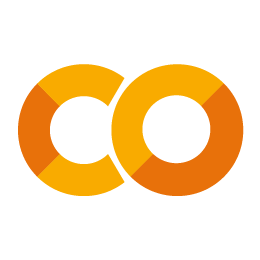
**CS 3120 - Machine Learning and**

**Neural Computing**

**Assignment 2 – Random Forest**







**Name : B.M.G.G.K. Rajapaksha**

**Index No : S14210**

**Faculty : Faculty of Science (Bioinformatics)**

**Contents**

**Introduction ………………………………………………………………………….. 3**

* **Decision trees ………………………………………………………………… 3**
* **Random forest ……………………………………………………………….. 3**
* **Why we use Random forest………………………………………………….. 4**

**Objectives ……………………………………………………………………………. 5**

**Dataset ……………………………………………………………………………….. 5**

**Problem ……………………………………………………………………………… 6**

**Approach ……………………………………………………………………………. 7**

**Methodology and steps ……………………………………………………………... 7**

**Discussion …………………………………………………………………………… 13**

**Conclusion ………………………………………………………………………….. 13**

**Colab notebook …………………………………………………………………….. 13**

**References …………………………………………………………………………... 13**

**Introduction**

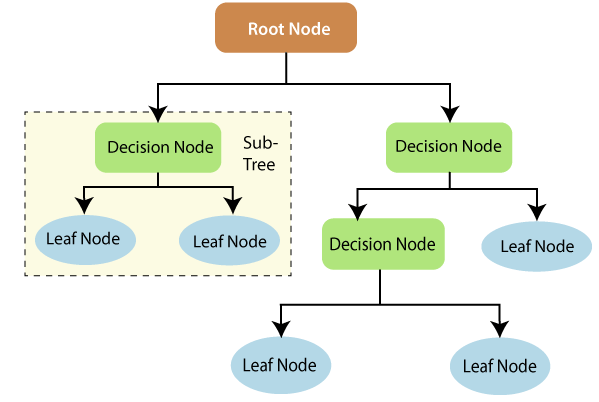
Machine learning is a part of artificial intelligence which focuses on the use of data and algorithms to imitate the way that humans learn to improve the accuracy of the problem solving by using computer science. Machine learning can be classified into three basic types as supervised learning, unsupervised learning and reinforcement learning.

Supervised learning which uses labelled datasets to train algorithms, falls into two groups as classification and regression. Classification is a kind of problem where in the outputs are categorical in nature like yes or no true or false or zero or one. In that particular framework there are the K-NN, Naive Bayes, the decision tree, and the random forest.

Though the Random forest and decision tree belong to the supervised classification learning, this can be used for regression tasks as well.

* **Decision trees**

Decision trees are one of the very basic type of machine learning model. It is a tree shaped structure and this hierarchical structure can be used to learn and understand a dataset.

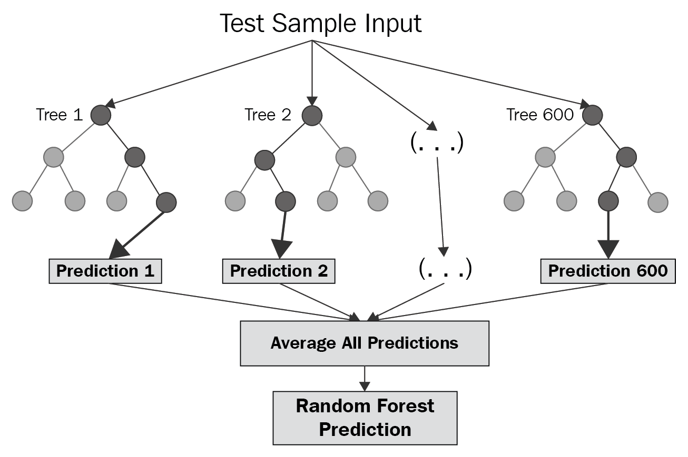


In this approach, first construct the decision tree structure based on the available set of features/attributes present in the dataset and later use this tree to classify unforeseen data points.

* **Random forest**

Random forest which is one of the most efficient classifiers in the domain of machine learning. It is a method that operates by constructing multiple decision trees. It is called as random forest because it is a forest of randomly created decision trees.

When training, each tree in a random forest learns from a random sample of the data points. The samples are drawn with replacement, known as bootstrapping. In bootstrapping some samples will be used multiple times in a single tree.



Each node in the decision tree works on a random subset of features to calculate the output. The random forest then combines the output of randomly created individual decision trees to generate the final output. The decision of the majority of the trees is the final decision.

* **Why we use random forest?**

The decision tree algorithm is quite easy to understand and interpret. But often, a single tree is not sufficient for produce effective results. This is where the Random Forest algorithm comes into the picture.

Random forest algorithm has more benefits and advantages over decision tree algorithm and other classification algorithms as follows.

There is no overfitting

Overfitting means that the model closely fits to the training data but fails to generalize the testing unforeseen data. This might happens in decision tree algorithm. But use of multiple trees in the random forest algorithm reduces the risk of overfitting.

High accuracy

Random forest algorithm produces highly accurate predictions even for large datasets. It runs efficiency in large databases.

Estimates missing data

Random forest algorithm can maintain the accuracy when a large proportion of the data is missing. It handles this by There are two ways to handle this by;

1. Using median values to replace the missing continuous variables.
2. Computing the proximity-weighted average of missing values.

Does not rely on the feature importance given by a single decision tree

Since the decision tree model gives high importance to a particular set of features, the random forest chooses features randomly during the training process. Therefore, it does not depend highly on any specific set of features.

The algorithm can be used in both classification and regression problems

Applicable for both Numerical and Categorical data

\*Random forest algorithm is not simple as compared to the decision tree algorithm.

Since the algorithm uses multiple decision trees to generate predictions, is becomes relatively slow. Hence sometimes it is time consuming than a single decision tree.

**Objective**

To demonstrate how to use the random forest classifier to solve the classification problem with high accuracy against a large dataset which was obtained from the UCI machine learning repository.

**Dataset**

**Name of the dataset :** Car evaluation database

**Repository**  : UCI Machine Learning Repository

**Creator** : Marko Bohanec in 1997

**Dataset Characteristics :** Multivariate categorical

**Number of Attributes** : 6

**Number of Instances** : 1728

**Missing Attribute Values** : none

Car evaluation database is a multivariate categorical database which can be used to evaluate and classify cars according to the following concept structure:

**Overall price**

* Buying price - Very high, High, Med, Low
* Maintenance price - Very high, High, Med, Low

**Technical characteristics**

* Comfort

\*No. of doors - 2, 3, 4, 5 or more

\*Capacity in terms of persons to carry - 2, 4, more

\*The size of luggage boot - Small, Big, Med

* Estimated safety of the car – Low, Med, High

For the easiness, creator selected “Buying Price”, “Maintenance price”, “No. of doors”, “No. of persons”, “Size of the luggage boot”, “Estimated safety” as variables/attributes/features and the “Car acceptability” level as the target class as follows.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Buying price | Price of the maintenance | No. of doors | No. of persons | Size of the luggage boot | Estimated safety | Car acceptability |
| 1 | vhigh | low | 3 | more | small | med | unacc |
| 2 | vhigh | low | 3 | more | small | high | acc |
| 3 | vhigh | low | 3 | more | med | low | unacc |
| 4 | vhigh | low | 3 | more | med | med | acc |
| 5 | vhigh | low | 3 | more | med | high | acc |
| 6 | vhigh | low | 3 | more | big | low | unacc |
| 7 | vhigh | low | 3 | more | big | high | acc |
| 8 | vhigh | low | 4 | 2 | small | low | unacc |
| 9 | med | med | 2 | 4 | big | high | vgood |
| 10 | med | low | 2 | 4 | med | high | good |

* **Class Distribution**
* Unacc -1210 (70.023 %)
* Acc - 384 (22.222 %)
* good - 69 ( 3.993 %)
* vgood - 65 ( 3.762 %)
* Total - 1728

**Problem**

Since the dataset has more than 1500 of different instances of being unacc, acc, good or vgood car, it is impossible to predict the target class of the unforeseen new instances just looking at the current dataset. If it is possible to make a machine learning classification model (in this case, random forest algorithm) to classify each instances perfectly into their target classes, it is helpful for both the seller and buyer when they select a new car.

