| Study Program: | **BSSE/BSCS** |
| --- | --- |
| **Semester:** | **Spring 2022** |
| **Course Code/Title:** | **Big Data Programming** |
| **Resource Person:** | **Mr. Mazhar Javed**  **Mazhar.awan@umt.edu.pk** |
| **Name,ID** | **Momina Jamil (F2017065219)** |
| **Due Date** | **18th July 2022** |

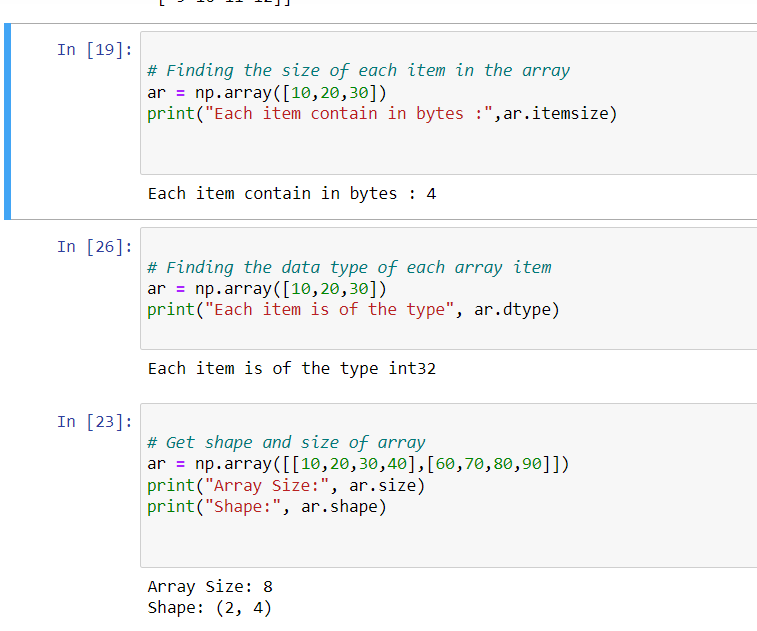
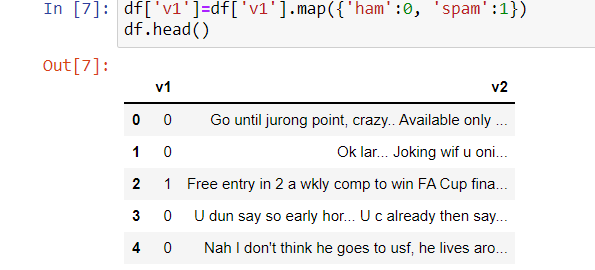
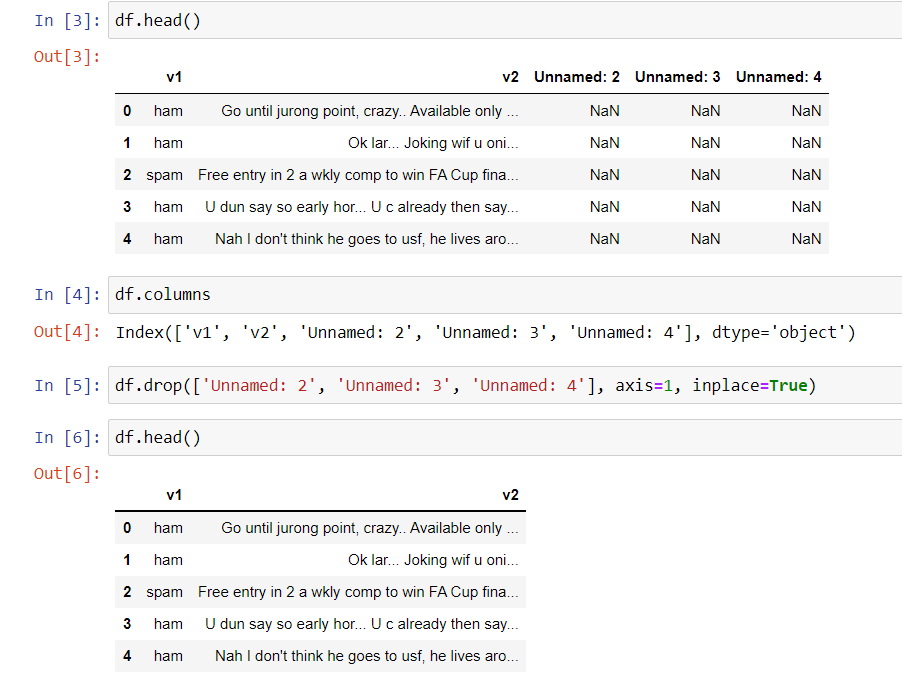
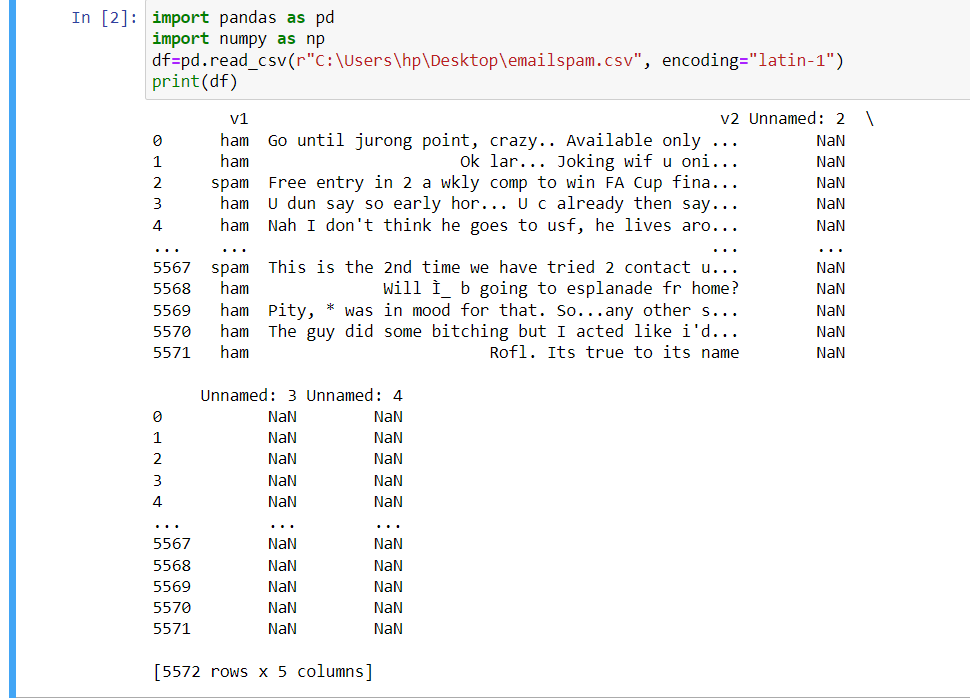
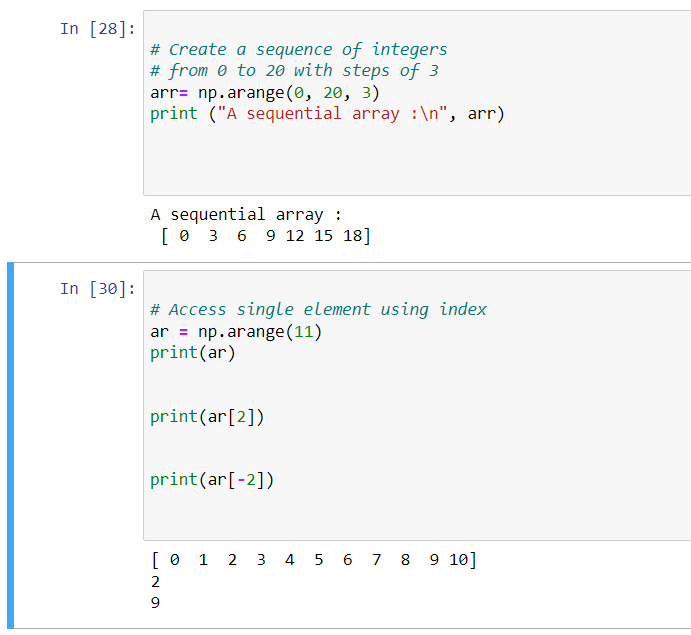
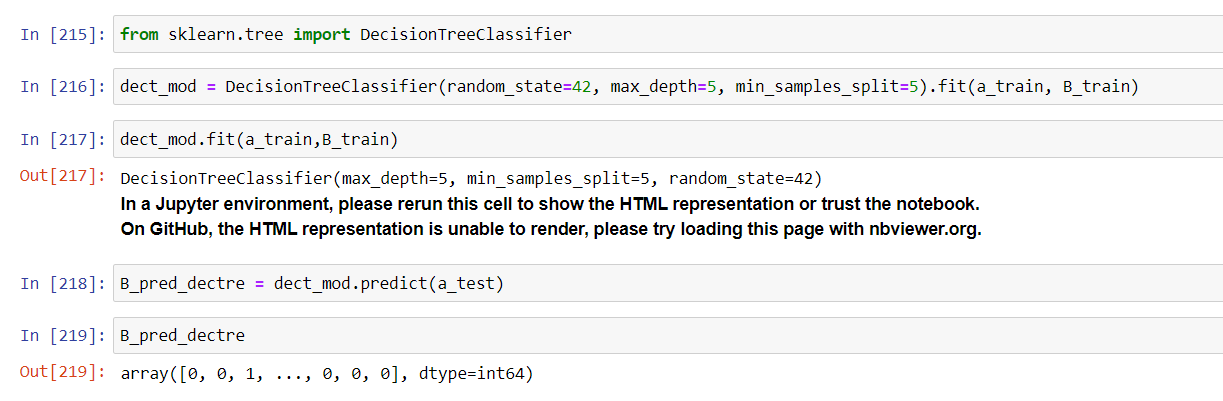
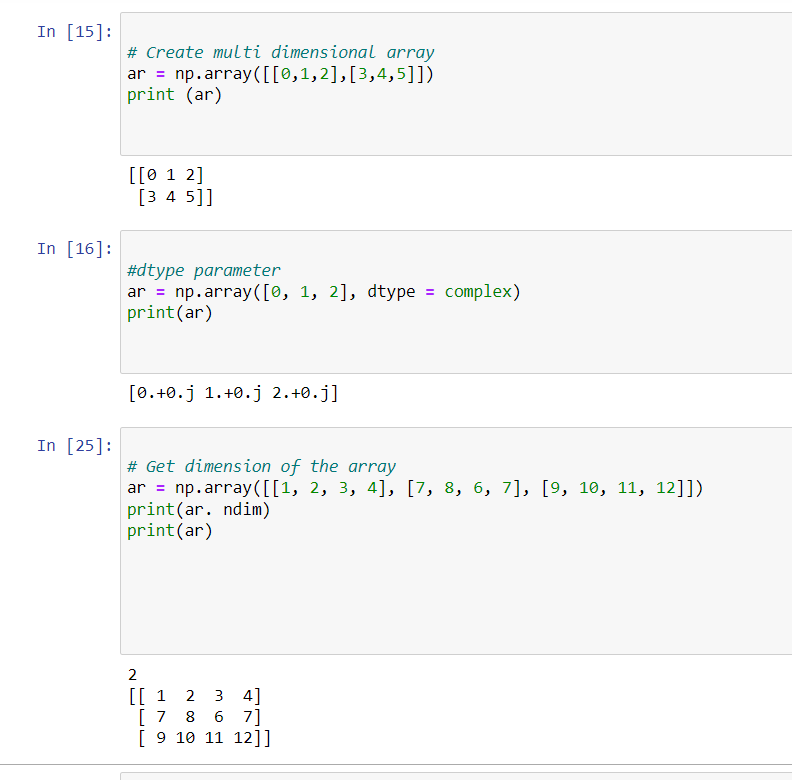
**Term Project ( Big Data)** 

