1. **For Breast cancer dataset build a machine learning model to predict or identify it and to perform the following operations**

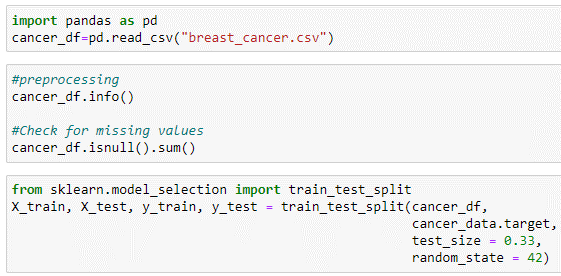
**i)  import libraries**

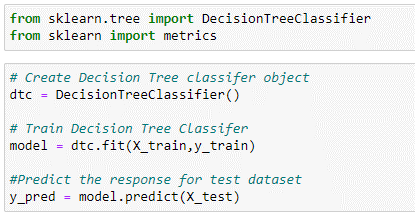
**ii) Perform pre-processing**

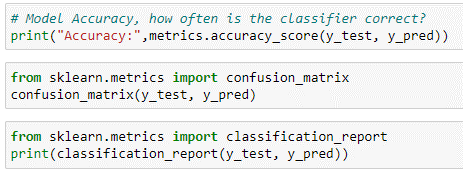
**iii)  Split the data set**

**iv) Find the accuracy**

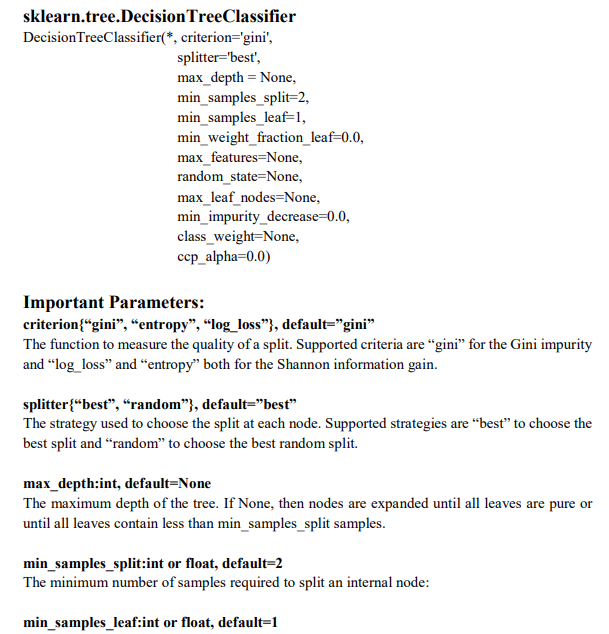
1. **Data prediction**





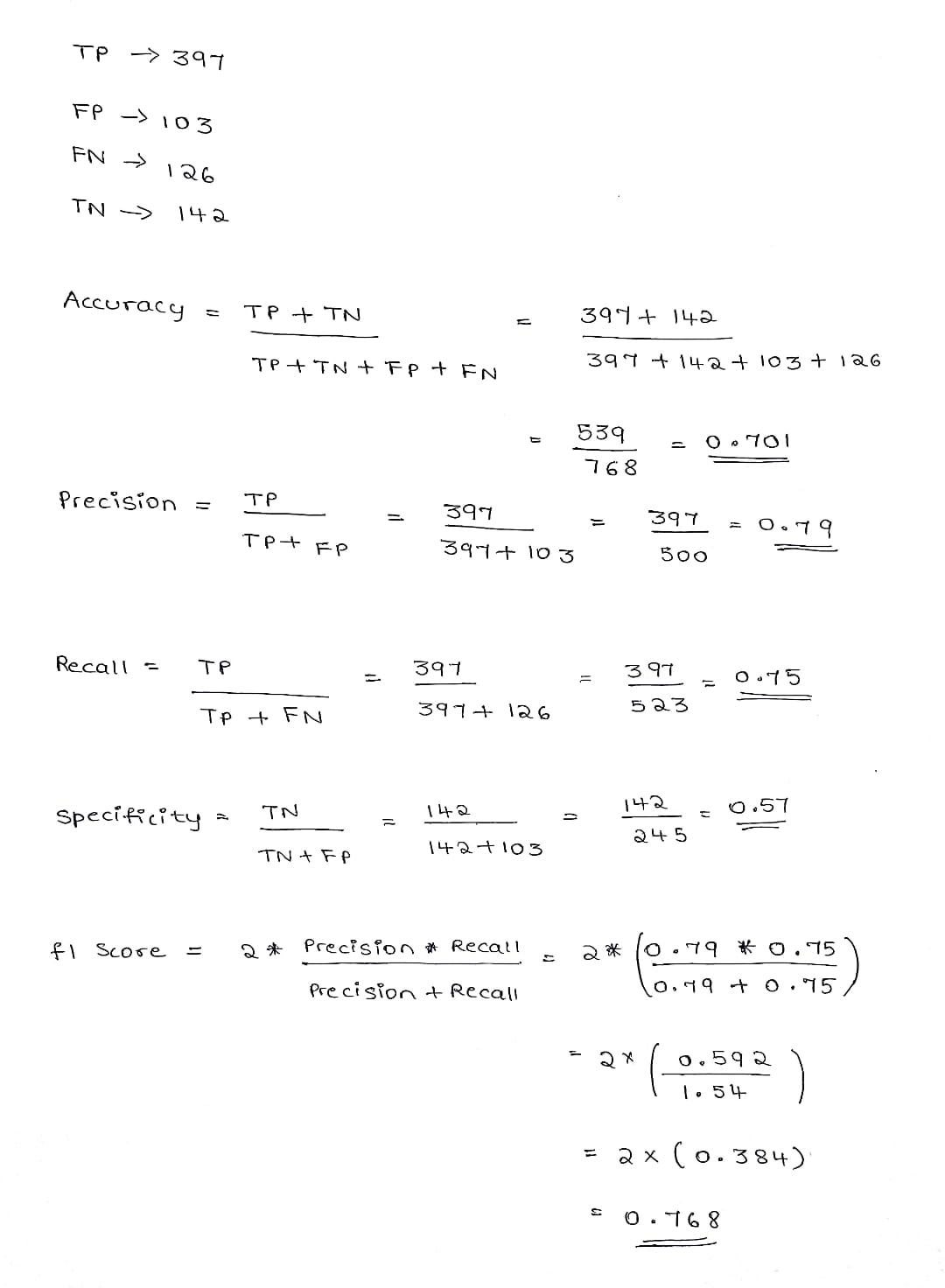