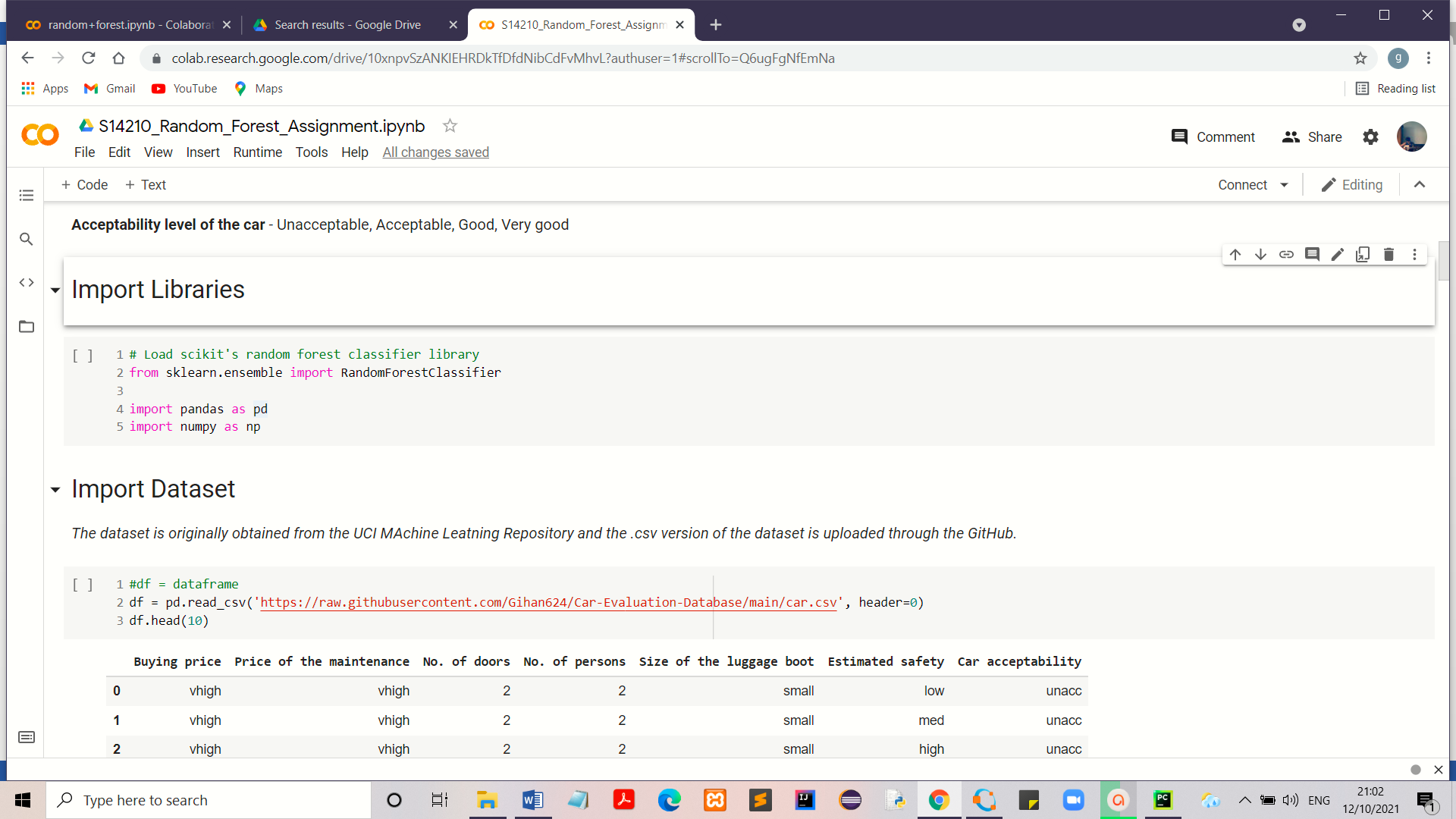
**Approach**

Since the dataset is large, random forest classifier is the best approach to solve the problem due to its high accuracy for the large datasets over the decision tree algorithm. Therefore in this scenario, My approach was to implement an accurate classification model for the problem by using random forest algorithm to predict the target classes for unforeseen data which were entered as input values.

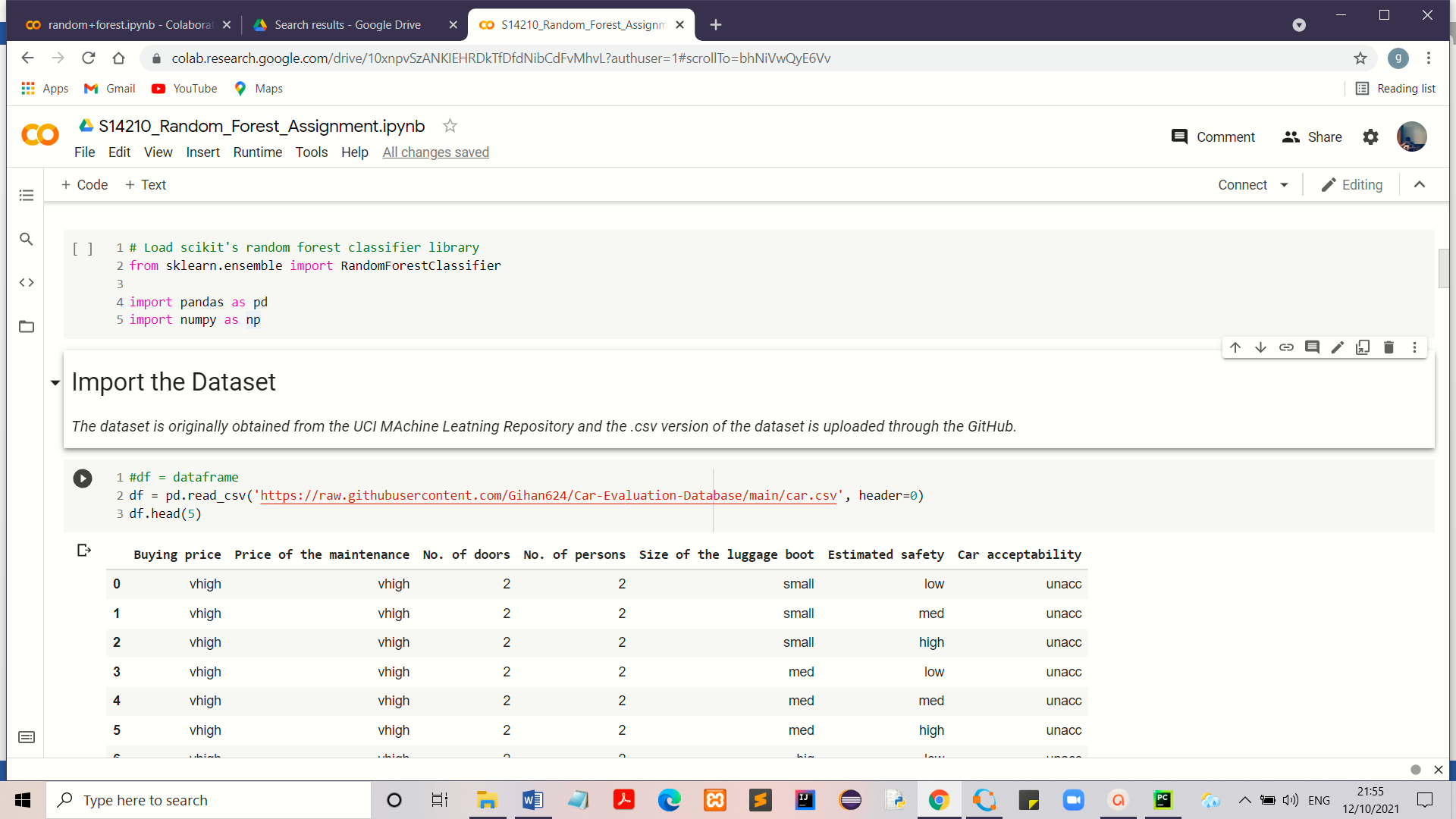
**Methodology and steps**

The dataset was initially downloaded from the UCI Machine Learning Repository as a text file. Then the Column names were added into the text file and the file was converted into a csv file. csv file was uploaded into the GitHub. Python language was selected to implement the program by using Google Colab web IDE.

