# MACHINE LEARNING

**(Finding Donors)**

Summer Internship Report Submitted in partial fulfillment of the requirement for undergraduate degree of

## Bachelor of Technology

In

## Computer science Engineering

By

## Vuyyuru Karthik

## 221710313063

Under the Guidance of

Assistant Professor



Department Of Computer Science And Engineering

GITAM School of Technology

GITAM (Deemed to be University)

Hyderabad-502329

July 2020

i

# DECLARATION

I submit this industrial training work entitled **“FINDING DONAR**” to GITAM (Deemed To Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer Science And Engineering**”. I declare that it was carried out independently by me under the guidance of Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD Vuyyuru Karthik

Date:12-07-2020 221710313063

ii

GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India Dated:12-07-2020

**CERTIFICATE**

This is to certify that the Industrial Training Report entitled **“Finding donors”** is being submitted by Vuyyuru Karthik (221710313063) in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science And Engineering** at GITAM (Deemed To Be University), Hyderabad during the academic year 2019-20

It is faithful record work carried out by him at the **Computer Science And Engineering Department**, GITAM University Hyderabad Campus under my guidance and supervision.

### 

Assistant Professor Professor and HOD

Department of ECE Department of CSE

iv

**ACKNOWLEDGEMENT**

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank respected **Dr. N. Siva Prasad,** Pro Vice Chancellor, GITAM Hyderabad and **Dr. CH. Sanjay,** Principal, GITAM Hyderabad

I would like to thank respected **Dr. K. Manjunathachari ,** Head of the Department of Electronics and Communication Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a internship report. It helped me a lot to realize of what we study for.

I would like to thank the respected faculties **Mr. M. Venkateswarlu** who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Vuyyuru Karthik

221710313063

v

## ABSTRACT

Machine learning algorithms are used to predict the values from the data set by splitting the data set in to train and test and building Machine learning algorithms models of higher accuracy to predict the values is the primary task to be performed on Cereals data set My perception of understanding the given data set has been in the view of undertaking a client’s requirement of overcoming the stagnant point of sales of the products being manufactured by client.

To get a better understanding and work on a strategical approach for solution of the client, I have adapted the view point of looking at ratings of the products and for further deep understanding of the problem, I have taken the stance of a consumer and reasoned out the various factors of choice of the products and they purchase , and my primary objective of this case study was to look up the factors which were dampening the sale of products and corelate them to ratings of products and draft out an outcome report to client regarding the various accepts of a product manufacturing , marketing and sale point determination

vi

# Table of Contents:

**CHAPTER 1:MACHINE LEARNING1**

* 1. INTRODUCTION…………………………………………………………1
  2. IMPORTANCE OF MACHINE LEARNING…………………………… 1
  3. USES OF MACHINE LEARNING… 2
  4. TYPES OF LEARNING ALGORITHMS… 3
     1. Supervised Learning… 3
     2. Unsupervised Learning… 3
     3. Semi Supervised Learning……………………………………… 4
  5. RELATION BETWEEN DATA MINING,MACHINE LEARNING AND DEEP LEARNING… 5

### **CHAPTER 2:PYTHON**

* 1. INTRODUCTOIN TO PYTHON… 6
  2. HISTORY OF PYTHON… 6
  3. FEATURES OF PYTHON… 6
  4. HOW TO SETUP PYTHON… 7
     1. Installation(using python IDLE)… 7
     2. Installation(using Anaconda)… 8
  5. PYTHON VARIABLE TYPES… 9
     1. Python Numbers… .10
     2. Python Strings… .10
     3. Python Lists… .11
     4. Python Tuples… .11
     5. Python Dictionary… ..12
  6. PYTHON FUNCTION… ……………12
     1. Defining a Function… 12

# CHAPTER 1:

# MACHINE LEARNING

## INTRODUCTION:

Machine Learning(ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence(AI).

## IMPORTANCE OF MACHINE LEARNING:

Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today’s data-rich world.

Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries.

With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to increased sophistication of machine learning, which helps analyze those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques

The process flow depicted here represents how machine learning works

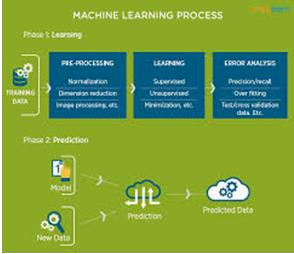


Figure 1 : The Process Flow

## USES OF MACHINE LEARNING:

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let’s consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data

Traditionally, data analysis was always being characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data.

By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

## TYPES OF LEARNING ALGORITHMS:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

## Supervised Learning :

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

Supervised machine learning algorithms uncover insights, patterns, and relationships from a labelled training dataset – that is, a dataset that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to “learn” how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.

Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

Classification, in which the algorithm attempts to label each example by choosing between two or more different classes. Choosing between two classes is called binary classification, such as determining whether or not someone will default on a loan. Choosing between more than two classes is referred to as multiclass classification.

## Unsupervised Learning:

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

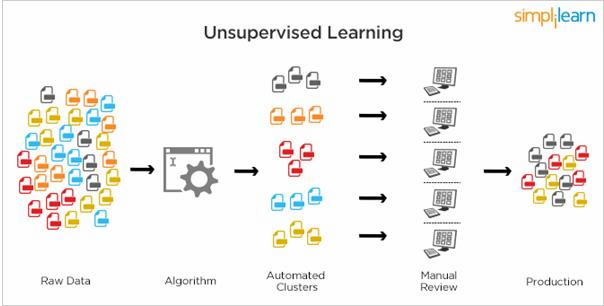


Figure 2 : Unsupervised Learning

Popular techniques where unsupervised learning is used also include self-organizing maps, nearest neighbor mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

## Semi Supervised Learning:

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labeled and unlabeled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabeled data.