# Project Description

Email Spam Filter,prediction is basically used to predict whether email is spam or not through naive bayes its predicting ham is those emails that are not spam and donated as 0 and spam is those mails that are spam and donated as 1 ,performed all the operations through numpy and pandas apply 4 models and run operation on dataframe using spark sql,context,and some common transformation and actions

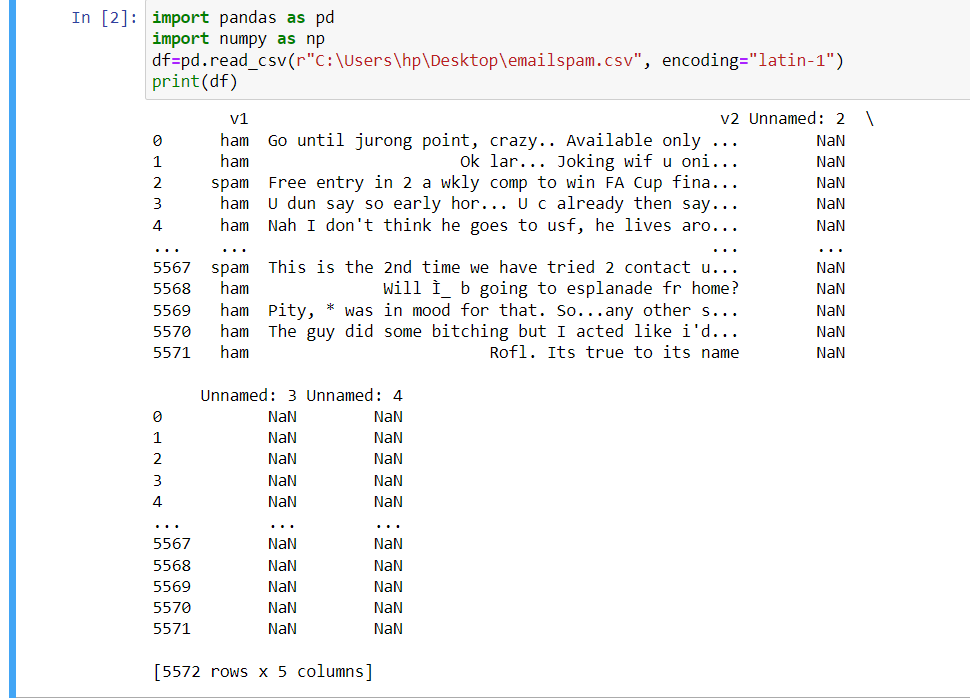
**Task 1: Operations on Resilient distributed datasets**

1. **Do all operations related to numpy and Pandas package on RDD**





1. **When to use DataFrames versus RDDs (type-safety, memory pressure, optimizations, i/o)**

Dataframe

Rdd

### **type safety**

* RDD – RDD provides a familiar object-oriented programming style with compile-time type safety.
* DataFrame – If you are trying to access the column which does not exist in the table in such case Dataframe APIs does not support compile-time error. It detects attribute error only at runtime.

### **Memory Pressure**

* RDD – Efficiency is decreased when serialization is performed individually on a java and scala object which takes lots of time.
* DataFrame – Use of off heap memory for serialization reduces the overhead. It generates byte code dynamically so that many operations can be performed on that serialized data. No need for deserialization for small operations.

**Optimization**

| RDD - No in-built optimization engine for RDDs. Developers need to write the optimized code themselves.  DataFrame -It uses a catalyst optimizer for optimization. |
| --- |

**Immutability and Interoperability**

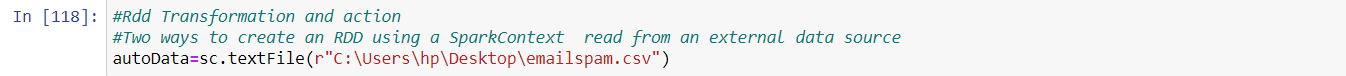
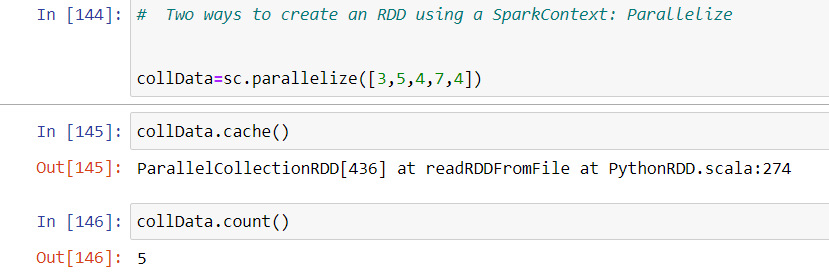
Rdd-Immutable partitions that easily transform into DataFrames.

Dataframe-Transforming into a DataFrame loses the original RDD.

1. **Two ways to create an RDD using a SparkContext: Parallelize and read from an external data source**

Explanation:

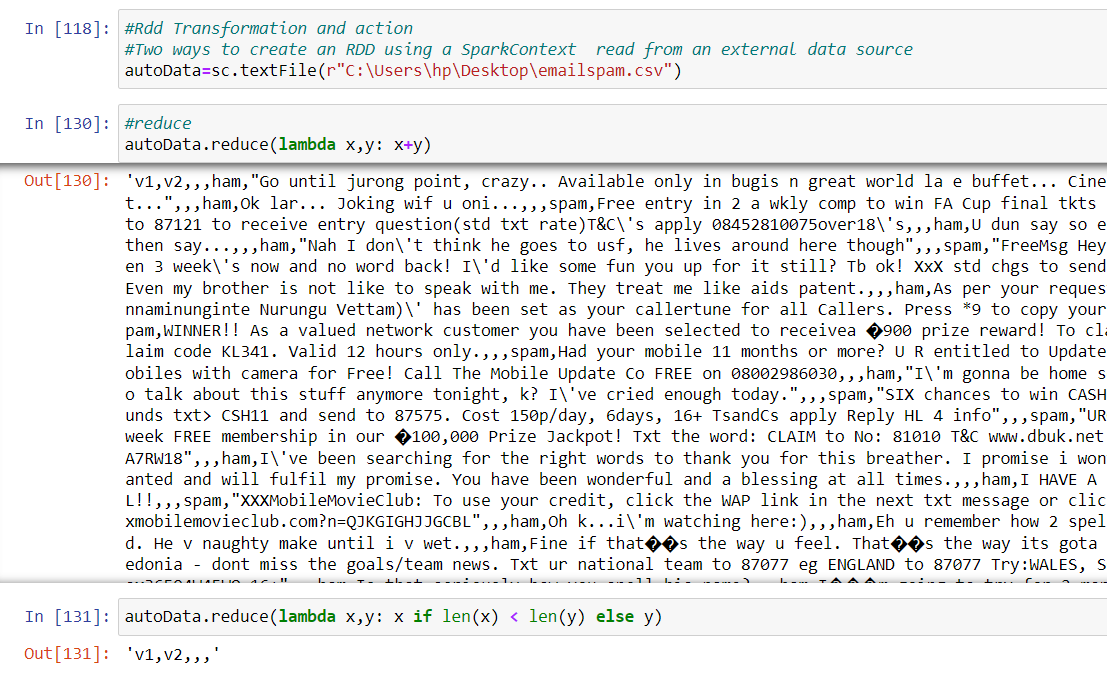
In Parallelize we are loading the data and through external source we are reading the file data