**Step 1 - Import libraries**



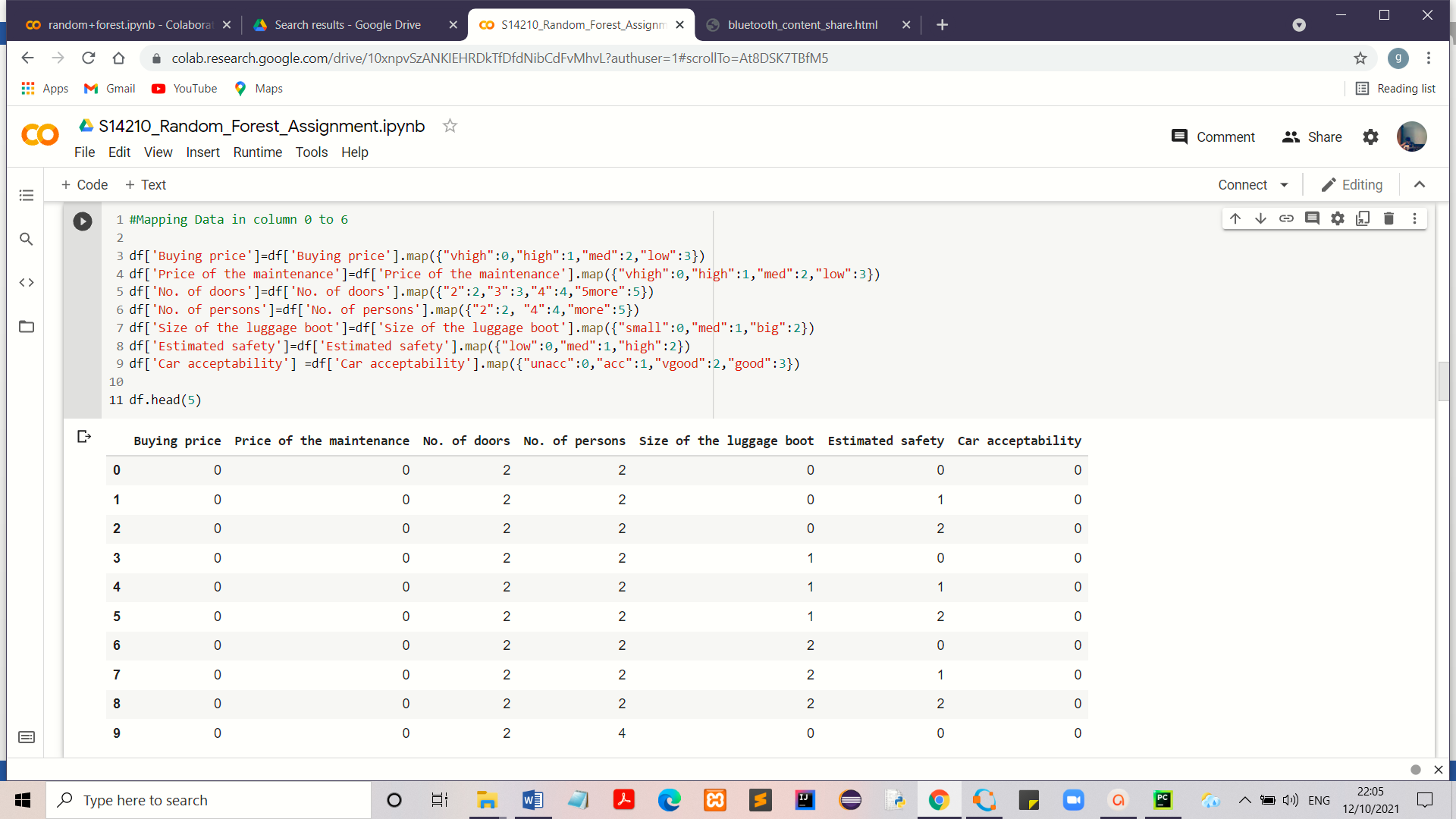
Sklearn (Scikit – learn) is a library in Python that provides many supervised and unsupervised machine learning algorithms. It also provides the random forest algorithm as “RandomForestClassifier” for classification tasks. Therefore this library was used to implement the random forest classification.