1.A) Explain the working and hyper parameters of the decision tree classifier.



B) A Machine learning model was built to classify patient as covid +ve or -ve. The confusion matrix for the model is as shown below. Compute other performance metrics and analyse the performance of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | | |
| Predicted |  | 1 | 0 |
| 1 | 397 | 103 |
| 0 | 126 | 142 |

****

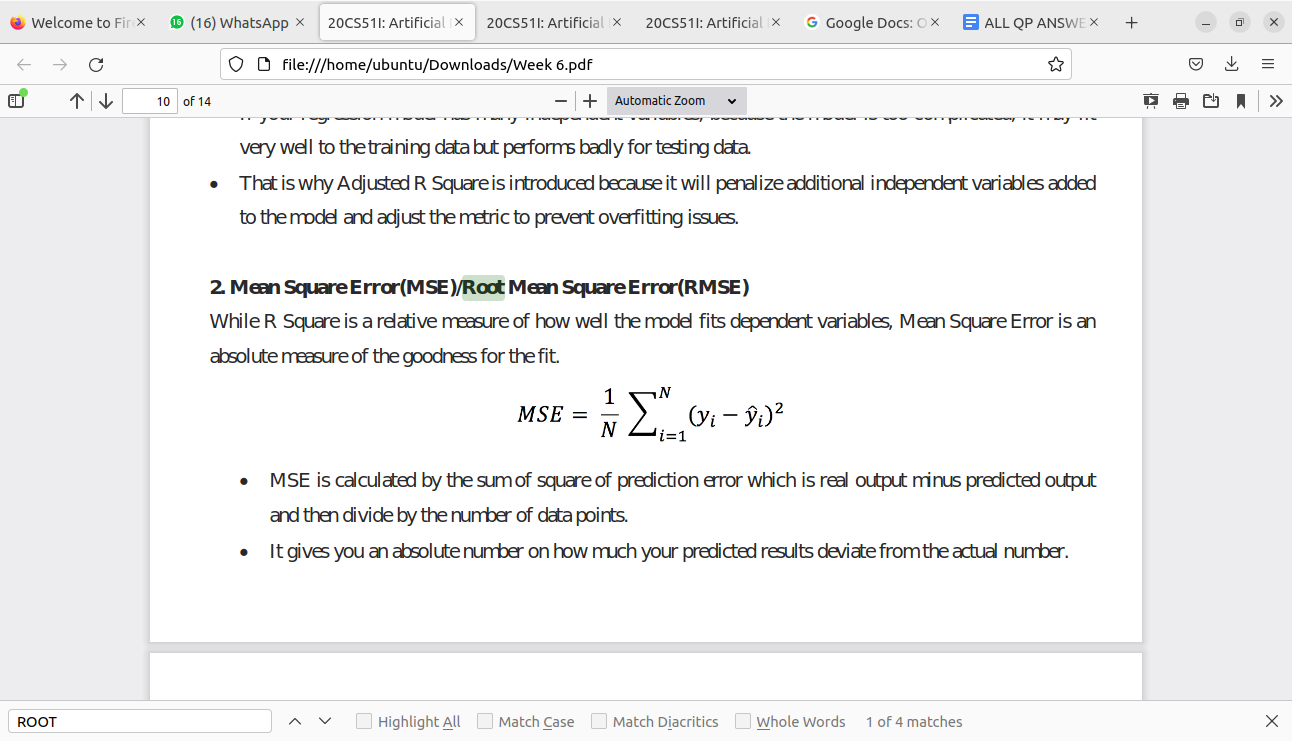
2A) Explain following evaluation metrics.

