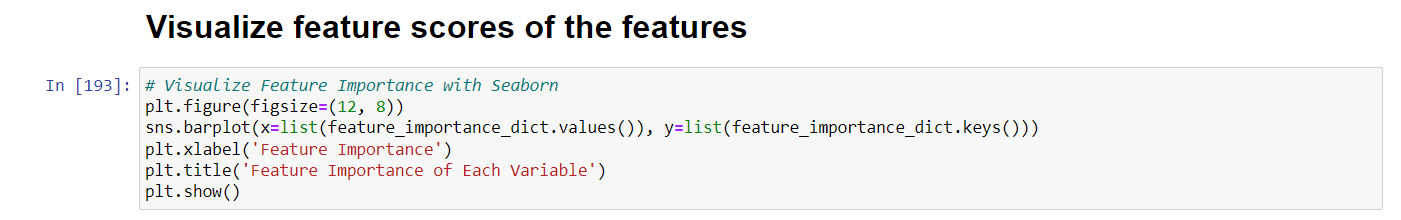
**15**. Here the final classification report

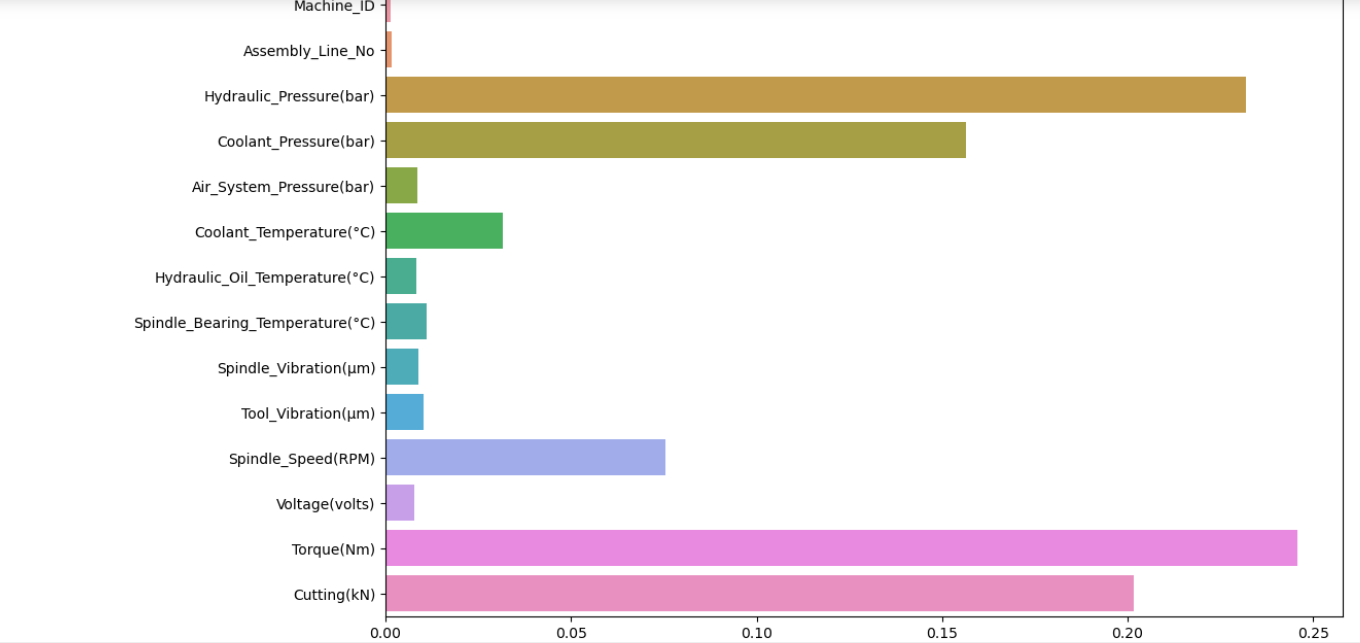


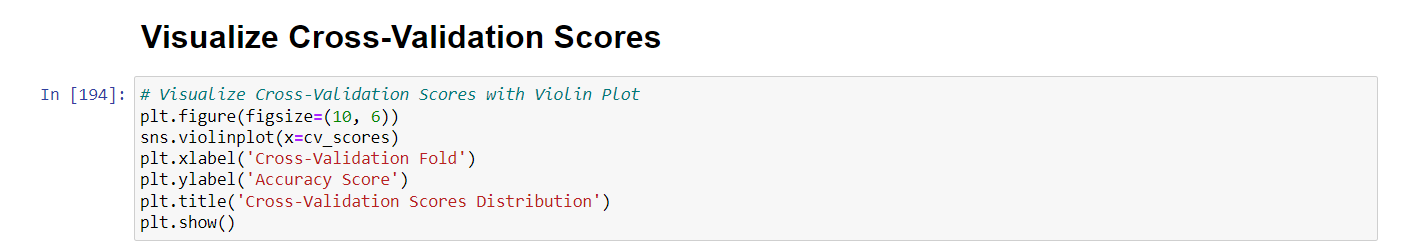
**STEP:-8 THE VISUALIZATION ON ABOVE SCORES AND TRAGET VARIABLE**

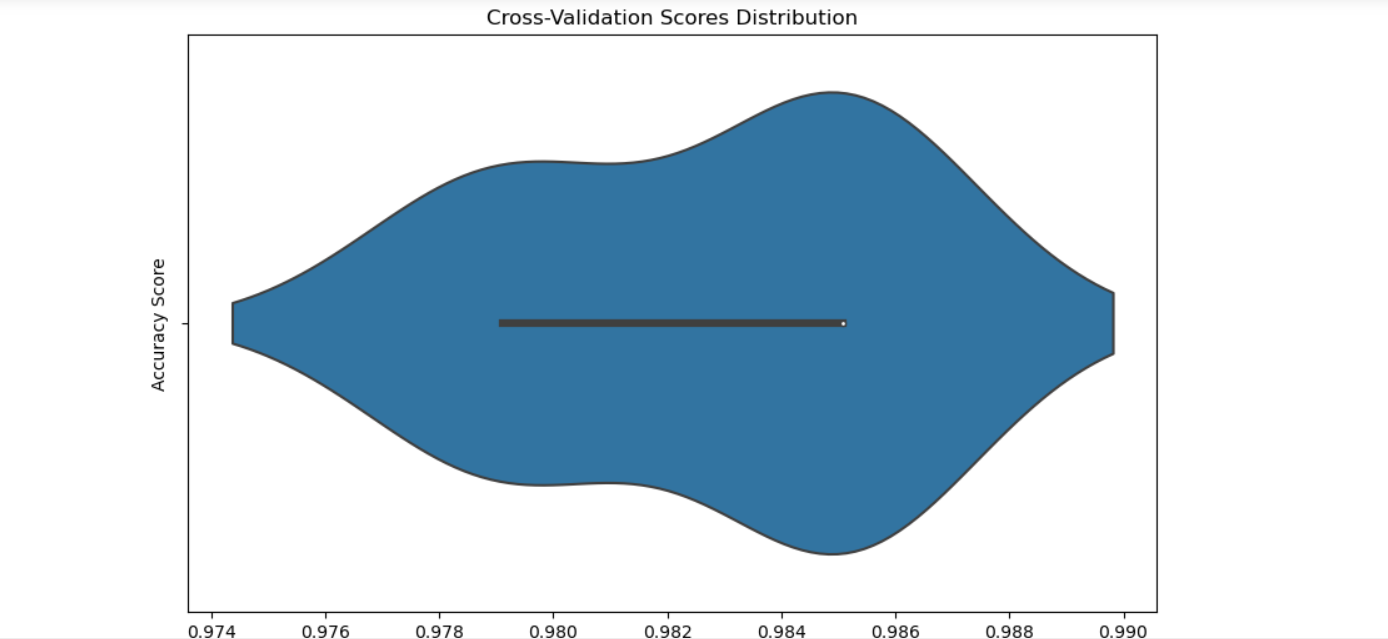




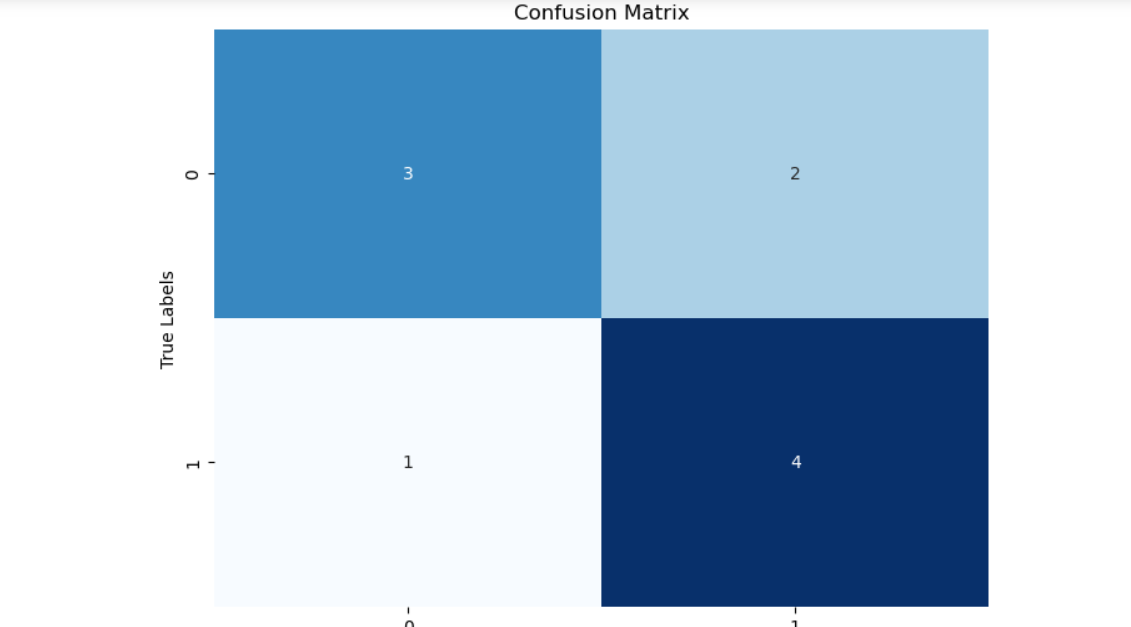


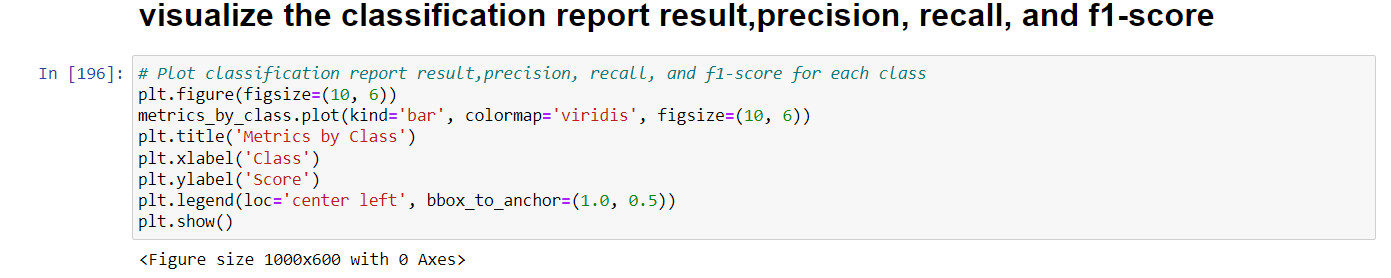


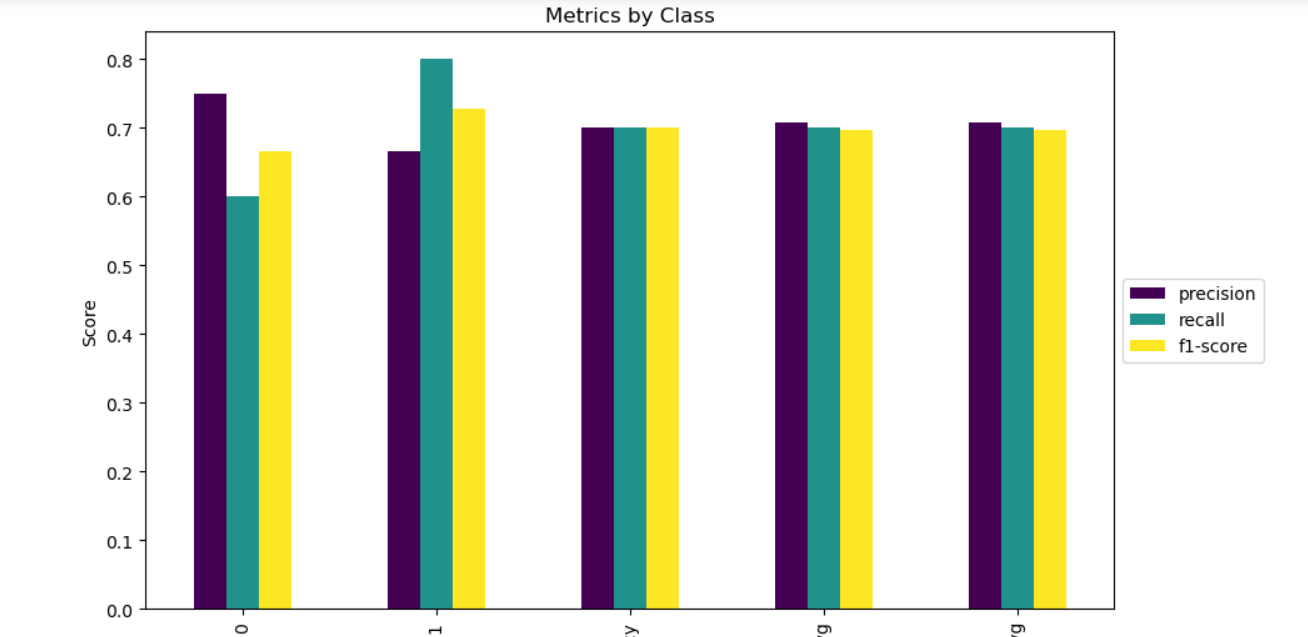












DOWN TIME variable from the model, rebuild it and checked its accuracy. The accuracy of the model with doors variable removed is 0.9264. The accuracy of the model with all the variables taken into account is 0.9247. So, we can see that the model accuracy has been improved with DOWN TIME variable from the model.

Now, based on the above analysis we can conclude that our classification model accuracy is very good. Our model is doing a very good job in terms of predicting the class labels.

But, it does not give the underlying distribution of values. Also, it does not tell anything about the type of errors our classifer is making.