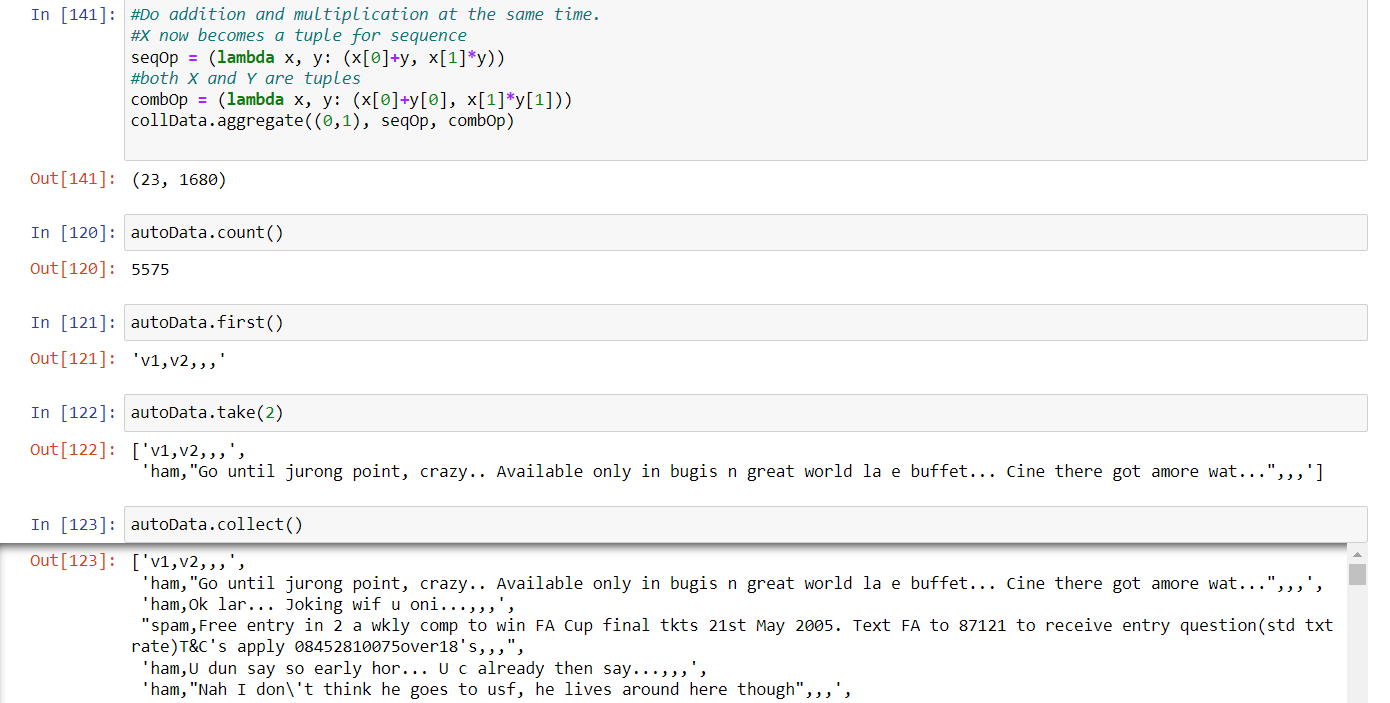
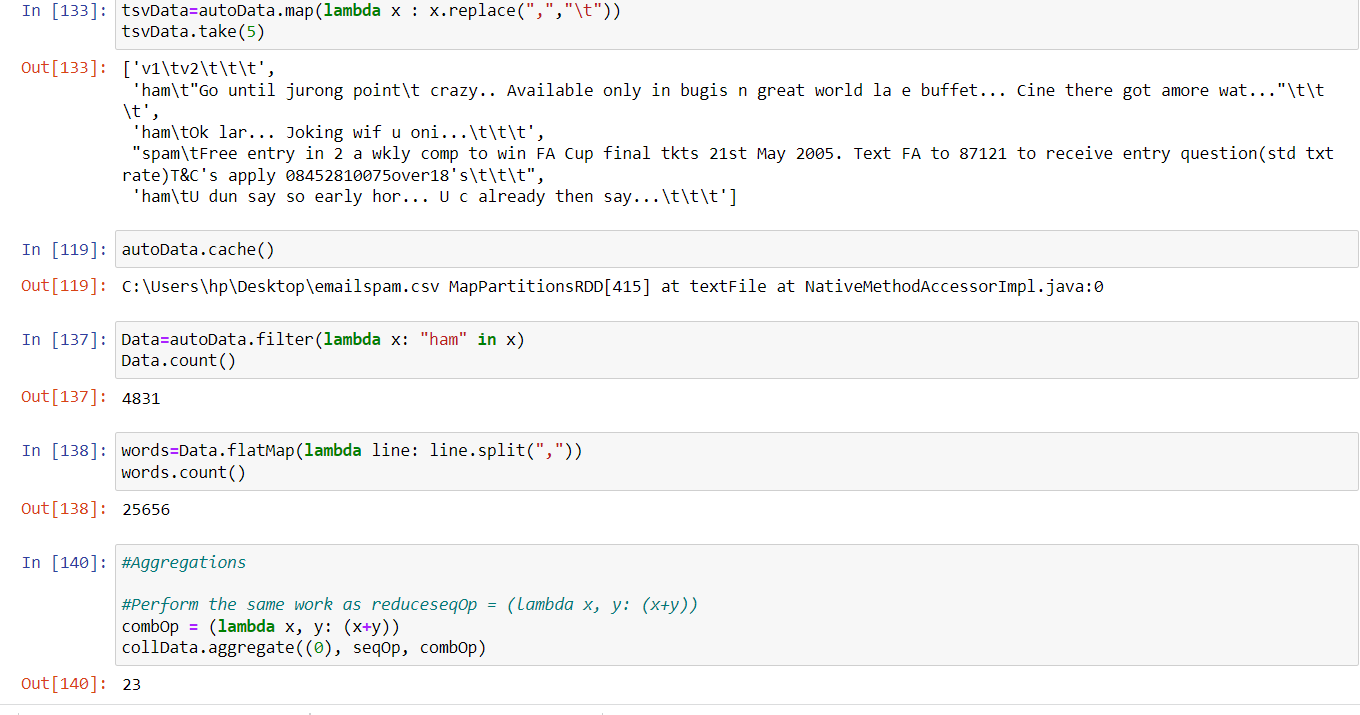