1. R.Squard

* R Square measures how much variability in dependent variable can be explained by the model. It is the square of the Correlation Coefficient(R) and that is why it is called R Square.
* R Square is calculated by the sum of squared of prediction error divided by the total sum of the square which replaces the calculated prediction with mean.
* R Square value is between 0 to 1 and a bigger value indicates a better fit between prediction and actual value.
* R Square is a good measure to determine how well the model fits the dependent variables.
* However, it does not take into consideration of overfitting problem.
* If your regression model has many independent variables, because the model is too complicated, it may fit very well to the training data but performs badly for testing data.
* That is why Adjusted R Square is introduced because it will penalize additional independent variables added to the model and adjust the metric to prevent overfitting issues.

1. Root mean squared error

* While R Square is a relative measure of how well the model fits dependent variables, Mean Square Error is an absolute measure of the goodness for the fit.



* MSE is calculated by the sum of square of prediction error which is real output minus predicted output and then divide by the number of data points.
* It gives you an absolute number on how much your predicted results deviate from the actual number.
* You cannot interpret many insights from one single result but it gives you a real number to compare against other model results and help you select the best regression model.
* Root Mean Square Error(RMSE) is the square root of MSE.
* It is used more commonly than MSE because firstly sometimes MSE value can be too big to compare easily.
* Secondly, MSE is calculated by the square of error, and thus square root brings it back to the same level of prediction error and makes it easier for interpretation.

1. Gradient descent

* Linear regression is about finding the line of best fit for a dataset. This line can then be used to makepredictions.
* Gradient descent is a tool to arrive at the line of best fit
* In gradient descent, you start with a random line. Then you change the parameters of the line (i.e. slope and y-intercept) little by little to arrive at the line of best fit.
* How do you know when you arrived at the line of best fit?
* For every line you try — line A, line B, line C, etc — you calculate the sum of squares of the errors. If lineB has a smaller value than line A, then line B is a better fit, etc.
* **Gradient descent is an algorithm that approaches the least squared regression line via minimizing sum of squared errors through multiple iterations.**
* **At a high level, this is how gradient descent works:**

* You start with a random line, let’s say line A. You compute the sum of squared errors for that line.
* Then, you adjust your slope and y-intercept.
* You compute the sum of squared errors again for your new line.
* You continue adjusting until you reach a local minimum, where the sum of squared errors is the smallest and additional tweaks does not produce better result.
* The way you adjust your slope and intercept will be covered in more details momentarily.

B) Explain the evaluation metrics for classification

The most important task in building any machine learning model is to evaluate its performance.Evaluation metrics are tied to machine learning tasks.Using different metrics for performance evaluation, we should be able to improve our model’s overall predictive power before we roll it out for production on unseen data.