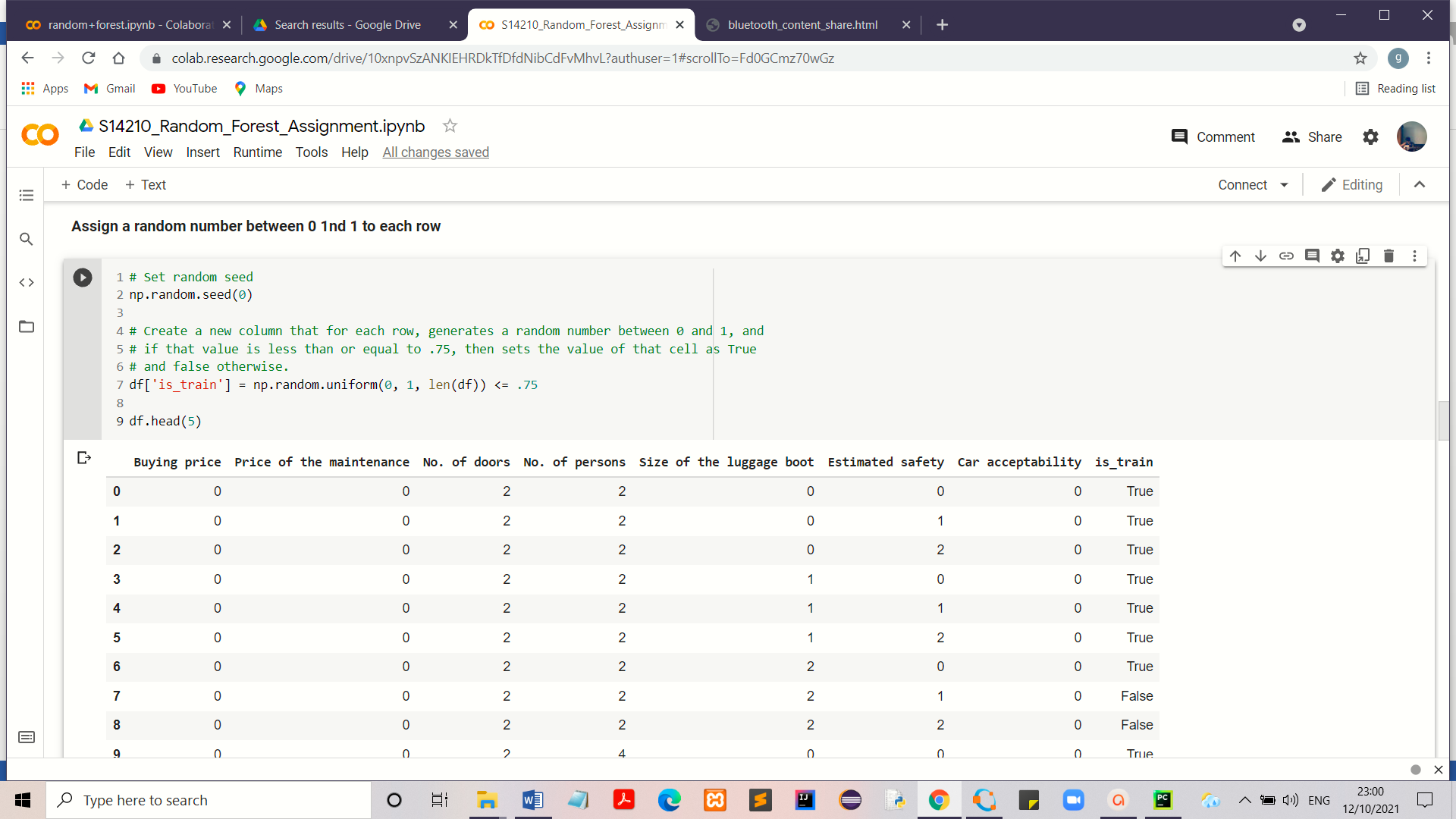
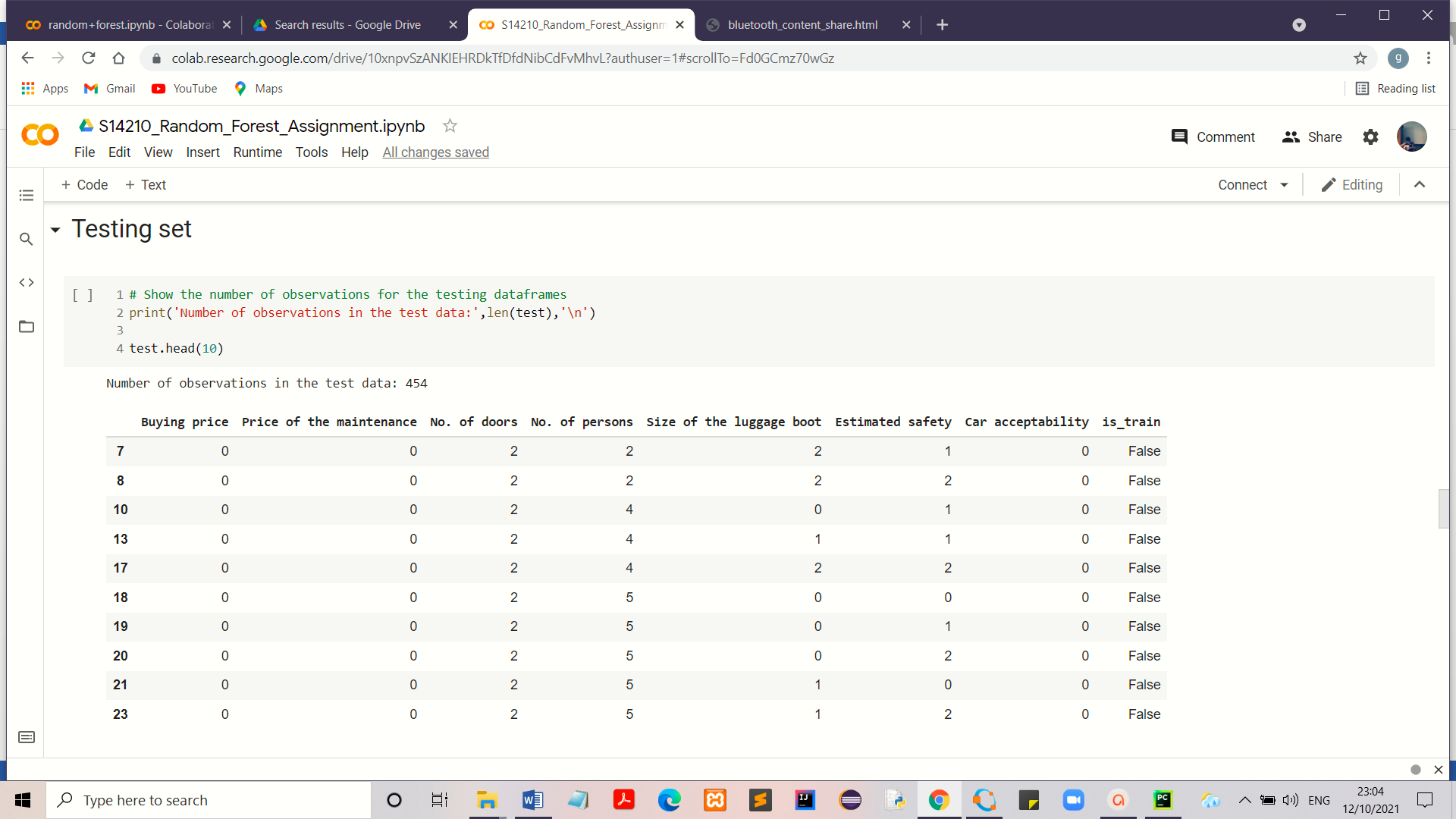
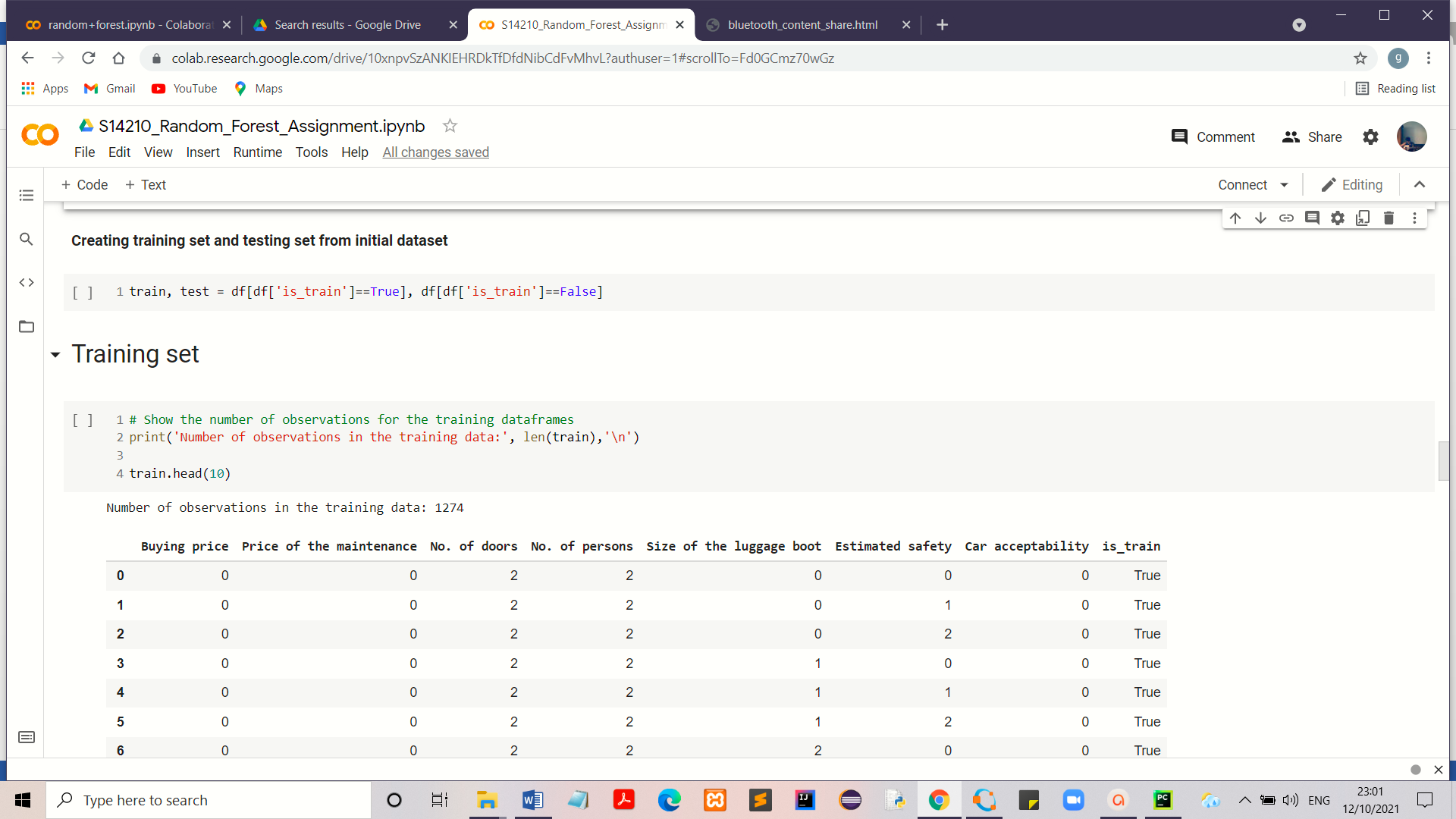
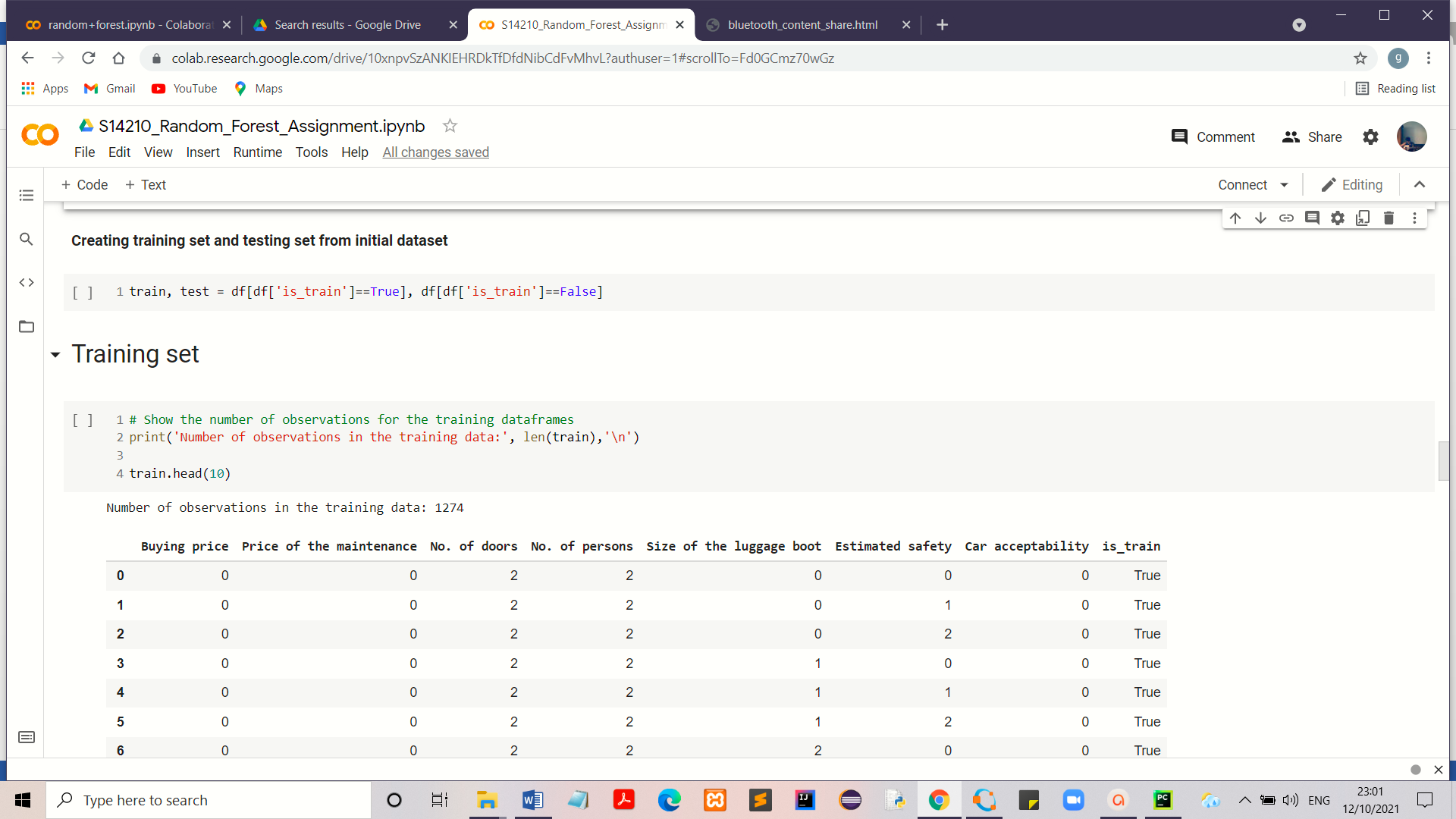
**Step 2 – Import Dataset from the GitHub**