1. **Do some Common transformations and actions as**

In this code i have used transformation functions

flatmap,filter,reduce,map and actions such as count(),collect(),cache(),take(),etc



****

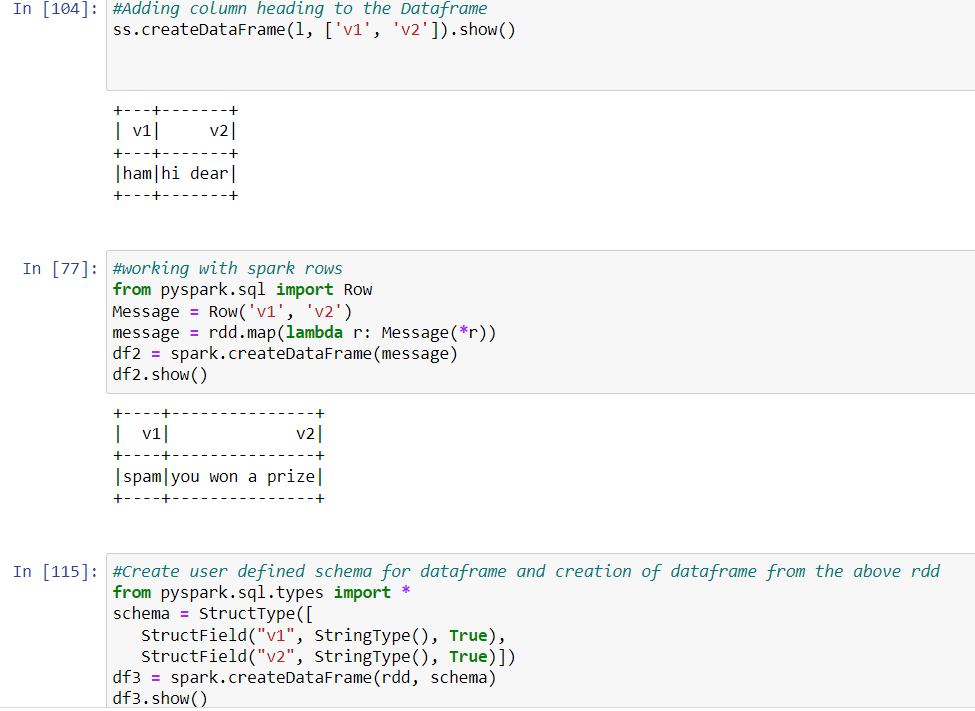
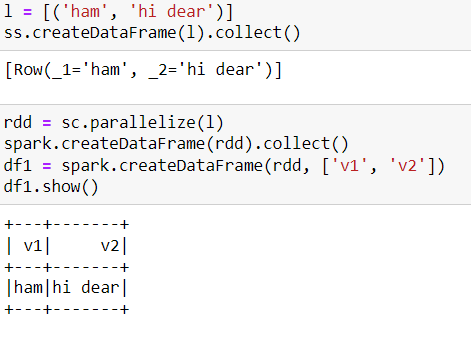
**Task 2: Operations on DataFrames and Spark SQL**

Use a SQLContext to create a DataFrame from different data sources (S3, JSON, RDBMS, HDFS, Cassandra, etc.)

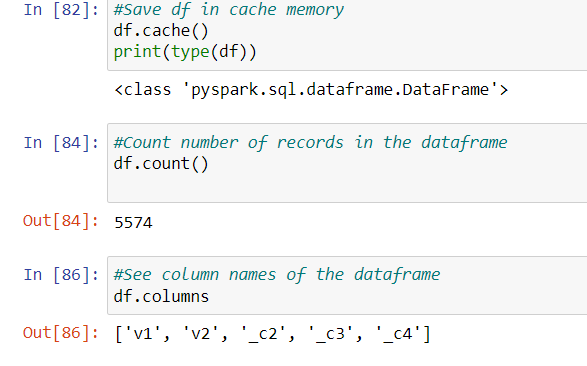
1. **Run some common operations on DataFrames to explore it**

Here in the code i have use some common operations list,adding column heading,converting rdd to spark frame

creating a object of spark context,spark session and importing them



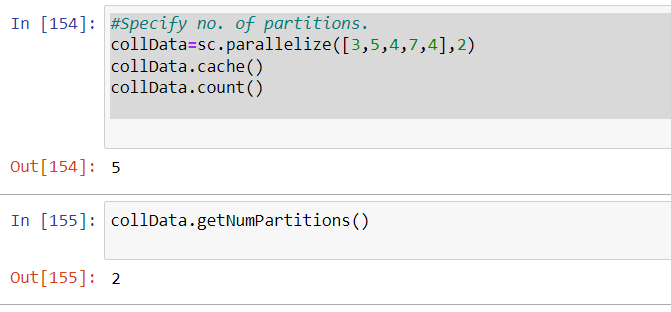
1. **Cache a DataFrame into memory**



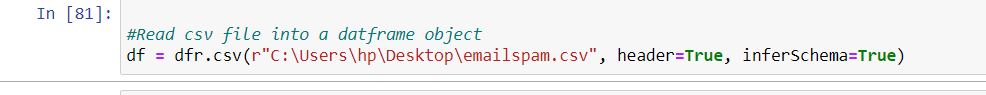
After saving data frame in to cache memory ,counting number of records and column of dataframe

1. **Correctly size the number of partitions in a DF, including the size of each partition**

In this code we have given no of partition in dataframe and each size of partition