We have another tool called Confusion matrix that comes to our rescue.

**CONFUSION METRIX**

A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give us a clear picture of classification model performance and the types of errors produced by the model. It gives us a summary of correct and incorrect predictions broken down by each category. The summary is represented in a tabular form.

Four types of outcomes are possible while evaluating a classification model performance. These four outcomes are described below:-

**True Positives (TP)** – True Positives occur when we predict an observation belongs to a certain class and the observation actually belongs to that class.

**True Negatives (TN)** – True Negatives occur when we predict an observation does not belong to a certain class and the observation actually does not belong to that class.

**False Positives (FP)** – False Positives occur when we predict an observation belongs to a certain class but the observation actually does not belong to that class. This type of error is called **Type I error.**

**False Negatives (FN)** – False Negatives occur when we predict an observation does not belong to a certain class but the observation actually belongs to that class. This is a very serious error and it is called **Type II error.**

These four outcomes are summarized in a confusion matrix given above.

**Classification report** is another way to evaluate the classification model performance. It displays the **precision**, **recall**, **f1** and **support** scores for the model. I have described these terms in later.

We can print a classification report as follows above

# ****Results and conclusion****

1. In this project, I build a Random Forest Classifier to predict the safety of the car. I build 3MODELS , one with 50 decision-trees and another one with 100 and 200 then hyperparameter grid serch to check wjich one is best modell again confirm the best one of accuracy decision-trees.
2. The model accuracy score with 50 decision-trees is 0.9247 but the same with 100 and 200 decision-trees is 0.9457. So, as expected accuracy increases with number of decision-trees in the model.
3. I have used the Random Forest model to find only the important features, build the model using these features and see its effect on accuracy.
4. Down time variable from the model, rebuild it and checked its accuracy. The accuracy of the model with down time  variable removed is 0.9264. The accuracy of the model with all the variables taken into account is 0.9247. So, we can see that the model accuracy has been improved withdown time variable from the model.
5. Confusion matrix and classification report are another tool to visualize the model performance. They yield good performance.

**9.PROJECT REFERENCE**

The work done in this project is inspired from following books and websites:-

1. Hands on Machine Learning with Scikit-Learn and Tensorflow by Aurélién Géron
2. Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido
3. <https://en.wikipedia.org/wiki/Random_forest>
4. <https://www.datacamp.com/community/tutorials/random-forests-classifier-python>
5. <http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/>
6. <https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/>