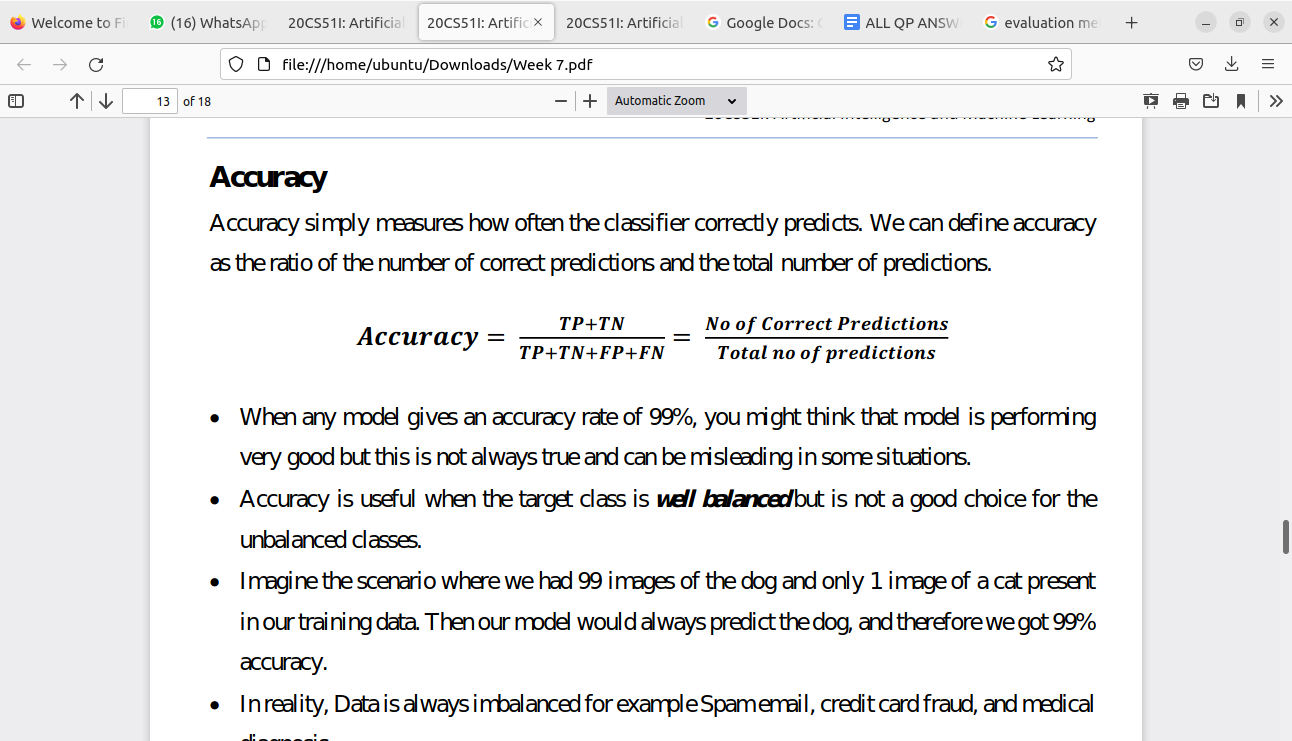
1. **Confusion Matrix**

Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values.

* A confusion matrix is defined as the table that is often used to describe the performance of a classification model on a set of the test data for which the true values are known.
* It is extremely useful for measuring the Recall, Precision, Accuracy, and AUC-ROC curves.
* **True Positive**: We predicted positive and it’s true. In the image, we predicted that a woman is pregnant and she actually is.
* **True Negative**: We predicted negative and it’s true. In the image, we predicted that a man is not pregnant and he actually is not.
* **False Positive (Type 1 Error)-** We predicted positive and it’s false. In the image, we predicted that a man is pregnant but he actually is not.
* **False Negative (Type 2 Error)**- We predicted negative and it’s false. In the image, we predicted that a woman is not pregnant but she actually is.

**2. Accuracy**

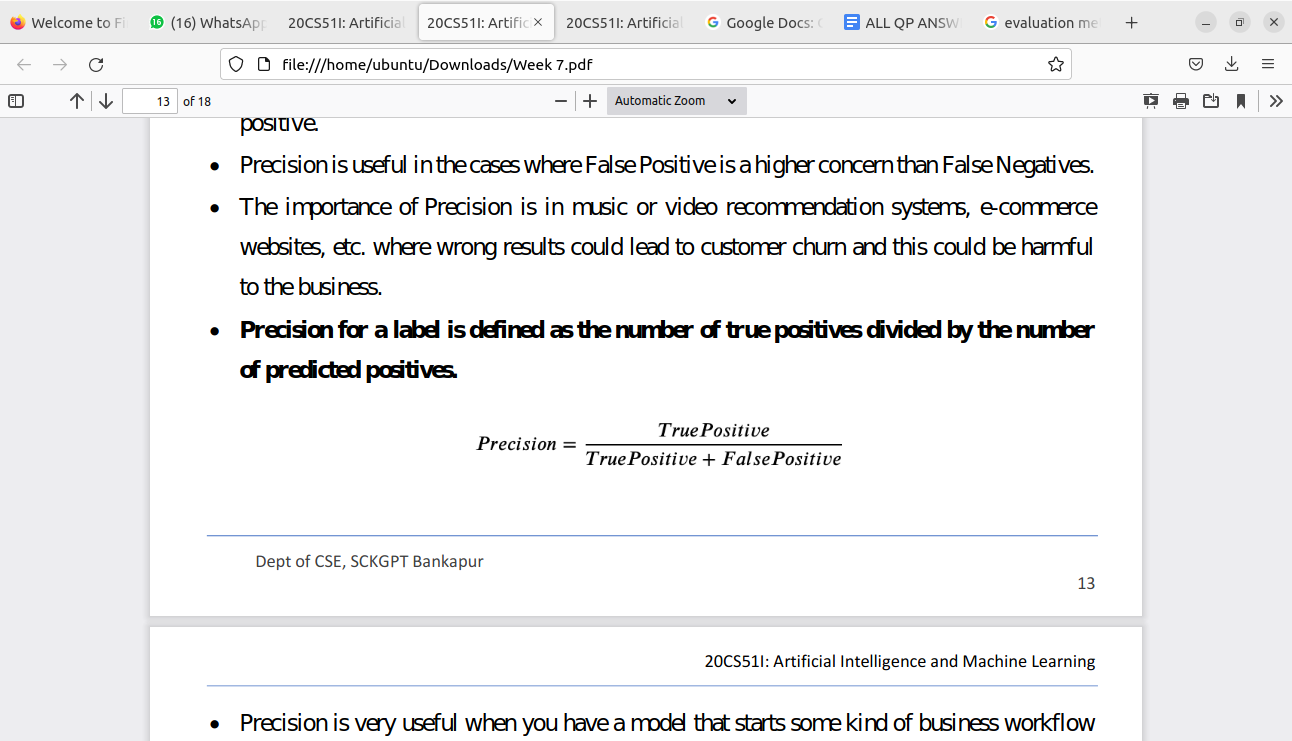
Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions.