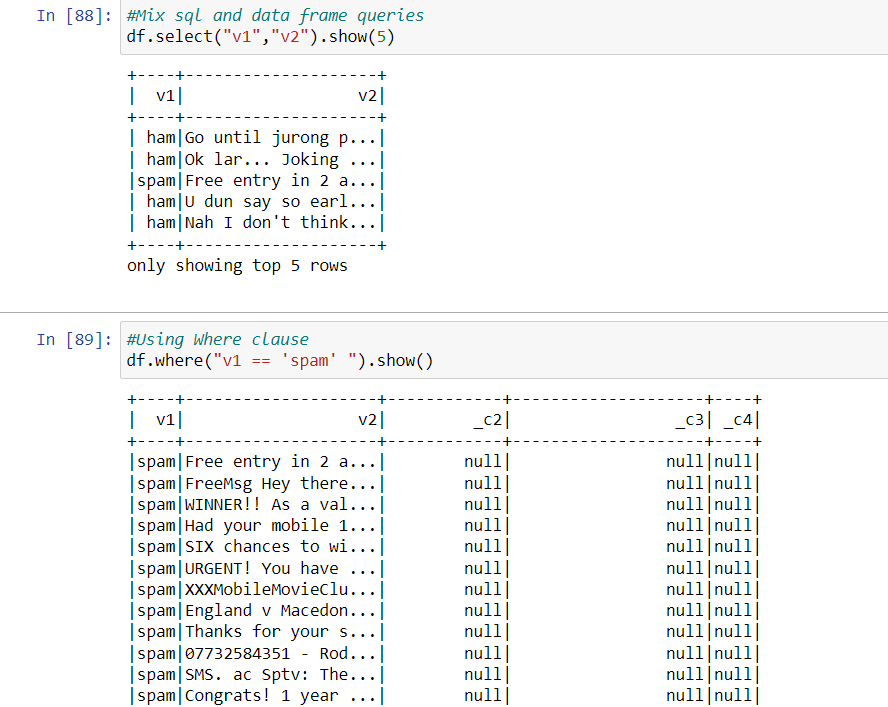


1. **Use the Spark CSV library from Spark Packages to read structured files**

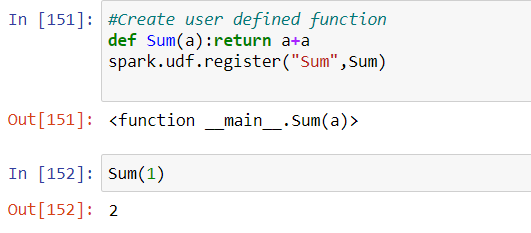


1. Mix SQL and DataFrame queries

Here in this code applying select ,where clause sql query in dataframe

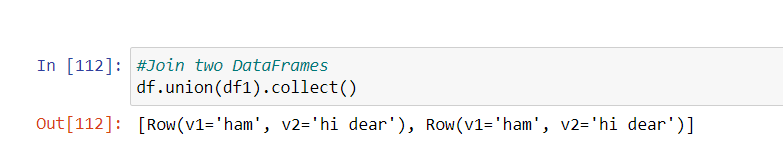


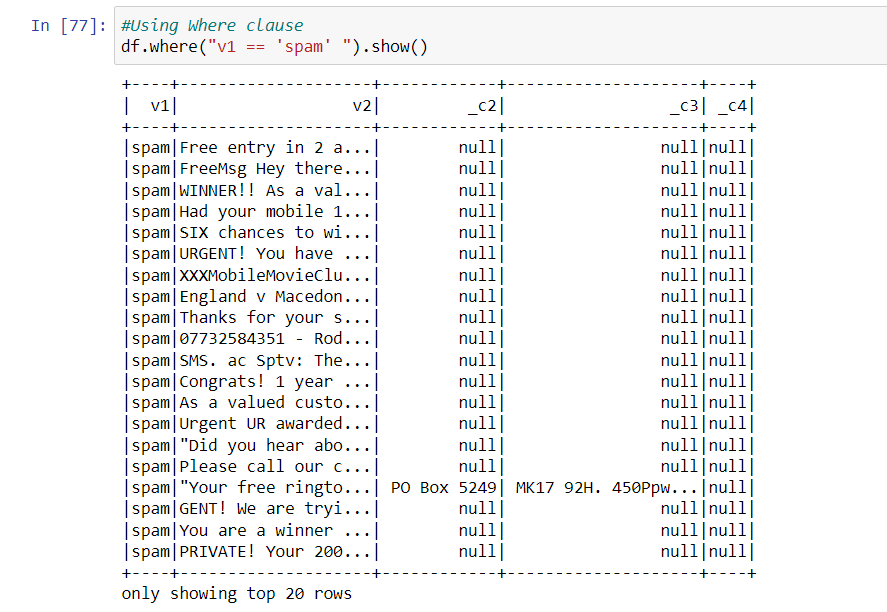
1. **Write a user-defined function (UDF)**



1. **Join two DataFrames**

Here in code Two dataframe are join through union function



1. **Overview of how Spark SQL’s Catalyst optimizer converts logical plans to optimized physical plans**

The main data type in Catalyst is a tree composed of node objects. Each node has a node type and zero or more children. New node types are defined in Scala as subclasses of the TreeNode class. These objects are immutable and can be manipulated using functional transformations, as discussed in the next subsection.

As a simple example, suppose we have the following three node classes for a very simple expression language:

* Literal(value: Int): a constant value
* Attribute(name: String): an attribute from an input row, e.g.,“x”
* Add(left: TreeNode, right: TreeNode): sum of two expressions.

These classes can be used to build up trees; for example, the tree for the expression x+(1+2), would be represented in Scala code as follows:

Add(Attribute(x), Add(Literal(1), Literal(2)))

1. Create visualizations using matplotlib, Databricks, and Google Visualizations



**Task 3: Shared variables (accumulators and broadcast variables)**

* Common use cases for shared variables

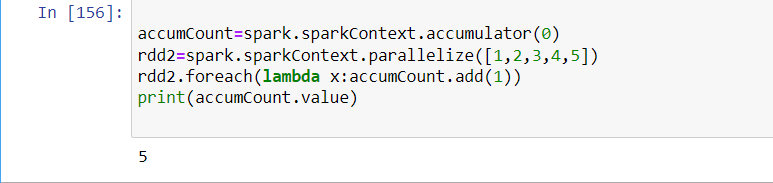
explanation:in it where dictionary keyword is use in data it will automatically give full name meaning in output for example in data its pr and in dictionary meaning is paris so in output its paris



* **How accumulators can be used to implement distributed counters in parallel**

Explanation:

These variables are used to add sum or counts and final results can be accessed only by driver program.



* Using broadcast variables to keep a read-only variable cached on each machine



* Differences between broadcast variables and closures/lambdas (across stages versus per stage)

explanation:in it where dictionary keyword is use in data it will automatically give full name meaning in output for example in data its pr and in dictionary meaning is paris so in output its paris

rdd map is applying to each element and in result there is new rdd



**Task 4: Spark machine learning**  
http://spark.apache.org/docs/latest/ml-guide.html

1. **Common use cases of machine learning with Spark**

Streaming Data,Machine Learning,. Interactive Analysis, Fog Computing,Spark in the Real World

**2) When to use Spark MLlib (w/ RDDs) versus Spark ML (w/ DataFrames)**

spark.mllib carries the original API built on top of RDDs.

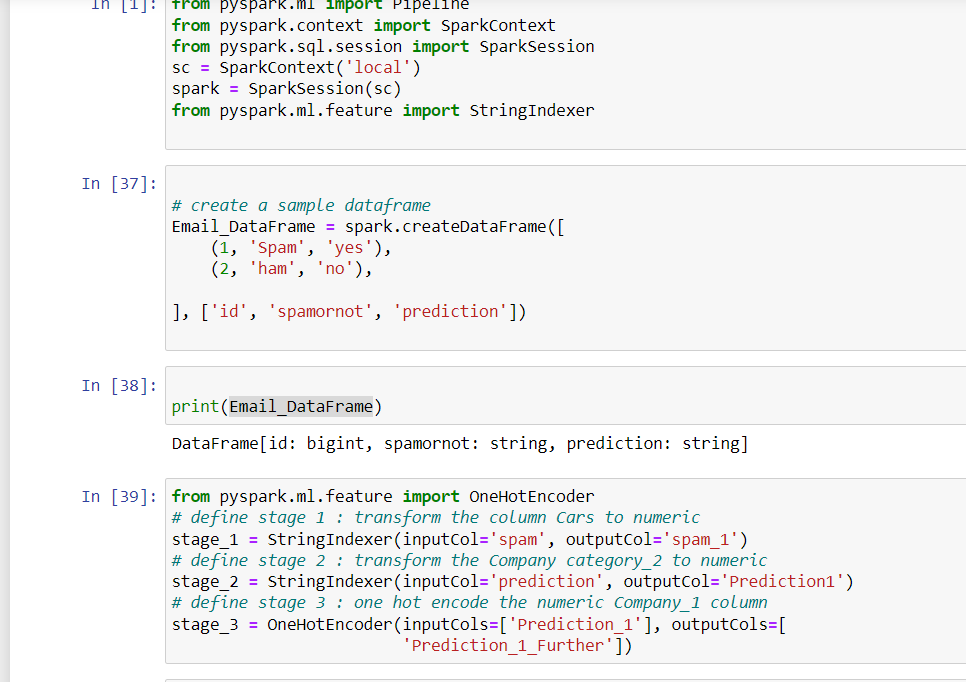
spark.ml contains higher-level API built on top of DataFrames for constructing ML pipelines.



3) ML Pipelines concepts:

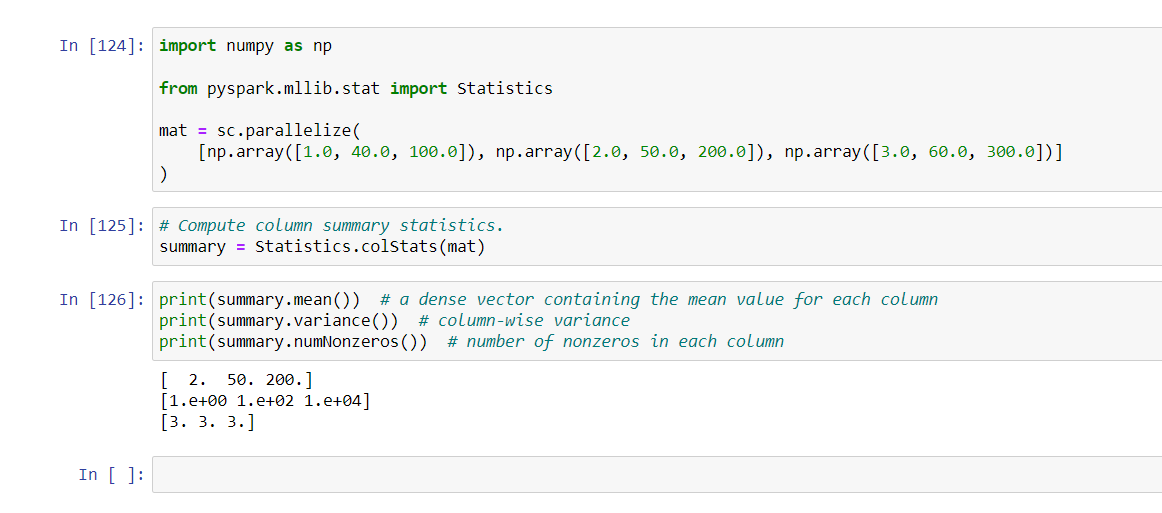
DataFrames, transformer, estimator, pipeline, parameter

* [**DataFrame**](https://spark.apache.org/docs/1.6.0/ml-guide.html#dataframe): Spark ML uses DataFrame from Spark SQL as an ML dataset, which can hold a variety of data types. E.g., a DataFrame could have different columns storing text, feature vectors, true labels, and predictions.
* [**Transformer**](https://spark.apache.org/docs/1.6.0/ml-guide.html#transformers): A Transformer is an algorithm which can transform one DataFrame into another DataFrame. E.gAn ML model is a Transformer which transforms a DataFrame with features into a DataFrame with predictions.
* [**Estimator**](https://spark.apache.org/docs/1.6.0/ml-guide.html#estimators): An Estimator is an algorithm which can be fit on a DataFrame to produce a Transformer. E.g., a learning algorithm is an Estimator which trains on a DataFrame and produces a model.
* [**Pipeline**](https://spark.apache.org/docs/1.6.0/ml-guide.html#pipeline): A Pipeline chains multiple Transformers and Estimators together to specify an ML workflow.
* [**Parameter**](https://spark.apache.org/docs/1.6.0/ml-guide.html#parameters): All Transformers and Estimators now share a common API for specifying parameters.