**Step 3 – Mapping Data**

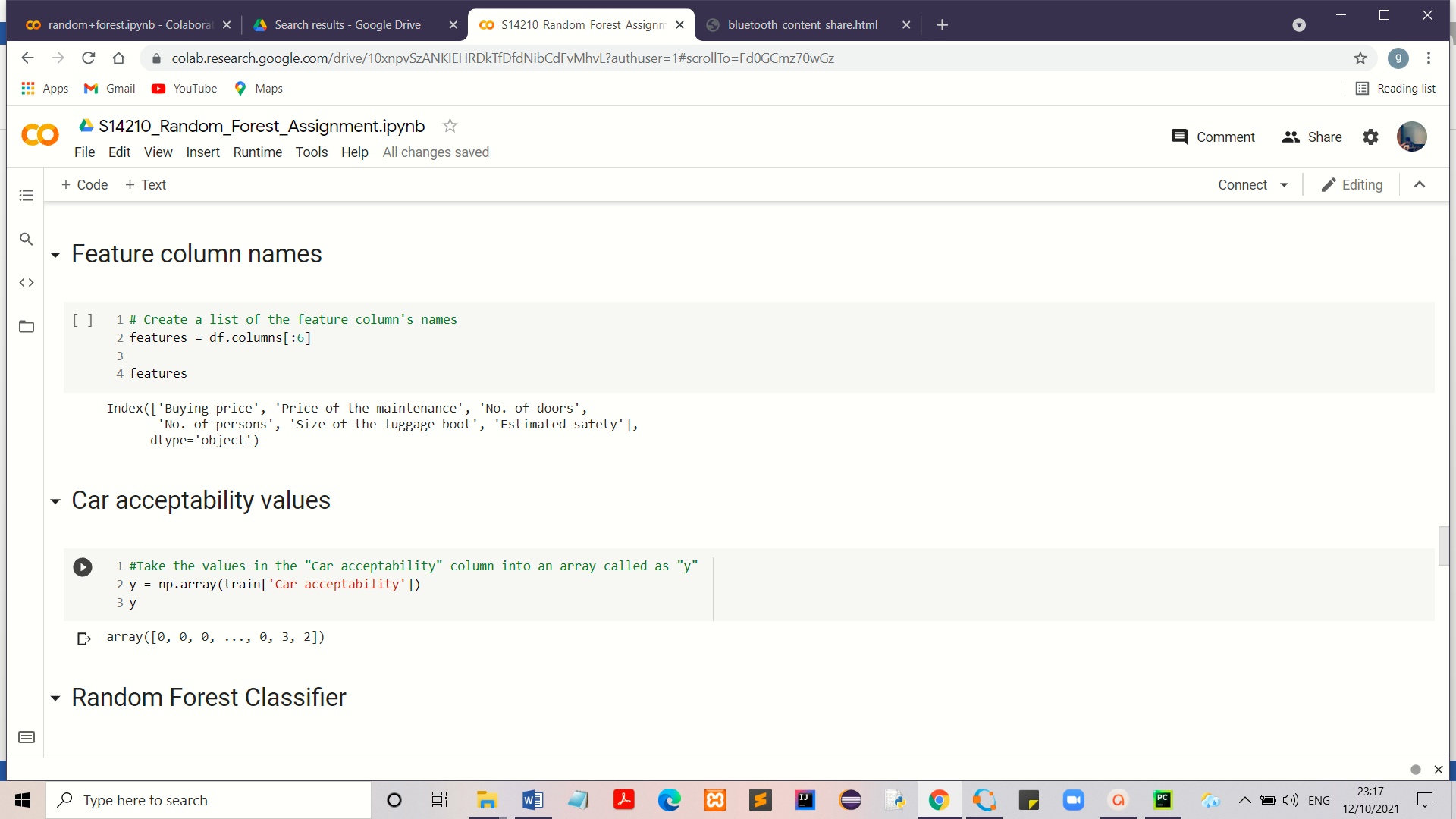
Though the decision trees can handle both categorical and numerical variables at the same time as features, the dataset was converted into numerical values due to the use of sklearn library which allows numerical data in order to perform machine learning tasks. Since the attributes in the dataset have ordinal variables, Integer encoding/label encoding was selected.

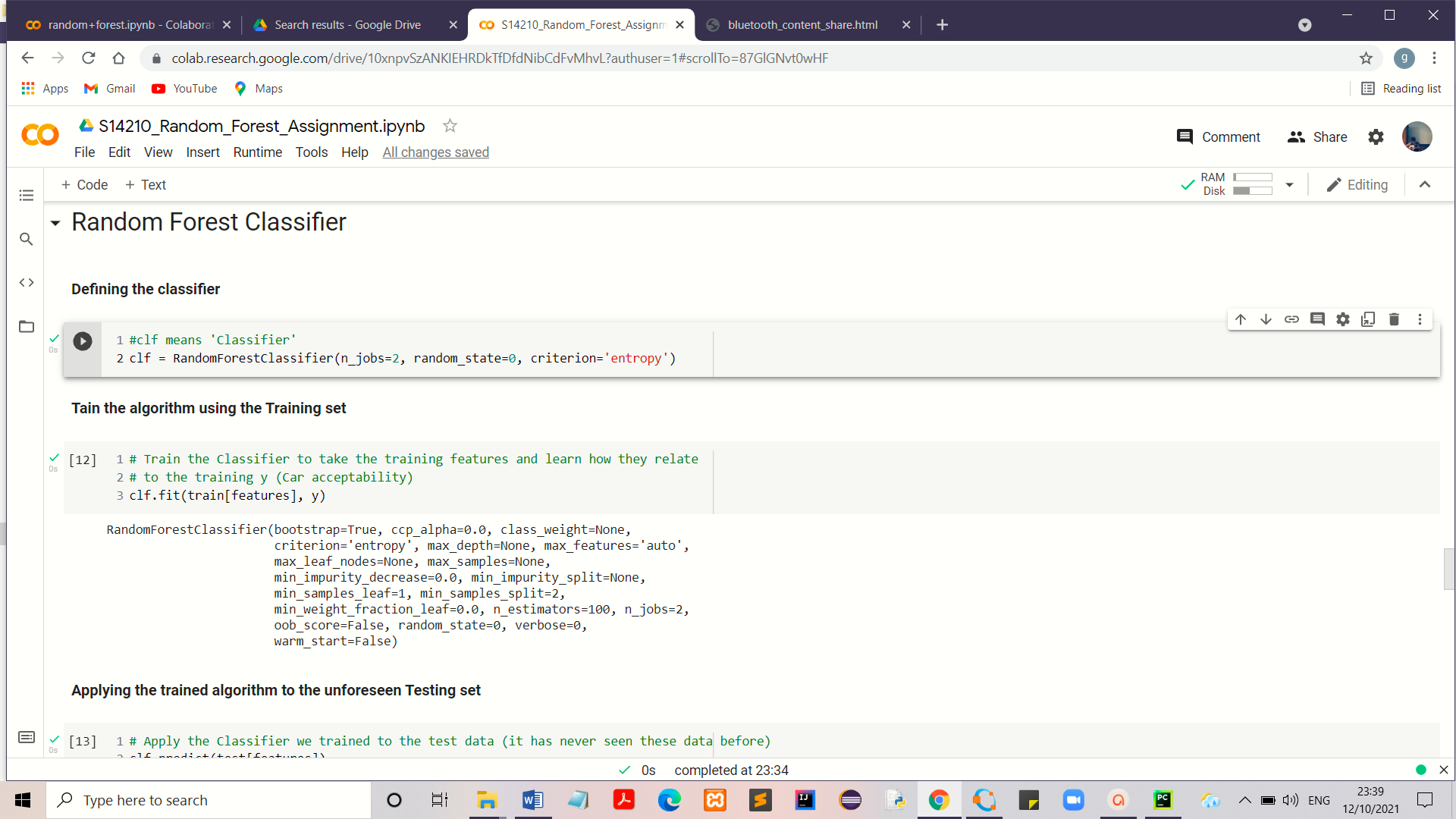
|  |  |  |
| --- | --- | --- |
| Column name | Categorical value | Numerical value |
| Buying price | vhigh | 0 |
|  | high | 1 |
|  | med | 2 |
|  | low | 3 |
| Price of the maintenance | vhigh | 0 |
|  | high | 1 |
|  | med | 2 |
|  | low | 3 |
| No. of doors | 2 | 2 |
|  | 3 | 3 |
|  | 4 | 4 |
|  | 5 or more | 5 |
| No. of persons | 2 | 2 |
|  | 4 | 4 |
|  | more | 5 |
| Size of the luggage boot' | small | 0 |
|  | med | 1 |
|  | big | 2 |
| Estimated safety | low | 0 |
|  | med | 1 |
|  | high | 2 |
| Car acceptability' | unacc | 0 |
|  | acc | 1 |
|  | vgood | 2 |
|  | good | 3 |

**Step 4 – Split the dataset into two groups as “Training set” and “Testing set”**

Since the dataset contains 1728 instances, it was split into two as training set and testing set into 4:3 ratio by randomly assigning a value for each row of the dataset between 0 and 1. If the value ≤ 0.75, the row consider as a training data otherwise a testing data. Training set was used to train the random forest algorithm while the testing set was used as the unforeseen data set to test the trained algorithm. Training set contains more data than the testing set in order to ensure the proper leaning process.