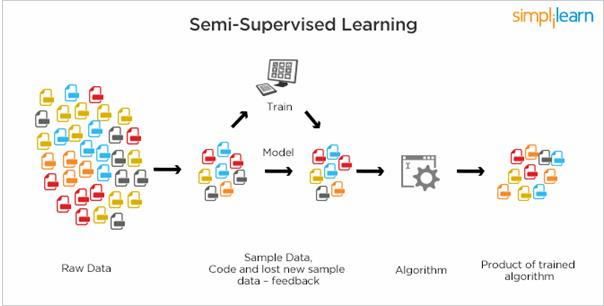


Figure 3 : Semi Supervised Learning

## RELATION BETWEEN DATA MINING,MACHINE LEARNING AND DEEP LEARNING:

Machine learning and data mining use the same algorithms and techniques as data mining, except the kinds of predictions vary. While data mining discovers previously unknown patterns and knowledge, machine learning reproduces known patterns and knowledge—and further automatically applies that information to data, decision-making, and actions.

Deep learning, on the other hand, uses advanced computing power and special

types of neural networks and applies them to large amounts of data to learn, understand, and identify complicated patterns. Automatic language translation and medical diagnoses are examples of deep learning.

# CHAPTER 2:

# PYTHON

Basic programming language used for machine learning is : PYTHON

## INTRODUCTION TO PYHTON:

* + - Python is a high-level, interpreted, interactive and object-oriented scripting language.
    - Python is a general purpose programming language that is often applied in scripting roles
    - Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
    - Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
    - Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

## HISTORY OF PYTHON:

* + - Python was developed by GUIDO VAN ROSSUM in early 1990’s
    - Its latest version is 3.7 , it is generally called as python3

## FEATURES OF PYTHON:

* Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax,This allows the student to pick up the language quickly.
* Easy-to-read: Python code is more clearly defined and visible to the eyes.
* Easy-to-maintain: Python's source code is fairly easy-to-maintaining.
* A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* Databases: Python provides interfaces to all major commercial databases.
* GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

## HOW TO SETUP PYTHON:

* + - Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.
    - The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

**2.4.1 Installation(using Python IDLE) :**

* + - * Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.
      * [Download python from www.python.org](http://www.python.org/)
      * When the download is completed, double click the file and follow the instructions to install it.
      * When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.

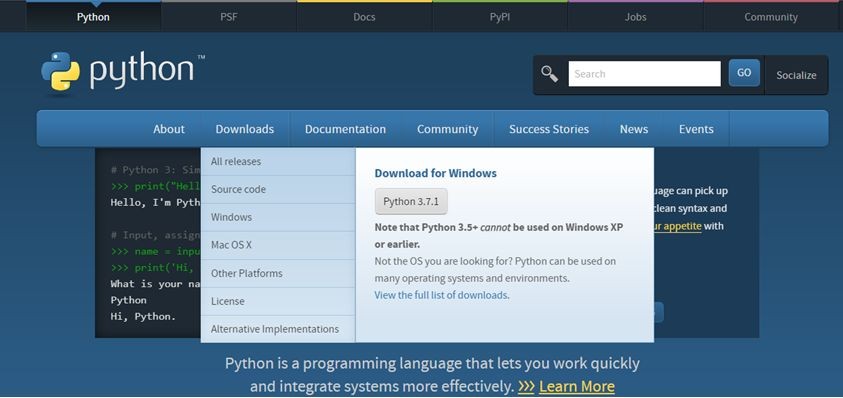


Figure 4 : Python download

## Installation(using Anaconda):

* + - * Python programs are also executed using Anaconda.
      * Anaconda is a free open source distribution of python for large scale data processing, predictive analytics and scientific computing.
      * Conda is a package manager quickly installs and manages packages.
      * In WINDOWS:
      * In windows
        + Step 1: Open Anaconda.com/downloads in web browser.
        + Step 2: Download python 3.4 version for (32-bitgraphic installer/64 -bit graphic installer)
        + Step 3: select installation type( all users)
        + Step 4: Select path(i.e. add anaconda to path & register anaconda as default python 3.4) next click install and next click finish
        + Step 5: Open jupyter notebook ( it opens in default browser)

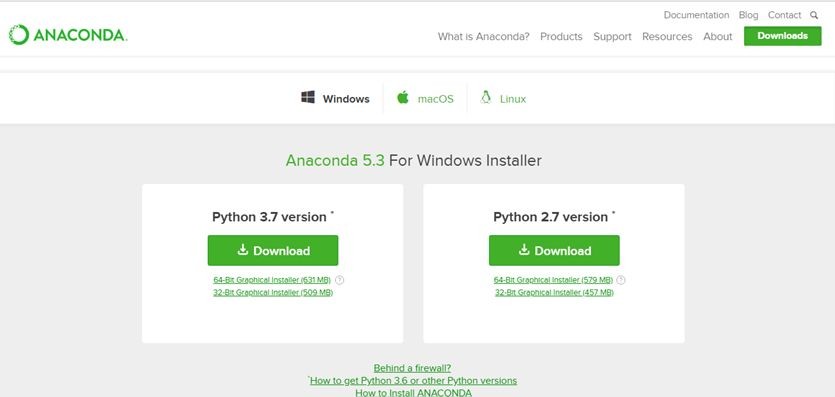


Figure 5 : Anaconda download

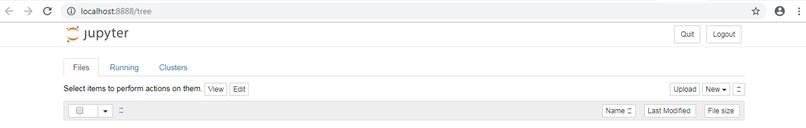


Figure 6 : Jupyter notebook

## PYTHON VARIABLE TYPES:

* Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.
* Variables are nothing but reserved memory locations to store values.
* Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory.
* Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.
* Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.
* Python has five standard data types –
  + Numbers
  + Strings
  + Lists
  + Tuples
  + Dictionary

## Python Numbers:

* + - * Number data types store numeric values. Number objects are created when you assign a value to them.
      * Python supports four different numerical types − int (signed integers) long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

## Python Strings:

* + - * Strings in Python are identified as a contiguous set of characters represented in the quotation marks.
      * Python allows for either pairs of single or double quotes.
      * Subsets of strings can be taken using the slice operator ([ ] and [:] ) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.
      * The plus (+) sign is the string concatenation operator and the asterisk (\*) is the repetition operator.

## Python Lists:

* + - * Lists are the most versatile of Python's compound data types.
      * A list contains items separated by commas and enclosed within square brackets

([]).

* + - * To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.
      * The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1.
      * The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

## Python Tuples:

* + - * A tuple is another sequence data type that is similar to the list.
      * A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.
      * The main differences between lists and tuples are: Lists are enclosed in brackets ( [

] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated.

* + - * Tuples can be thought of as read-only lists.
      * For example − Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple. Tuples have no append or extend method. You can't remove elements from a tuple. Tuples have no remove or pop method.

## Python Dictionary:

* + - * Python's dictionaries are kind of hash table type. They work like associative arrays

or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

* + - * Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).
      * You can use numbers to "index" into a list, meaning you can use numbers to find out what's in lists. You should know this about lists by now, but make sure you understand that you can only use numbers to get items out of a list.
      * What a dict does is let you use anything, not just numbers. Yes, a dict associates one thing to another, no matter what it is.

## PYTHON FUNCTION:

* + 1. **Defining a Function:**

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (i.e.()).

Any input parameters or arguments should be placed within these parentheses.

You can also define parameters inside these parentheses

The code block within every function starts with a colon (:) and is indented. The statement returns [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

## Calling a Function:

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

## PYTHON USING OOP’s CONCEPTS:

* + 1. **Class:**
       - Class: A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.
       - Class variable: A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.
       - Data member: A class variable or instance variable that holds data associated with a class and its objects.
       - Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.
       - Defining a Class:
         * We define a class in a very similar way how we define a function.
         * Just like a function ,we use parentheses and a colon after the class name(i.e. ():) when we define a class. Similarly, the body of our class is

indented like a functions body is.

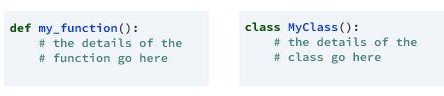


Figure 7 : Defining a Class

## init method in Class:

* + - * The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.
      * The init method has a special name that starts and ends with two underscores: init ().

# CHAPTER 3:

# CASE STUDY

**PROBLEM STATEMENT:** To find the donar for charity using Machine Learning algorithm .

**DATA:**

**Features**

* age: Age
* workclass: Working Class (Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked)
* education\_level: Level of Education (Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool)
* education-num: Number of educational years completed
* marital-status: Marital status (Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse)
* occupation: Work Occupation (Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces)
* relationship: Relationship Status (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
* race: Race (White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black)
* sex: Sex (Female, Male)
* capital-gain: Monetary Capital Gains
* capital-loss: Monetary Capital Losses
* hours-per-week: Average Hours Per Week Worked
* native-country: Native Country (United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands)

**Target Variable**

* income: Income Class (<=50K, >50K)

**CHAPTER 4:**

**MODEL BUILDING**

## PREPROCESSING OF THE DATA:

Preprocessing of the data actually involves the following steps:

## GETTING THE DATASET:

We can get the data set from the database or we can get the data from client.

## IMPORTING THE LIBRARIES:

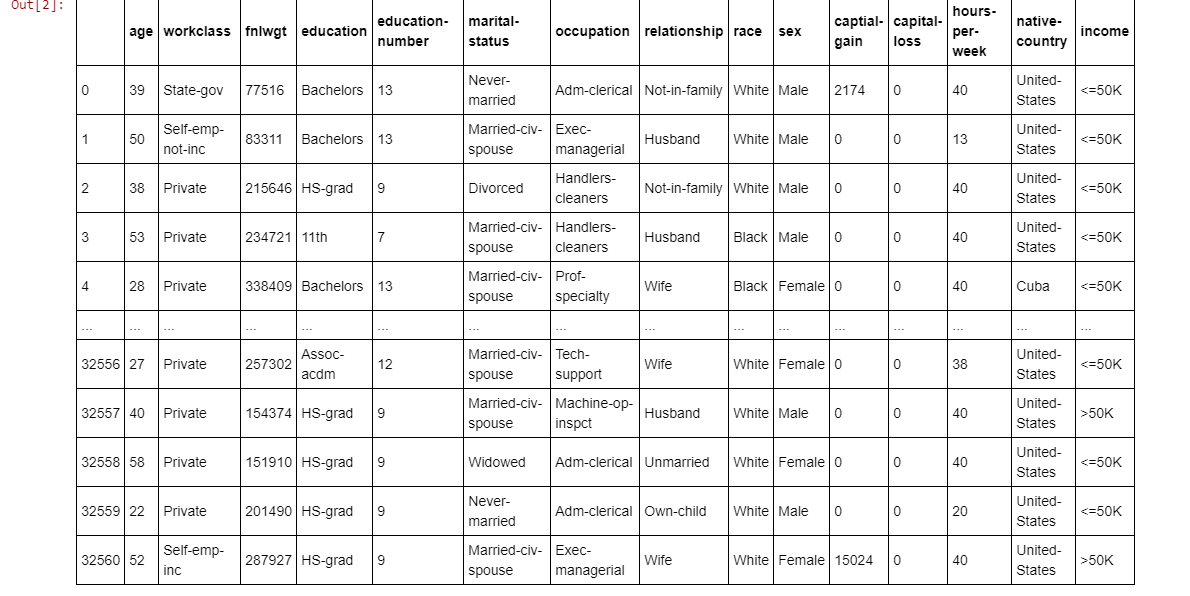
We have to import the libraries as per the requirement of the algorithm.

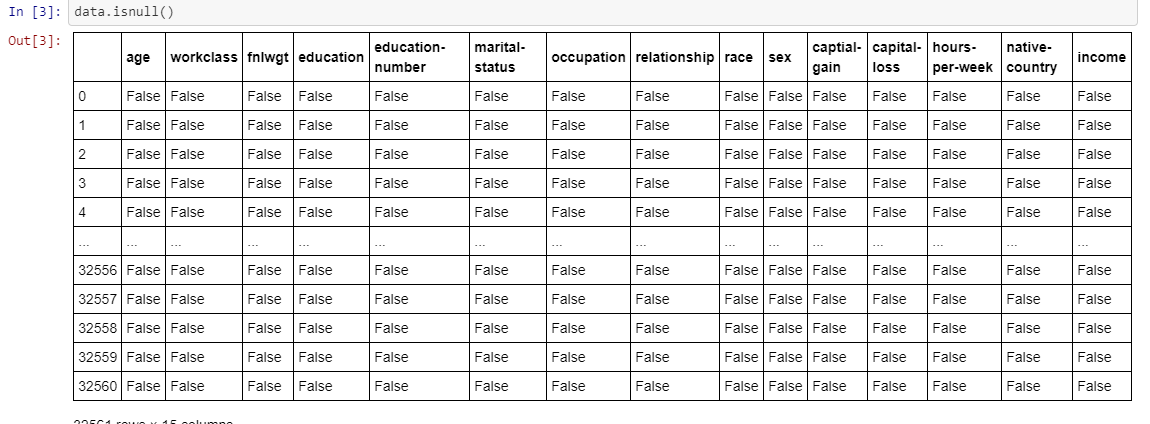


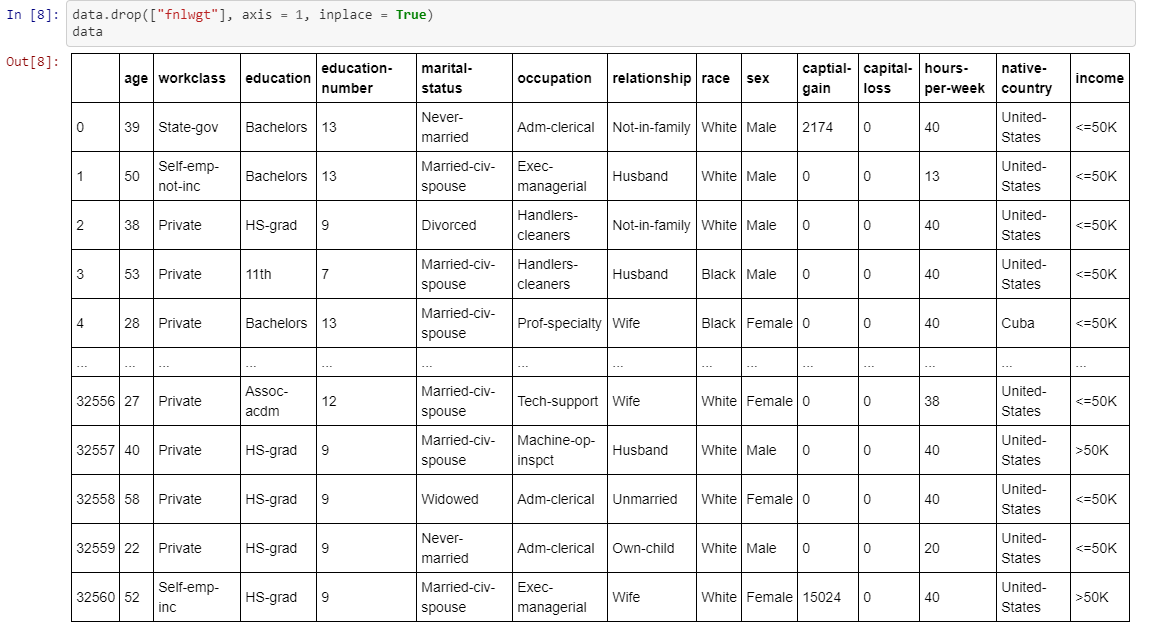
## IMPORTING THE DATA-SET:

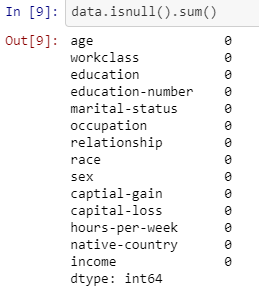
Pandas in python provide an interesting method read\_csv(). The read\_csv function reads the entire dataset from a comma separated values file and we can assign it to a DataFrame to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be access using the dataframe. Any missing value or NaN value have to be cleaned.

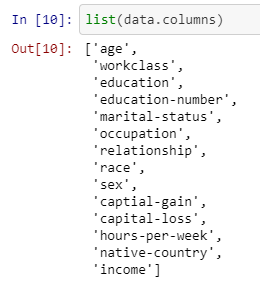


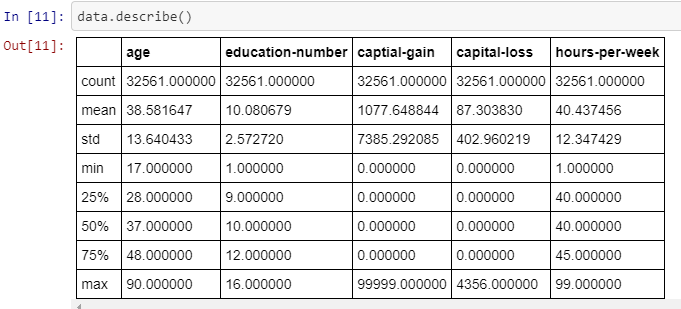


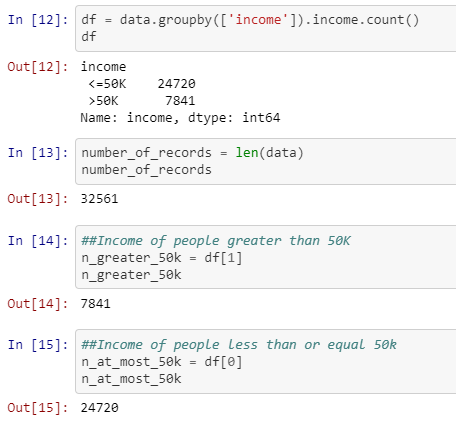








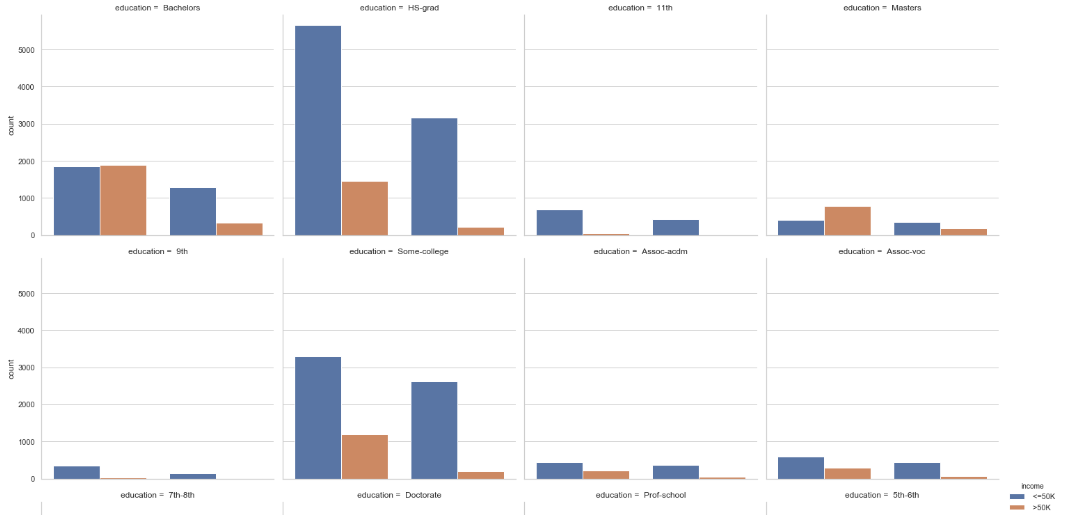


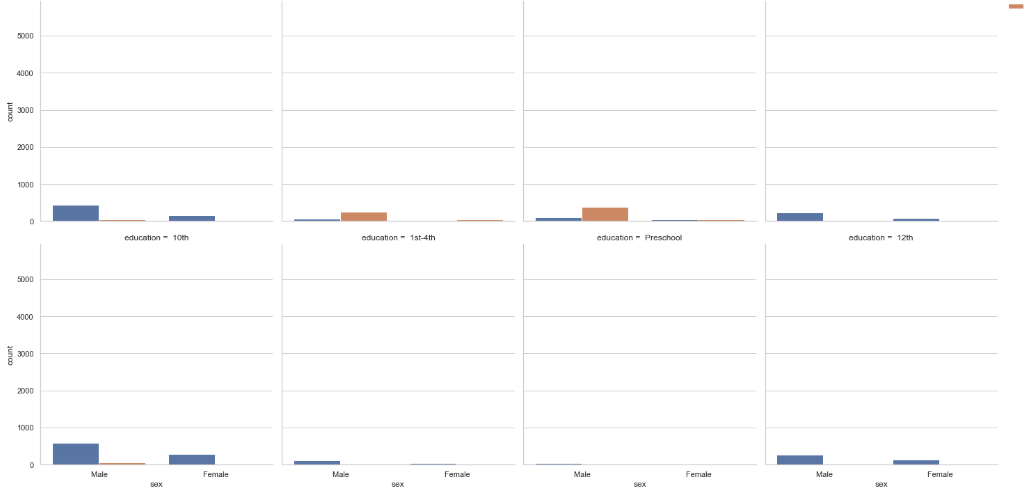


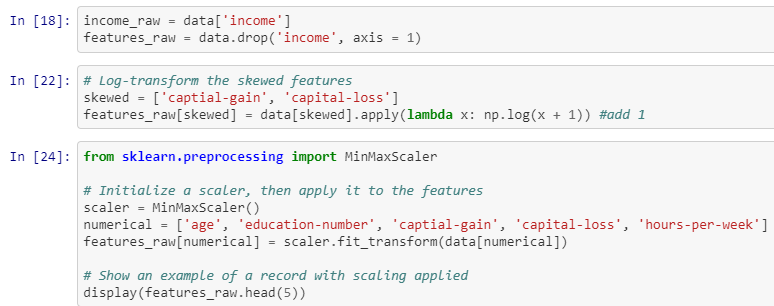
From the above figure we are checking the income of people if it is greater than or less than 50k.



Before we move on to modeling we're going to perform some preprocessing on our dataset to adjust the quality of our variables. For example, since we’re dealing with a monetary response variable income, it’s common to perform log transformations to normalize its distribution.





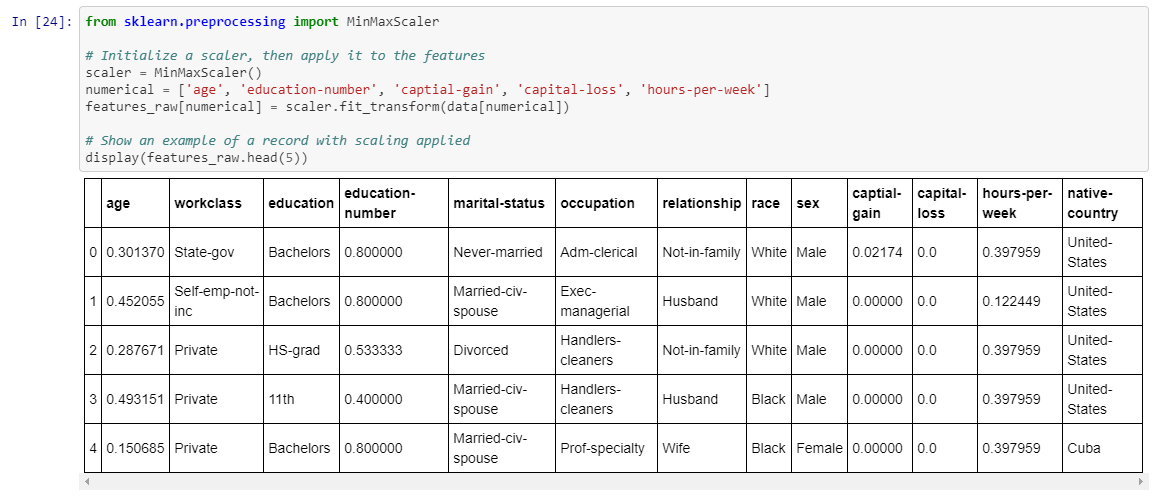


### **Transforming Skewed Continuous Features**

A dataset may sometimes contain at least one feature whose values tend to lie near a single number, but will also have a non-trivial number of vastly larger or smaller values than that single number. Algorithms can be sensitive to such distributions of values and can underperform if the range is not properly normalized. With the census dataset two features fit this description: ‘capital-gain' and 'capital-loss'.

### **Shuffle Split the Data**

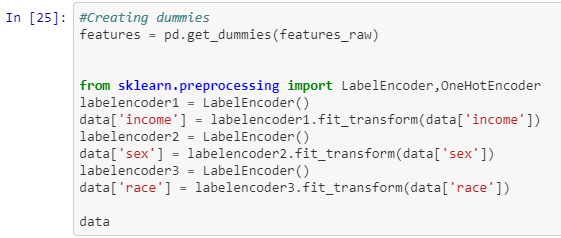
Now that we've transformed and scaled our numeric features and encoded both our categorical features as well as our response variable we're ready to split our data set into training and testing sets.



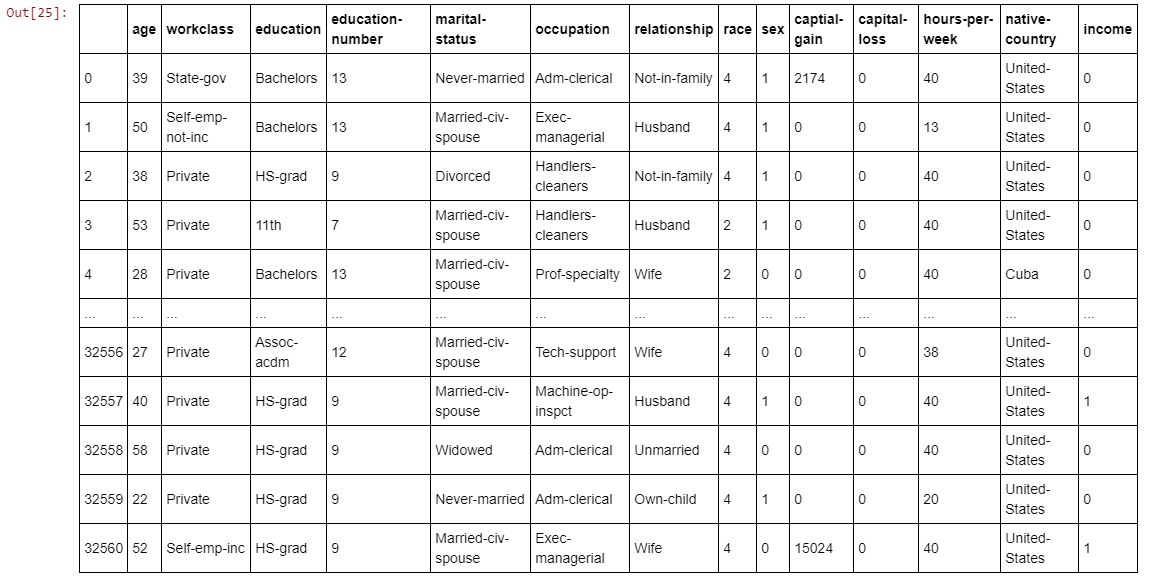
* + Getting dummies using label encoder from scikit learn package

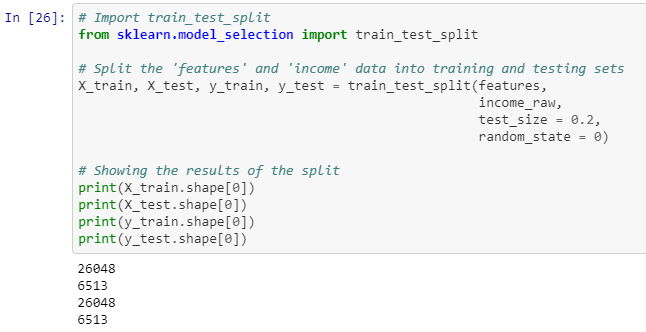
We have a method called label encoder in scikit learn package .we need to import the label encoder method from scikitlearn package and after that we have to fit and transform the data frame to make the categorical data into dummies.

If we use this method to get dummies then in place of categorical data we get the numerical values (0,1,2….)



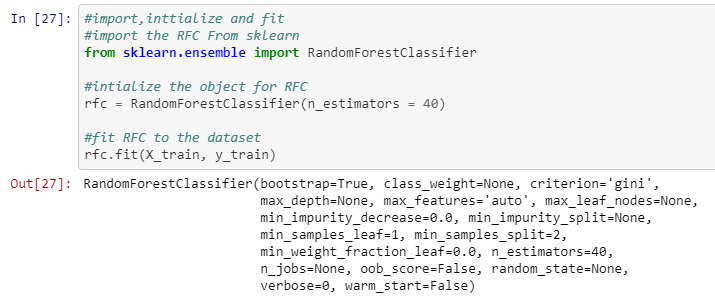
Here we are creating dummies

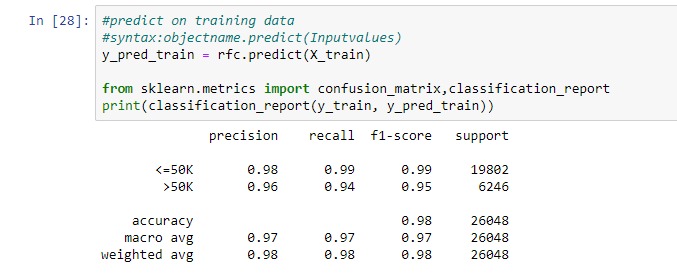


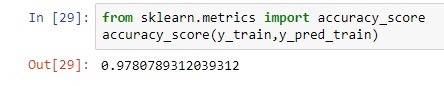


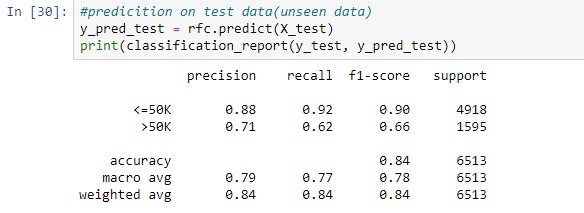
Now all categorical variables have been converted into numerical features, and all numerical features have been normalized. As always, we will now split the data (both features and their labels) into training and test sets. 80% of

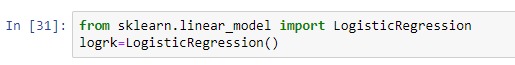
the data will be used for training and 20% for testing.

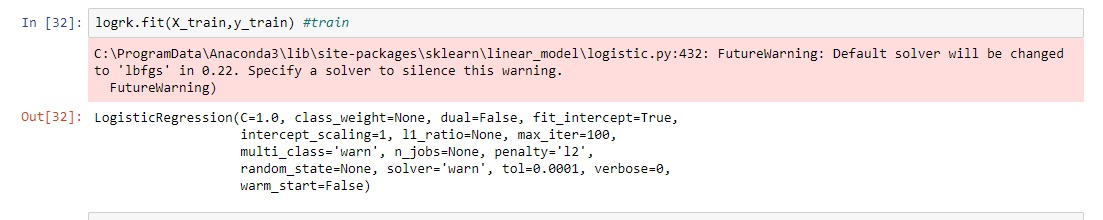




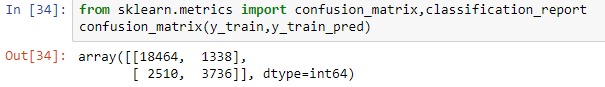


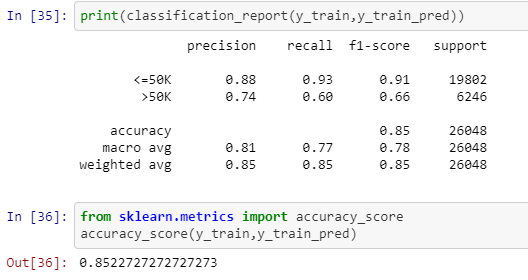


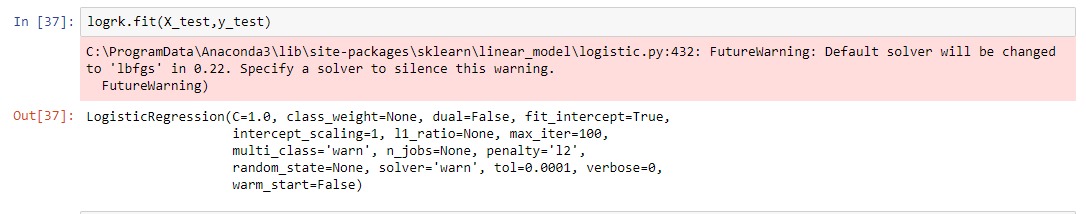


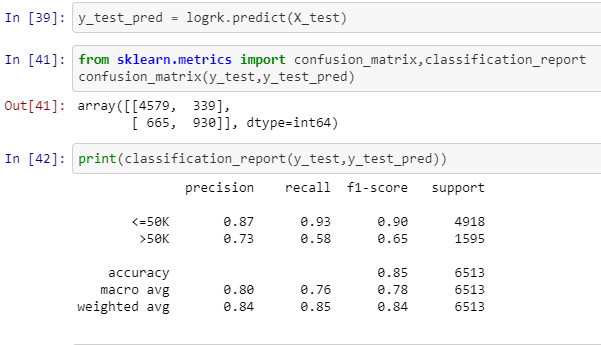


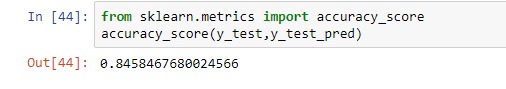


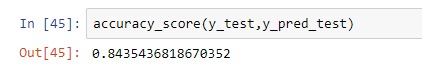


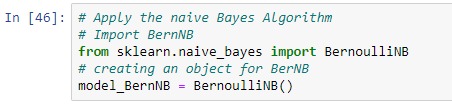


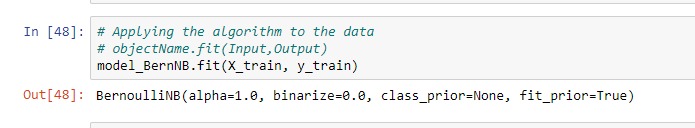




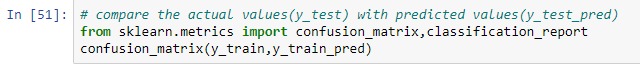


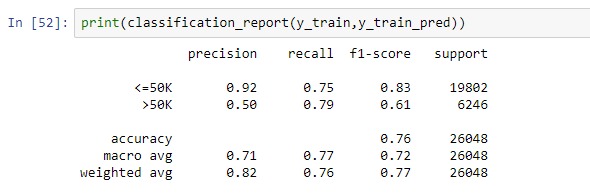


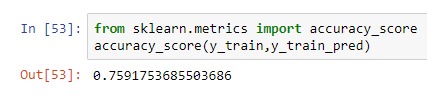


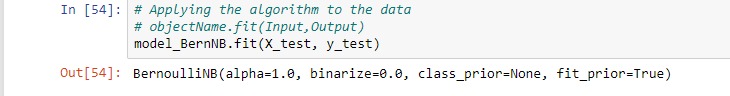




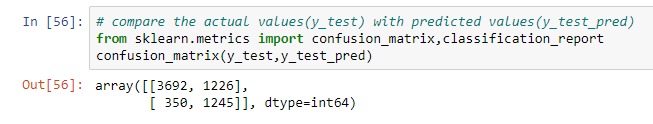


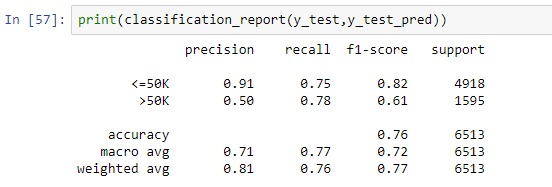


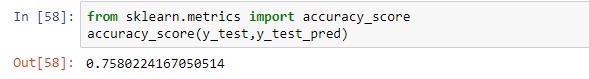












**Conclusion:**

We got a data set and we set a target to classify people that make more than $50,000 annually. We cleaned the data, normalized and converted the necessary variables into numerical features so that we can use them in our models. We shuffled and split our data into training and testing sets.

**References:**

* <https://sajalsharma.com/portfolio/finding_donors>
* <https://etav.github.io/projects/classification_pipeline.html>
* <https://medium.com/@koushik.flitcher/finding-donors-for-charity-using-machine-learning-supervised-learning-e1d1d61cba53>
* <https://towardsdatascience.com/lets-find-donors-for-charity-with-machine-learning-models-7ad1da9f5538>
* <https://towardsdatascience.com/classification-project-finding-donors-853db66fbb8c>