**4) Basic statistics with MLlib**

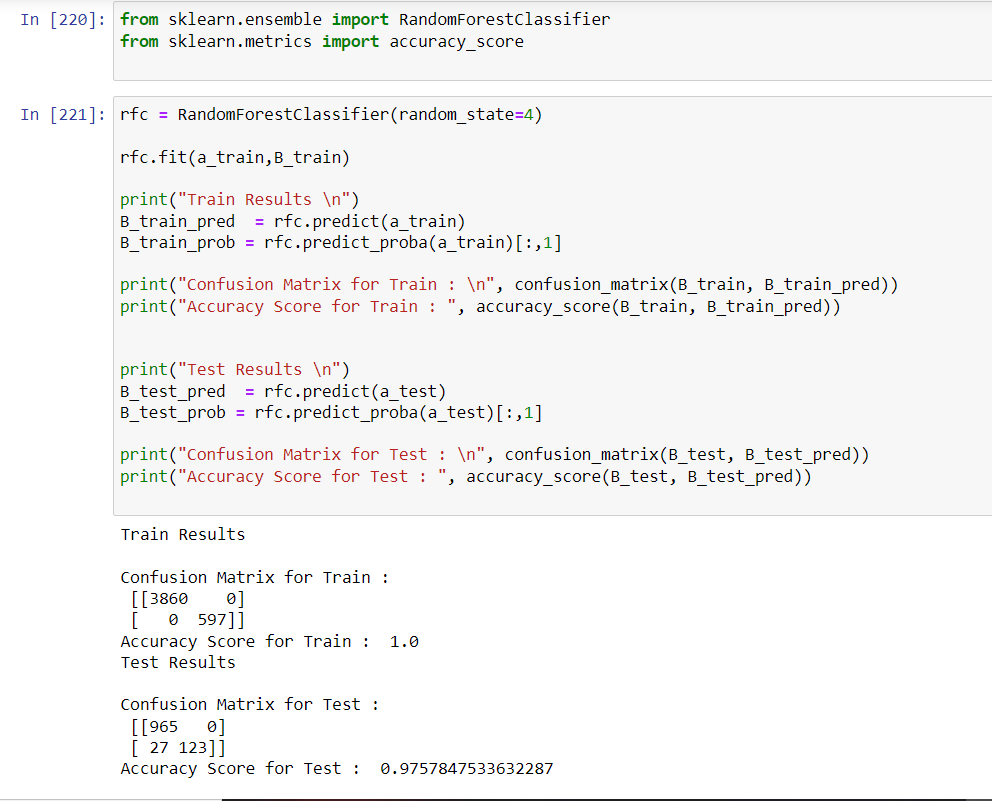
Coll stats()returns an instance of which contains the column-wise max, min, mean, variance, and number of nonzeros, as well as the total count.

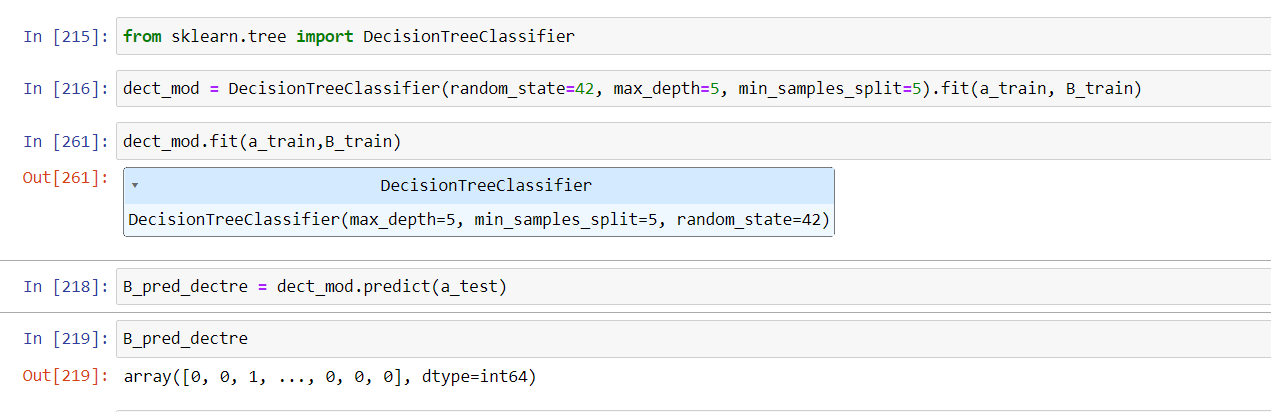


**5)Apply 4 to 5 machine learning (k-means, linear regression, Naive Bayes, Decision Tree, Random Forest, logistic regression or any algorithm like ensemble or boosting )**

In RandomForest As the **Accuracy Score** and is 1.0.

It is clear that, our model is overfitting means, it is learning too much from the data.

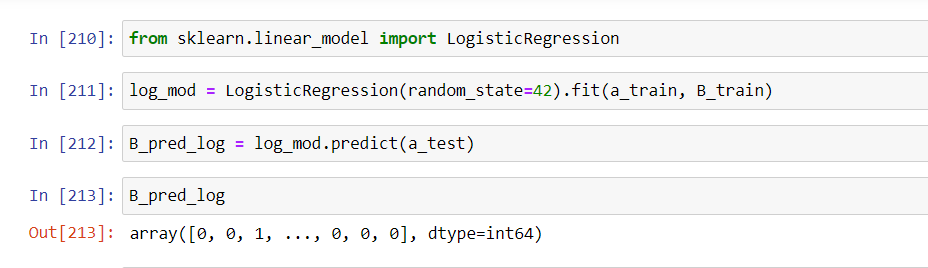




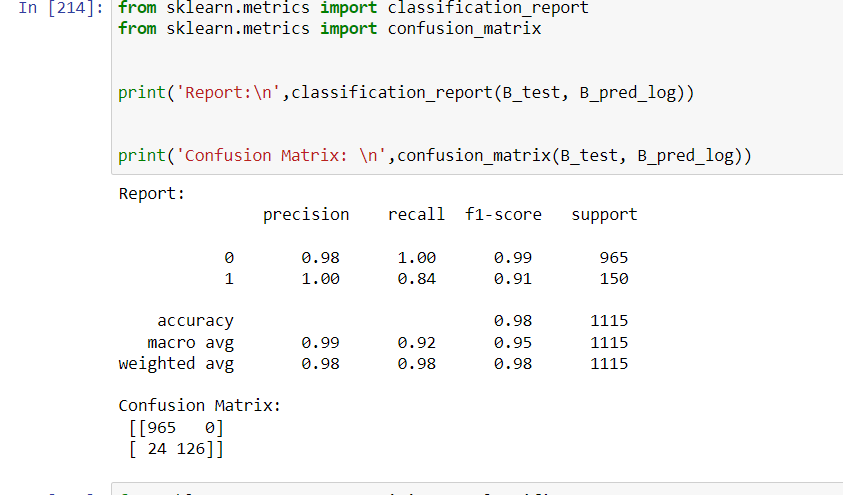


Email Spam is donated as 1 and not spam is donated as 0

and accuracy is 99.4



**6)Evaluation: confusion matrix, classification report**



**7 flask**

i have run this code on vs by installing flask