**Step 5 – Take feature column names into a list and target class values into an array**

Here all the columns except the “Car acceptability” column belongs to the feature columns. Car acceptability column is the column that has target class values.

**Step 6 – Define and train the random forest classifier**

Random forest classifier has several parameters. In here;

* Splitting criteria – Entropy (denoted by “criteria”)
* No. of trees in the forest – 100 trees (denoted by “n\_estimators”)
* Maximum features – Auto

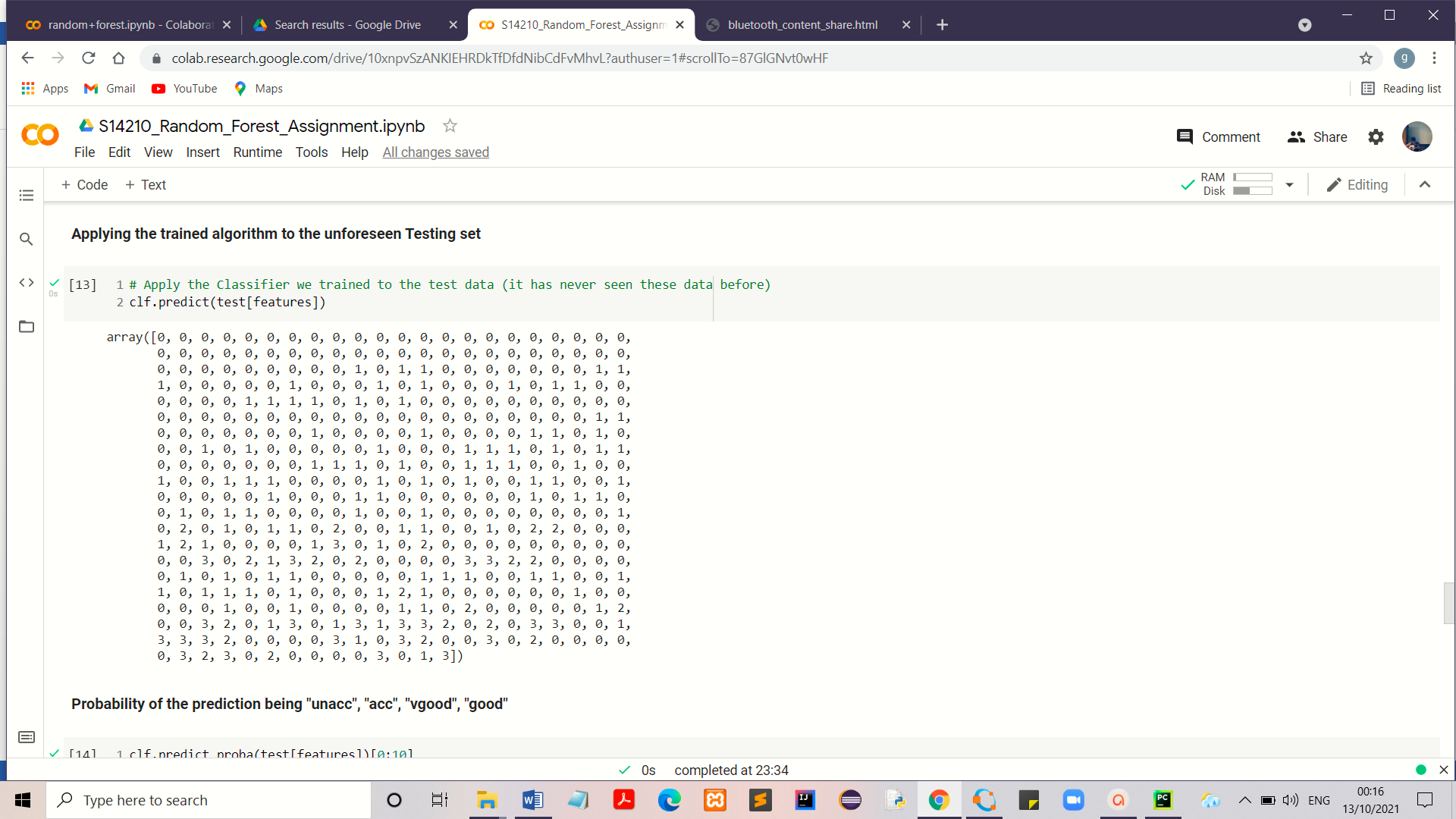
This represents the number of features to consider when looking for the split. By selecting it as “auto”, the maximum number of features going to be selected are equal to the square root of the total number of features (square root of 6).