* When any model gives an accuracy rate of 99%, you might think that model is performing very good but this is not always true and can be misleading in some situations.
* Accuracy is useful when the target class is well balanced but is not a good choice for the unbalanced classes.
* Imagine the scenario where we had 99 images of the dog and only 1 image of a cat present in our training data. Then our model would always predict the dog, and therefore we got 99% accuracy.
* In reality, Data is always imbalanced for example Spam email, credit card fraud, and medical diagnosis.
* Hence, if we want to do a better model evaluation and have a full picture of the model evaluation, other metrics such as recall and precision should also be considered

**3.Precision**

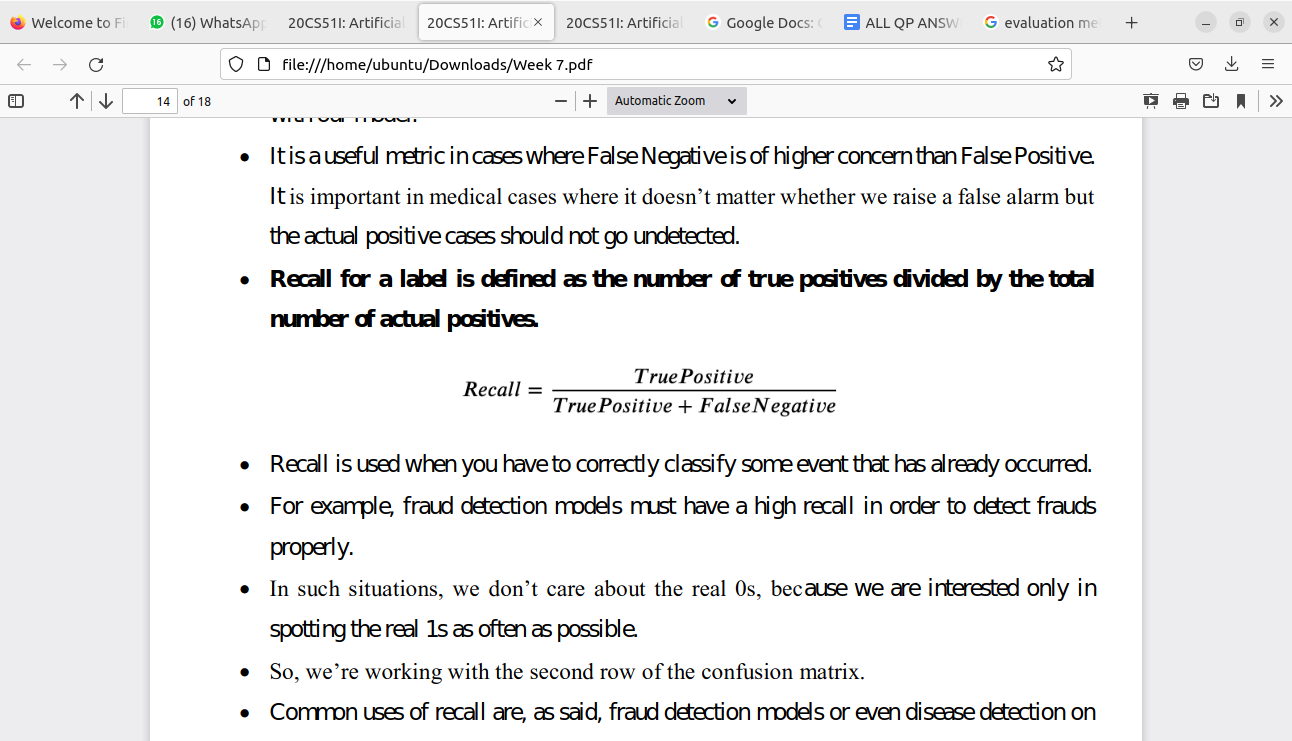
* Precision explains how many of the correctly predicted cases actually turned out to be positive.
* Precision is useful in the cases where False Positive is a higher concern than False Negatives.
* The importance of Precision is in music or video recommendation systems, e-commerce websites, etc. where wrong results could lead to customer churn and this could be harmful to the business.
* **Precision for a label is defined as the number of true positives divided by the number of predicted positives.**

****

* Precision is very useful when you have a model that starts some kind of business workflow (e.g. marketing campaigns) when it predicts 1.So, you want your model to be as correct as possible when it says 1 and don’t care too much when it predicts 0.
* Precision is very much used in marketing campaigns, because a marketing automation campaign is supposed to start an activity on a user when it predicts that they will respond successfully. That’s why we need high precision, which is the probability that our model is correct when it predicts 1.
* Low values for precision will make our business lose money, because we are contacting customers that are not interested in our commercial offer.

**4. Recall (Sensitivity)**

* Recall explains how many of the actual positive cases we were able to predict correctly with our model.
* It is a useful metric in cases where False Negative is of higher concern than False Positive.
* It is important in medical cases where it doesn’t matter whether we raise a false alarm but the actual positive cases should not go undetected.
* **Recall for a label is defined as the number of true positives divided by the total number of actual positives.**

****

* Recall is used when you have to correctly classify some event that has already occurred.For example, fraud detection models must have a high recall in order to detect fraud properly.In such situations, we don’t care about the real 0s, because we are interested only in spotting the real 1s as often as possible.
* So, we’re working with the second row of the confusion matrix.
* Common uses of recall are, as said, fraud detection models or even disease detection on a patient. If somebody is ill, we need to spot their illness avoiding the false negatives.
* A false negative patient may become contagious and it’s not safe. That’s why, when we have to spot an event that already occurred, we need to work with recall.

**5.Specificity (True negative rate)**

Specificity (SP) is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR).

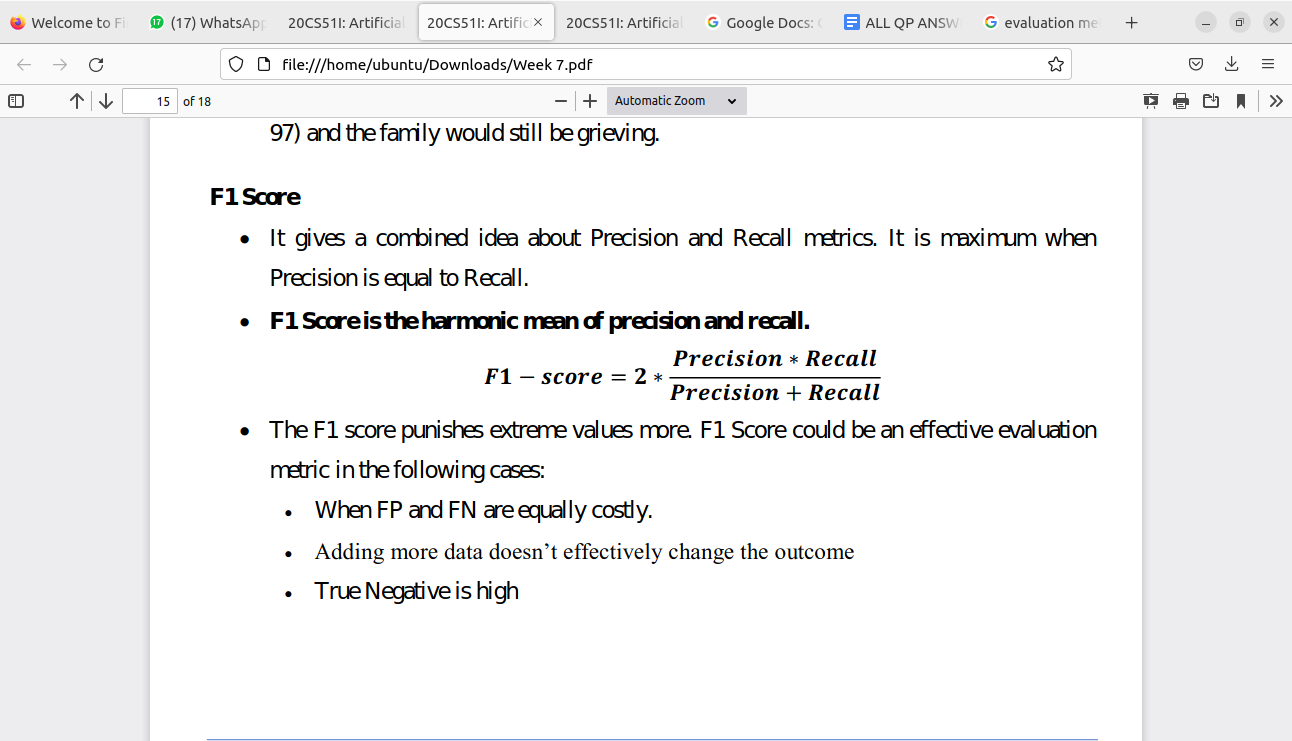
This metric is of interest if you are concerned about the accuracy of your negative rate and there is a high cost to a positive outcome.

* Let’s say we wanted to send a handwritten note to the family of each passenger who died as identified by our model for titanic dataset.
* Since the Titanic sunk in 1912, we feel the families have had time to heal from their loss and so would not be distraught by receiving a note.
* However, we feel it would be incredibly insensitive to send a note to a family of a survivor, as their death would have been more recent (the last Titanic survivor died in 2009 at age 97) and the family would still be grieving.

**6.F1 Score**

It gives a combined idea about Precision and Recall metrics. It is maximum when Precision is equal to Recall.

F1 Score is the harmonic mean of precision and recall.

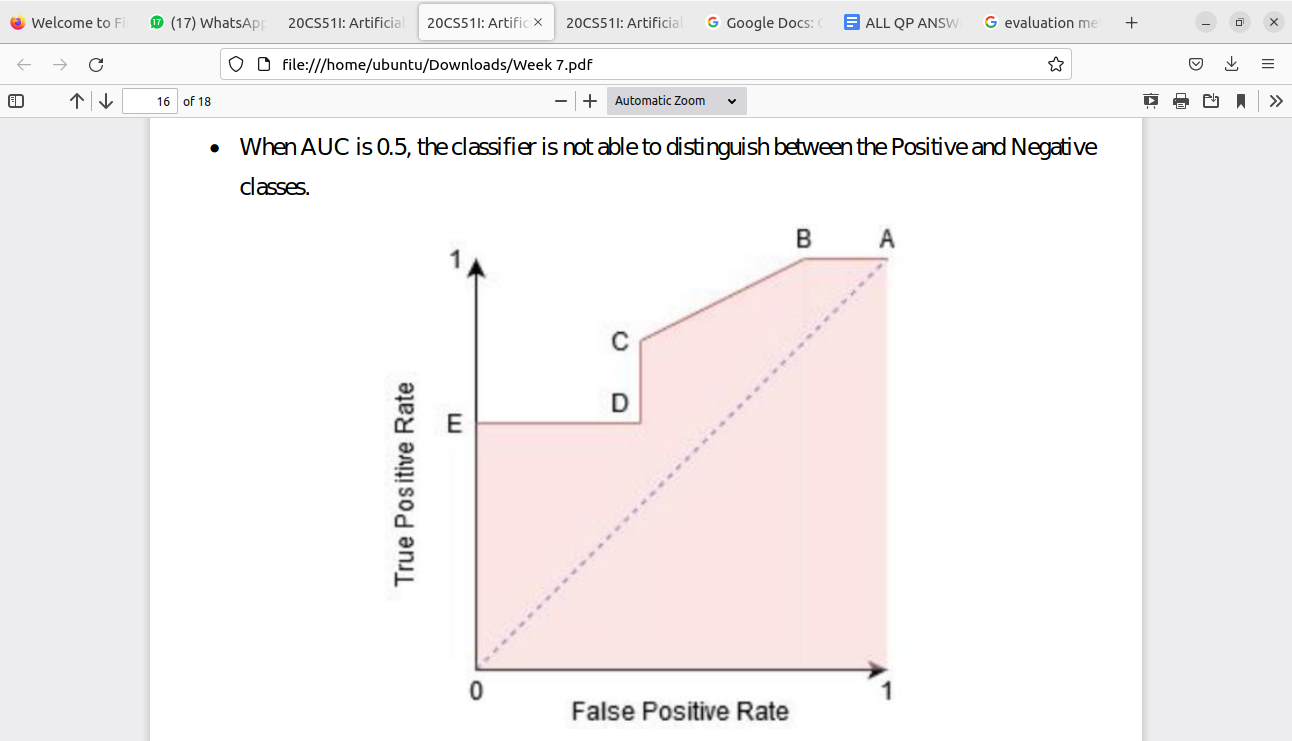


The F1 score punishes extreme values more. F1 Score could be an effective evaluation metric in the following cases:

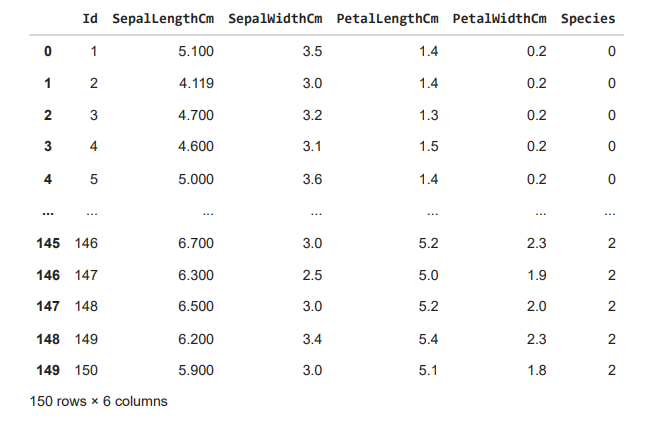
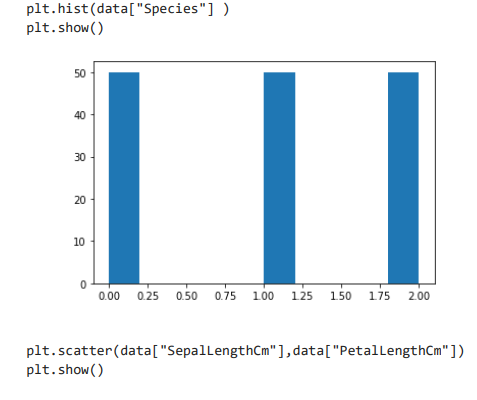
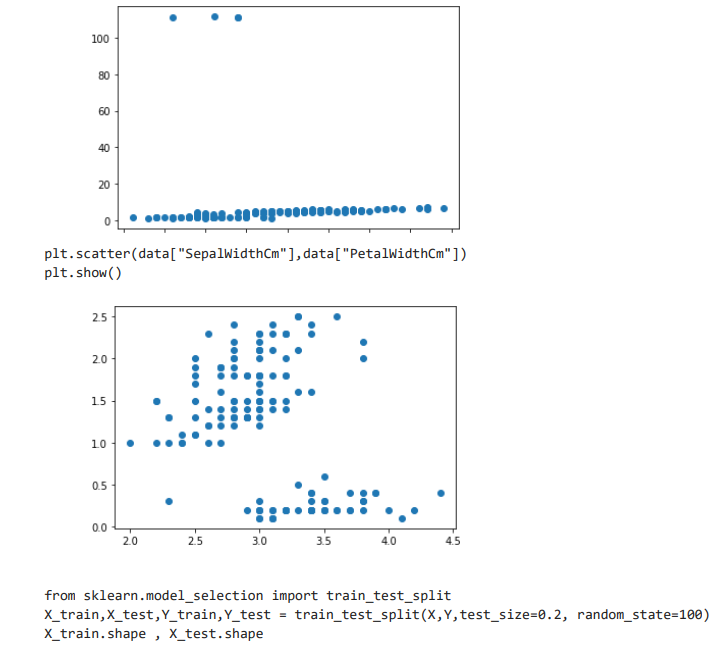
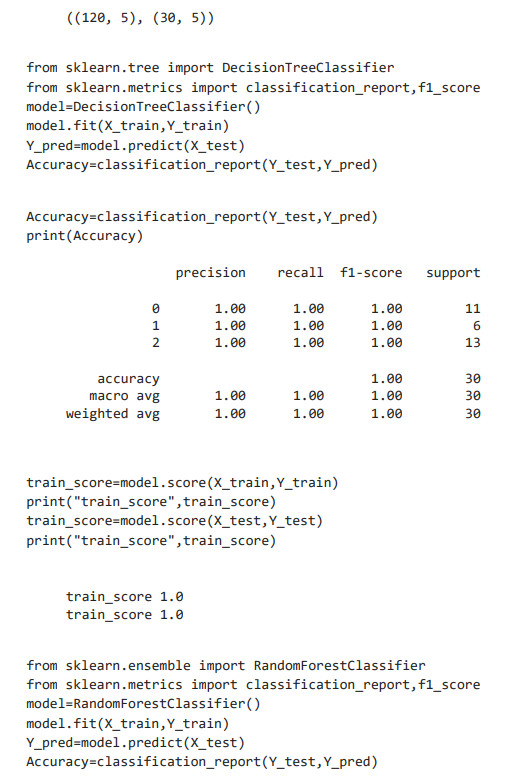
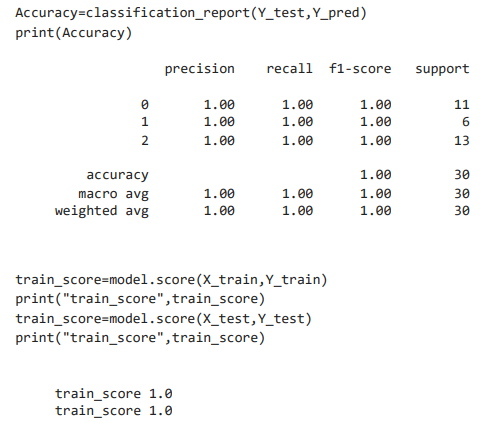
* When FP and FN are equally costly.
* Adding more data doesn’t effectively change the outcome True Negative is high

**7.AUC-ROC**

* The Receiver Operator Characteristic (ROC) is a probability curve that plots the TPR(True Positive Rate) against the FPR(False Positive Rate) at various threshold values and separates the ‘signal’ from the ‘noise’.
* The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes. From the graph, we simply say the area of the curve ABDE and the X and Y-axis.
* From the graph shown below, the greater the AUC, the better is the performance of the model at different threshold points between positive and negative classes.
* This simply means that When AUC is equal to 1, the classifier is able to perfectly distinguish between all Positive and Negative class points.
* When AUC is equal to 0, the classifier would be predicting all Negatives as Positives and vice versa.
* When AUC is 0.5, the classifier is not able to distinguish between the Positive and Negative classes.



1. *Build decision tree classifier and random forest classifier model using Scikit* learn to iris dataset. Perform data exploration, preprocessing splitting and evaluate metrics.

1. Build multiple linear regression model to house dataset. Perform data exploration, preprocessing splitting and evaluate the metrics R2, RMSE.