* Maximum depth of the tree – null

Hence nodes are expended until all leaves are pure

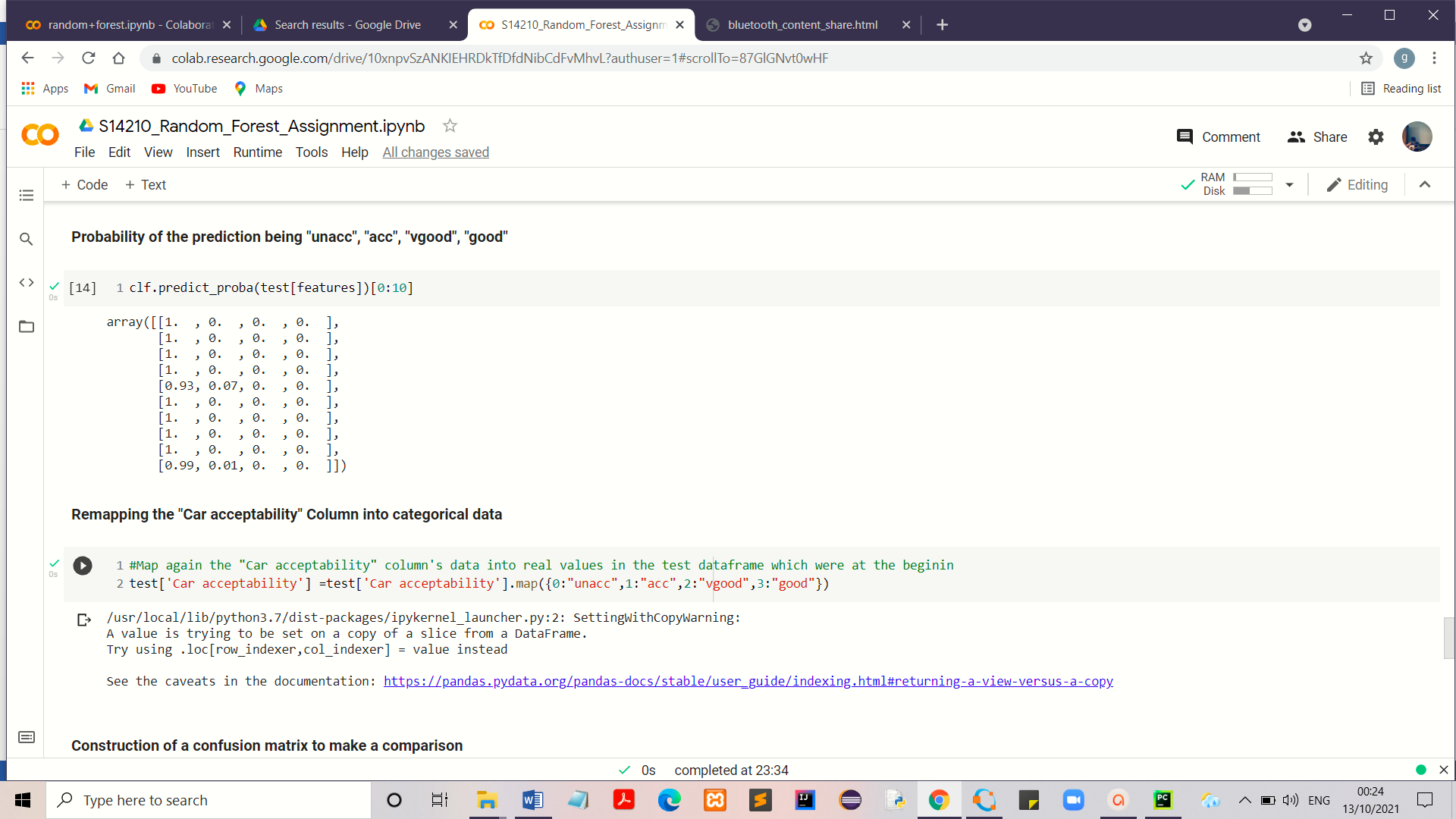
* Bootstrap – True

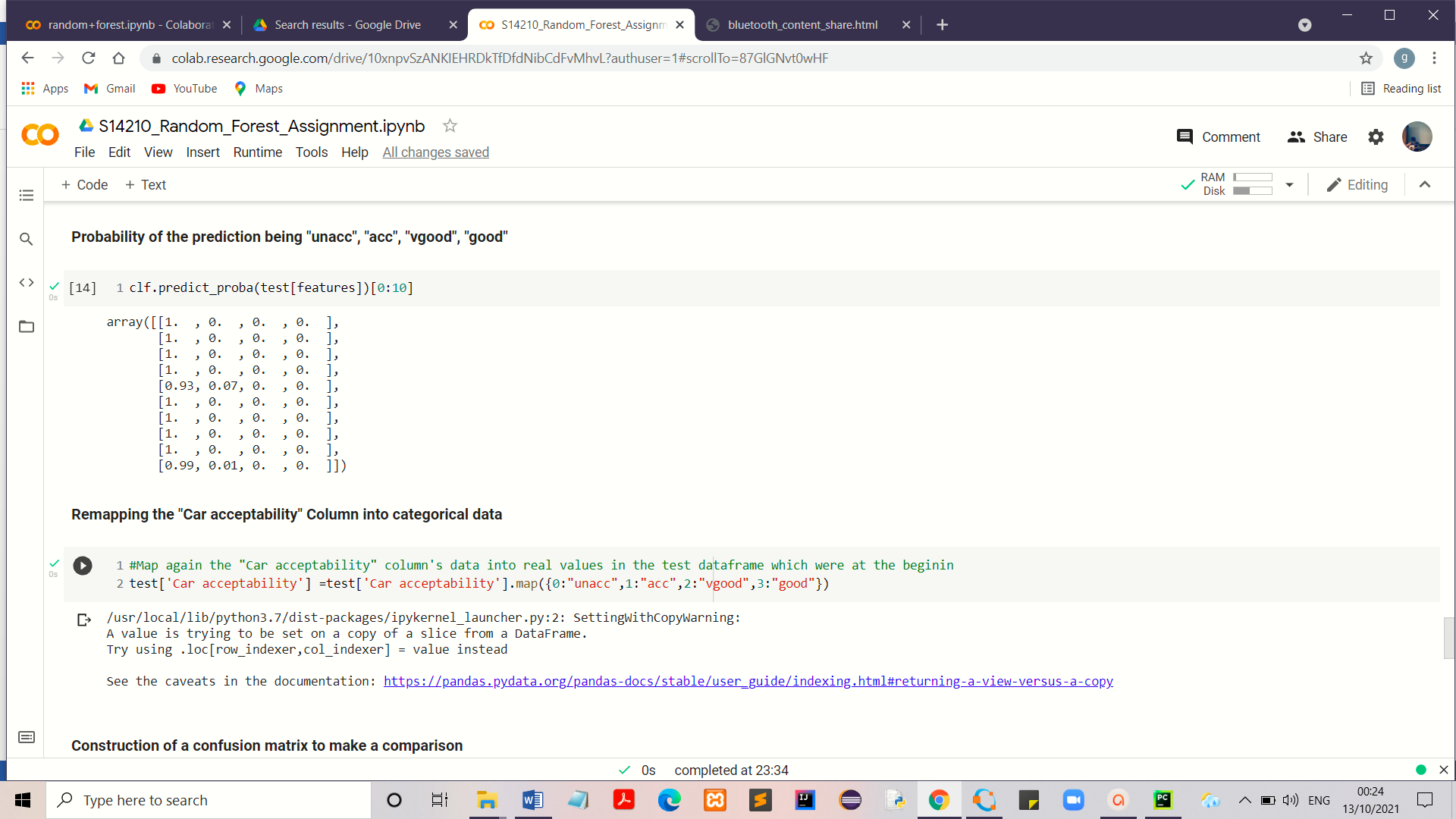
It shows whether bootstrap samples are used when building trees. If the samples are drawn with the replacement it known as bootstrapping. It helps to reduce the variance for the random forest algorithm. If bootstrap = False, the whole dataset is used to build each tree.

Step 7 – Applying the trained algorithm to the unforeseen Testing set

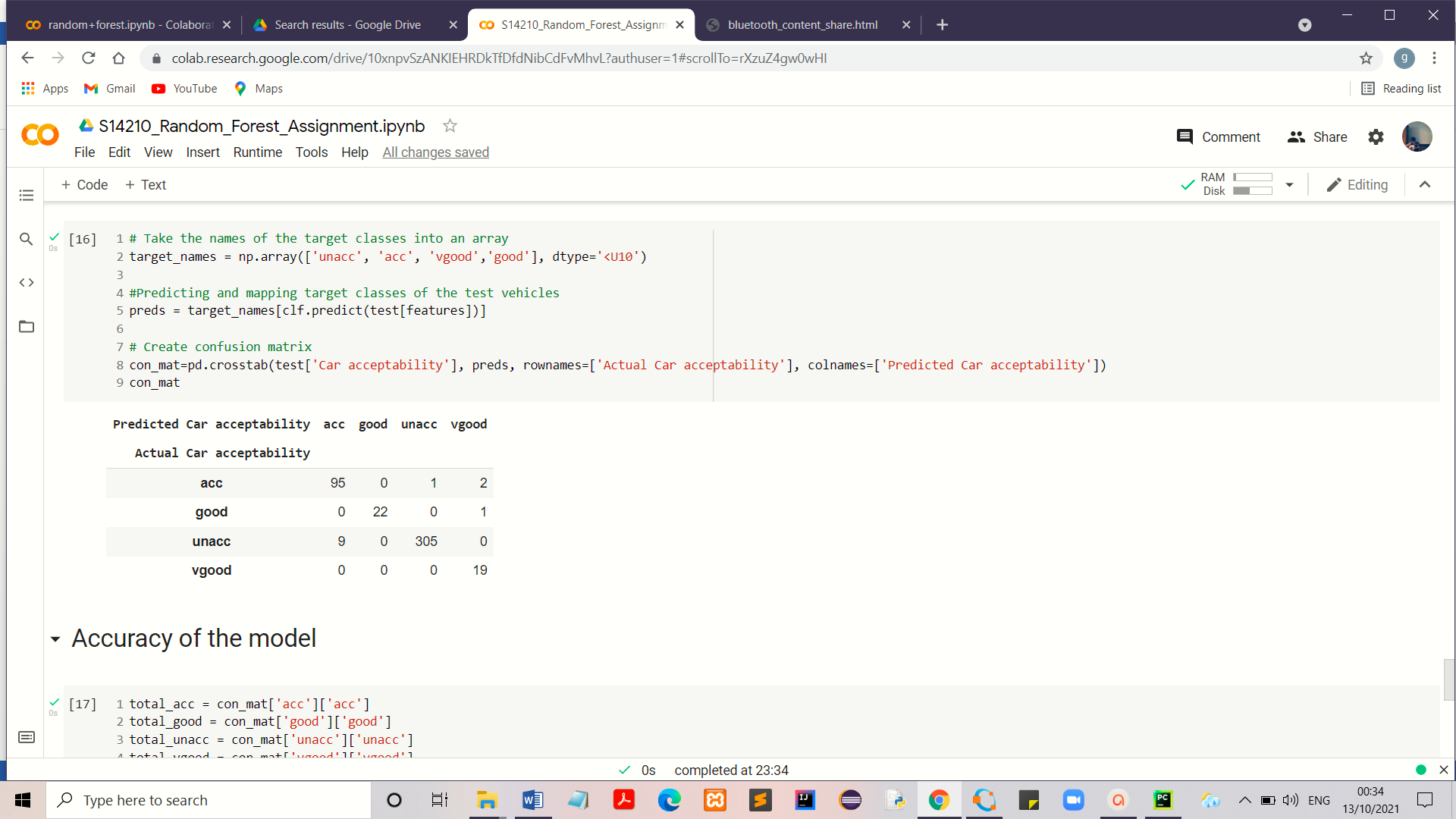
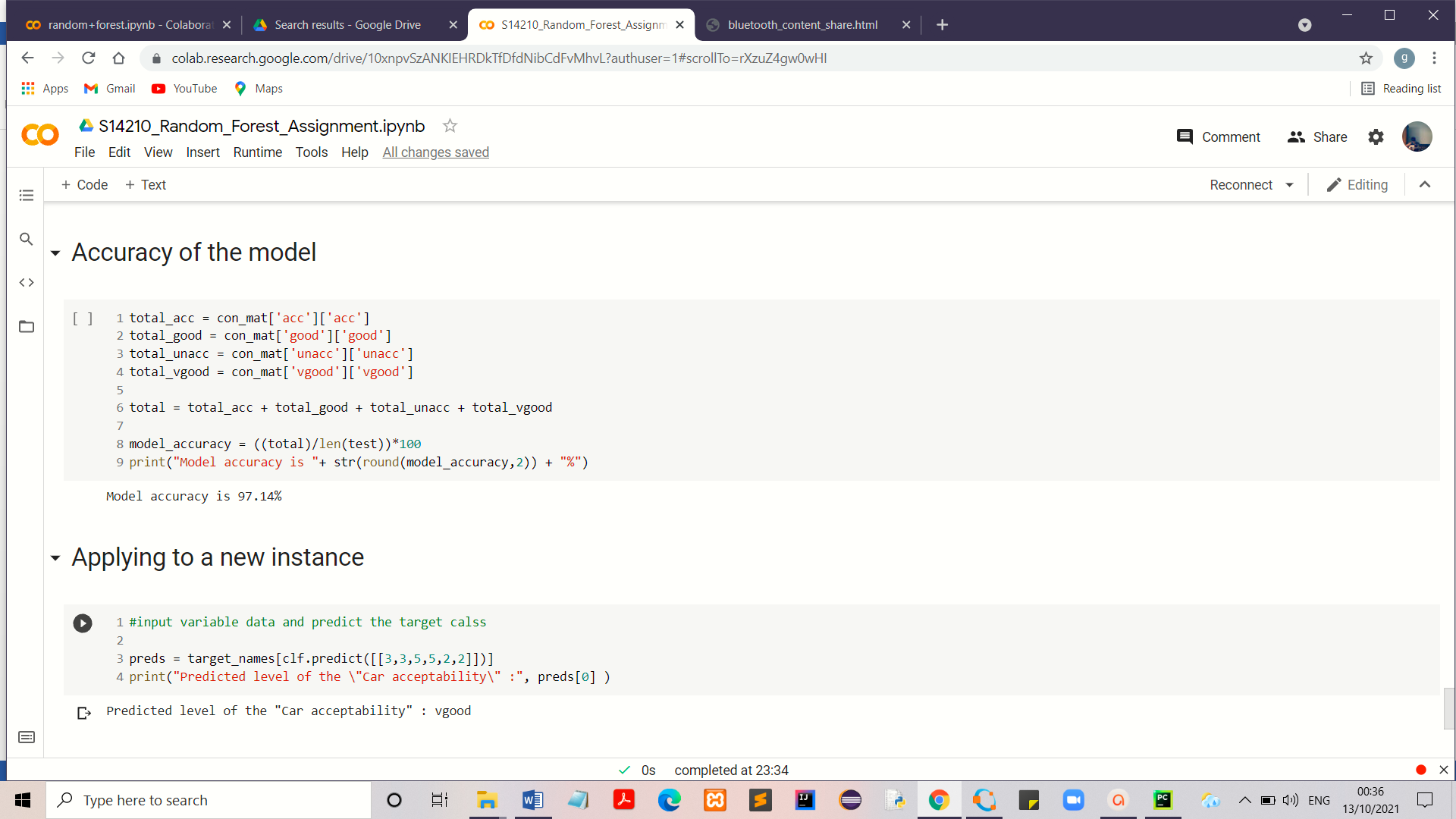
Here the trained algorithm is used to predict the classes of the instances of the testing dataset.

**Step 8 – Predict the probability of being “unacc”, “acc”, “vgood” and “good” of each instance.**

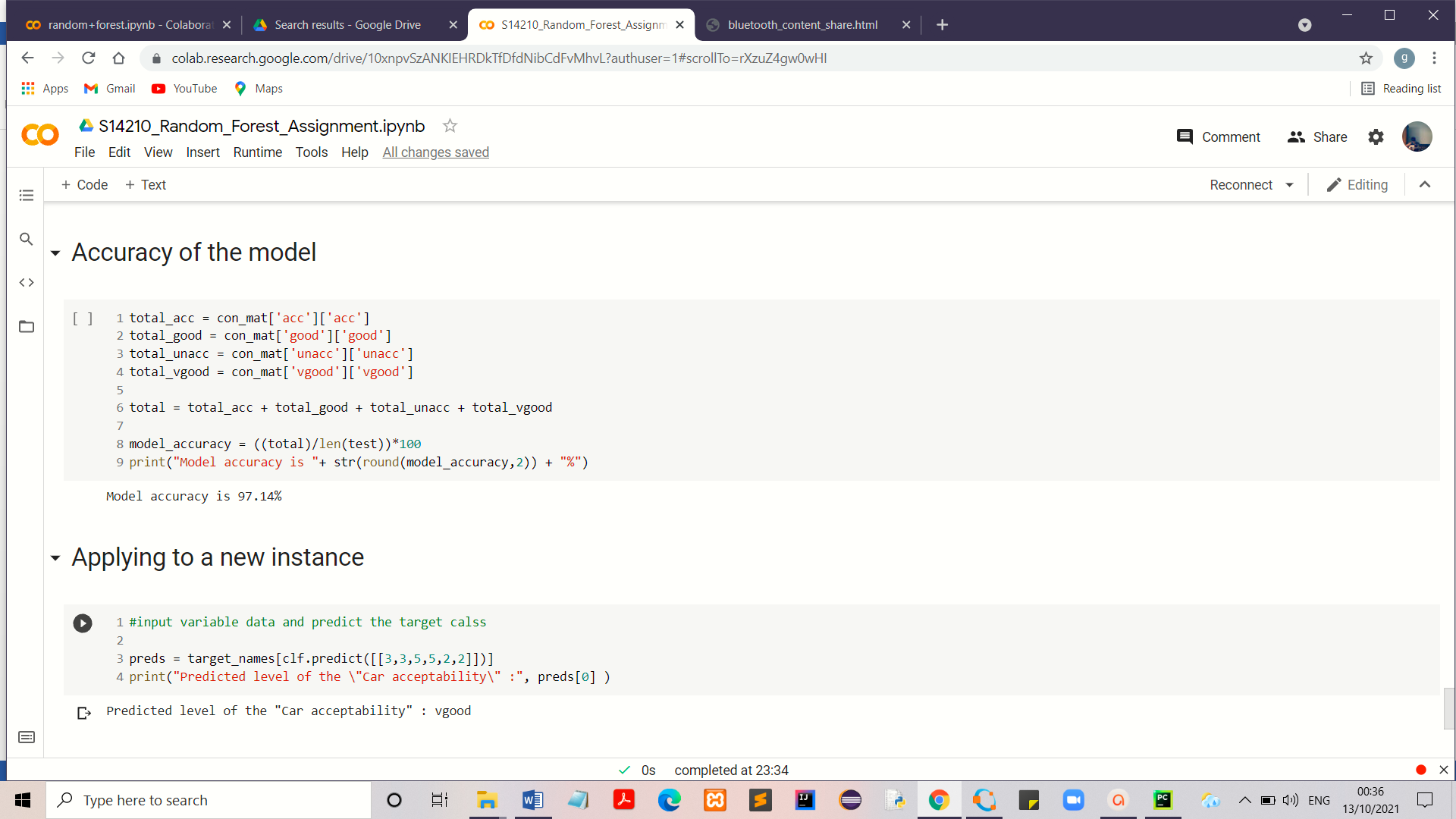


**Step 9 – Remapping the “Car acceptability”, the target column data into categorical values**

**Step 10 – Construction of a confusion matrix to make a comparison and predict the accuracy of the model**



**Step 11 – Applying a new instances to predict its target class**

Apply the numerical values of the features here to predict the target class of the new car.

**Discussion**

Use of the random forest approach to solve the car evaluation task gives highly accurate predictions for the data which were obtained from the testing dataset. We can take more efficient and more accurate trained model by providing more instances to the training dataset. It might be time consuming, but the output is more reliable. We also can manage the accuracy of the model by changing the parameters of the random forest classifier.

Since the scikit-learn library allows numerical inputs in order to do the classification task, chosen of the correct data conversion method is essential. If the dataset has ordinal categorical variables, integer encoding is more preferable otherwise one-hot encoding is preferable.

By adding new values of the cars into the entry field, customers can predict the acceptance level of the desired car. Therefore this model helps to make a decision for the customer without further help of other car expertise.

**Conclusion**

Random forest algorithm is a perfect and highly accurate model for classification tasks. It provides highly accurate outputs even for the large datasets.

**Colab Notebook**

<https://colab.research.google.com/drive/10xnpvSzANKIEHRDkTfDfdNibCdFvMhvL?usp=sharing>

**References**

* <https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/>
* <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
* <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>
* <https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm>