**TABLE OF CONTENTS**

1. **INTRODUCTION** **1**

1.1 OVERVIEW OF THE PROJECT 1

1.2 OBJECTIVE OF THE PROJECT 1

1.3 PROJECT CATEGORY 1

1.4 TOOLS AND PLATFORM TO BE USED 1

1.5 OVERVIEW OF THE TECHNOLOGIES USED 1

1.5.1 Hardware Requirements 1

1.5.2 Software Requirements 1

1.5.3 Python Resources 2

1.6 ORGANIZATION PROFILE 2

1.7 STRUCTURE OF THE PROGRAM 2

1.8 STATEMENT OF THE PROBLEM 3

1. **METHODOLOGY** **3**

2.1 Data Gathering 4

2.2 Preprocessing and Cleaning 4

2.3 Data Splitting 8

2.4 Training the Data 9

2.4.1 Logistic Regression 9

2.4.2 Naïve Bayes 10

2.4.3 Kneighbors Classifier 12

2.4.4 Decision Tree Classifier 13

2.4.5 Random Forrest Classifier 14

2.4.6 Support Vector Classifier 15

1. **CODING** **17**
2. **LIST OF ABBREVIATION** **72**
3. **CONCLUSION** **73**
4. **BIBILIOGRAPHY** **74**

**1.INTRODUCTION**

1.1 OVERVIEW OF THE PROJECT

Placement of scholars is one in every of the vital activities in academic establishments. Admission and name of establishments primarily depends on placements. Hence all institutions strive to Strengthen placement department. The main Objective is to analyze previous year’s student’s historical data and predict placement possibilities of current students and aids to increase the placement percentage of the institutions. The system that predicts whether the current student will be placed or not based on the data of previously placed students. Different machine learning classification algorithms, namely Logistic Regression, Naive Bayes Classifier KNearest Neighbors [KNN], Decision Tree , Random Forest and Support Vector algorithm are used. These algorithms independently predict the results and we then compare the efficiency of the algorithms, which is based on the dataset. This model helps the position cell at intervals a corporation to spot the potential students and concentrate to and improve their technical and social skills.

1.2 OBJECTIVE OF THE PROJECT

The main Objective of the project is to analyze previous year’s student’s historical data and predict placement possibilities of current students and aids to increase the placement percentage of the institutions. The system predicts whether the current student will be placed or not.

This model helps the position cell at intervals a corporation to spot the potential students and concentrate to and improve their technical and social skills.

1.3 PROJECT CATEGORY

Machine Learning using Python

1.4 TOOLS AND PLATFORM TO BE USED

Anaconda IDE and Jupyter Notebook

Kaggle dataset

1.5 OVERVIEW OF THE TECHNOLOGIES USED

1.5.1 Hardware Requirements

1.5.2 Software Requirements

Language: Python

IDE: Anaconda

Python Libraries : Scikit- Learn, Pandas, Numpy, Matplotlib

1.5.3 Python Resources

Python: Python is a very powerful and flexible programming language . This is an open-source language has created quite a few tools to efficiently work with a python. There are some basic libraries which are essential for building a project.

Pandas: Pandas is a python package created for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Matplotlib: Matplotlib is a 2D plotting python library, with which we can plot various plots in python across various environments. It is an alternate option to seaborn, and seaborn is related with matplotlib.

Scikit-learn: It is an easy-to-use Python library that is used to build a machine learning model. It is built on NumPy, SciPy, and matplotlib. Below is the official documentation for the sci-kit learn library.

Numpy: NumPy is a python library also called as Numeric python which can execute scientific computing. You all must know that python never provides an array data structure, only with the help of a NumPy library you can create and perform manipulations on an array.

1.6 ORGANIZATION PROFILE

Cognitive Solutions a rapidly growing company in the field of computer application implementation, solutions and services. Cognitive Solutions a service provider of Web-based Development & Web based Software Development Solutions, Mobile Application Development, Graphic Design and Windows Applications.

Cognitive Solutions headquartered in Mangalore ,with the Business Development in UAE,Saudi Arabia Qatar. In short span of 8+ years, our products as well as services & solutions have been widely accepted by the global market. Today ,Cognitive Solution has the experience to undertake any IT development or deployment works on a single point responsibility basis.

Our efficient and experienced team is greatest resource. Intellect's infrastructure houses A-team of young and competitive professional having experience in Web Designing and Software Development who are dedicated to providing high-end solution to our clients .We develop software and web-based applications with latest technologies. For web development projects, we also provide hosting and facility for customers, so they don’t need to bother about that .Our Products Services are user friendly with easy controls and are of superior specifications. We are always proactive to fulfil client's needs and requirements to the best possible extent of their satisfaction. We manage interactive sessions with clients throughout project development.

1.7 STRUCTURE OF THE PROGRAM

Placement Prediction Module: This module predicts whether the student will be recruited or not based on the details provided.

1.8 STATEMENT OF THE PROBLEM

This data set consists of Placement data of students in a XYZ campus. It includes secondary and higher secondary school percentage and specialization. It also includes degree specialization, type and Work experience and salary offers to the placed students, Based on the details the system will predicted if a student will be placed or not.

**2. METHODOLOGY**

**DATA GATHERING**

**DATA PREPROCESSING AND CLEANING**

**SPLITTING THE DATA**

**FEATURE SELECTION**

**TEST MODEL**

**TRAIN MODEL**

**OUTPUT**

**EVALUATE MODEL**

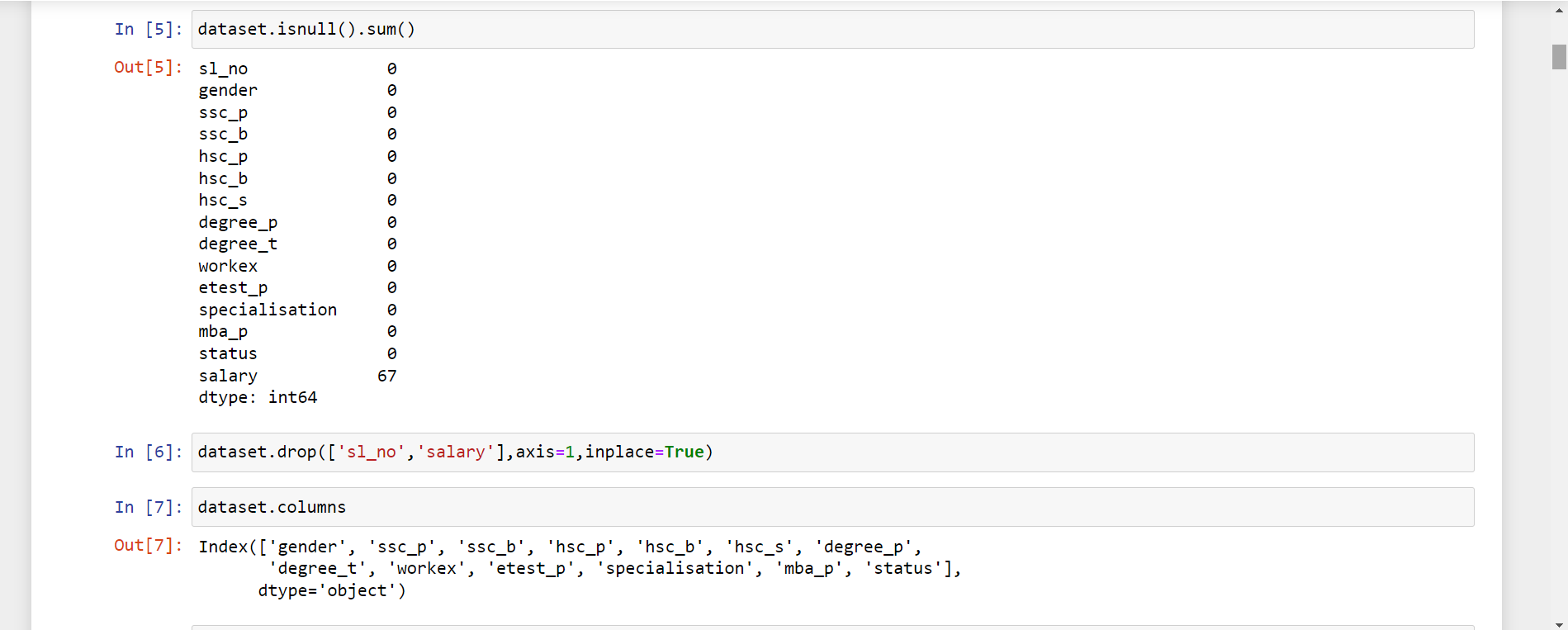
### 2.1 Data Gathering

In this phase gather the previous year data from training and placement Office. The combination of various attributes determines whether the student is placed or not.

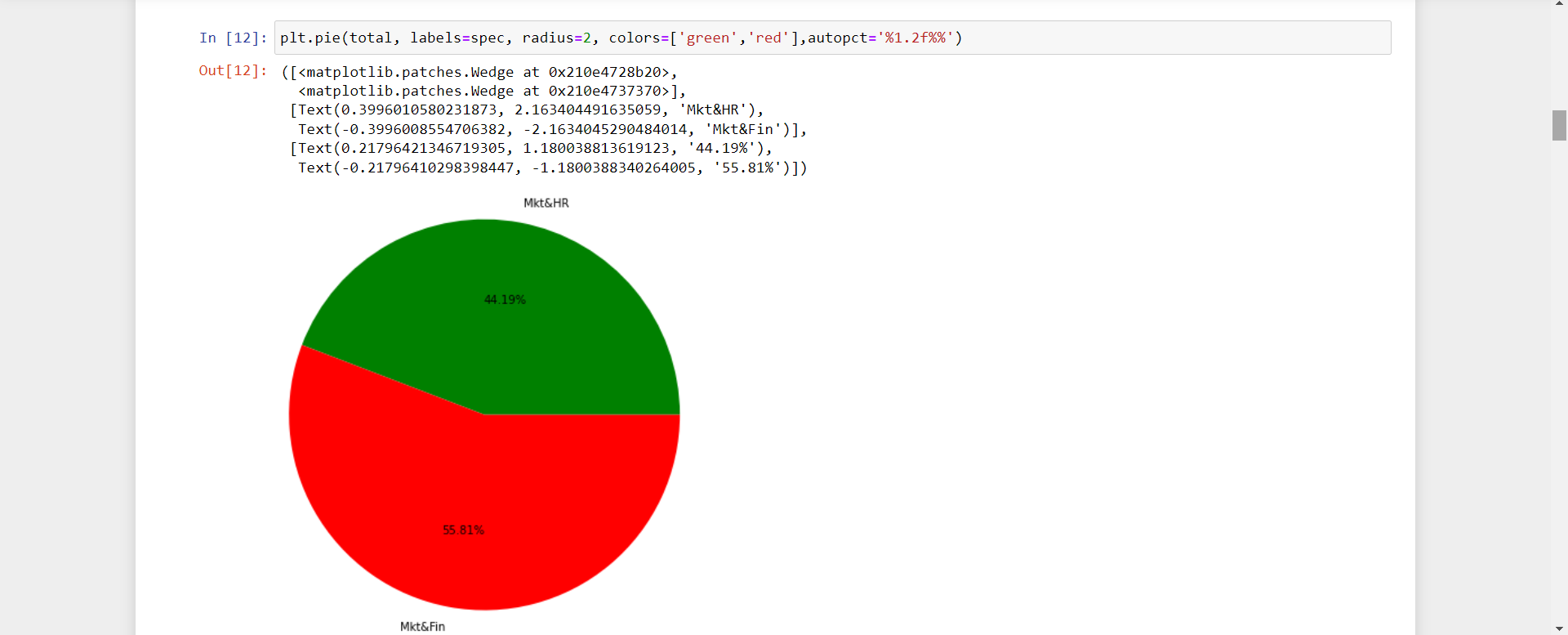
### 

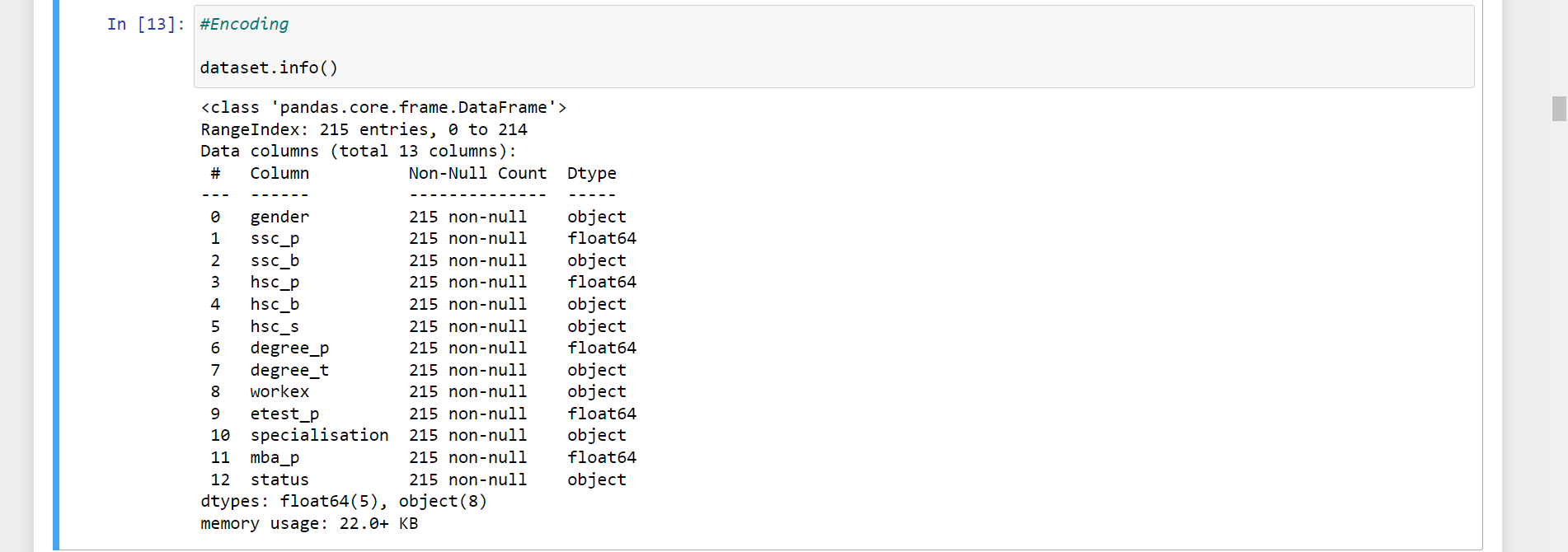
### 2.2 Preprocessing and Cleaning

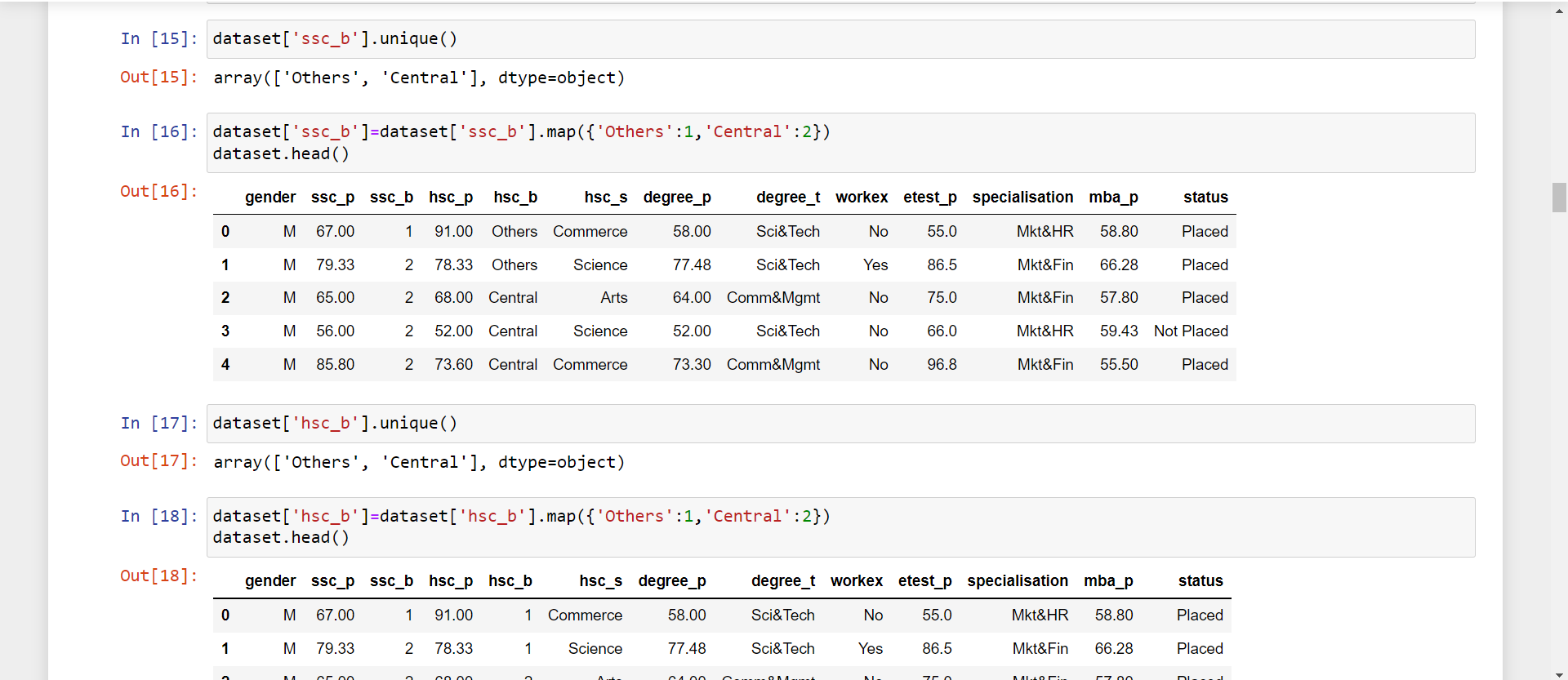
Data preprocessing is the first step towards building a data science model. It is a technique that is used to convert the raw dataset into a clean dataset. Whenever the data is gathered from different sources it is gathered in raw format which is not feasible for the analysis. Therefore certain steps like data cleaning, data integration, encoding, transformation, and reduction are executed to convert the data into a clean data set.







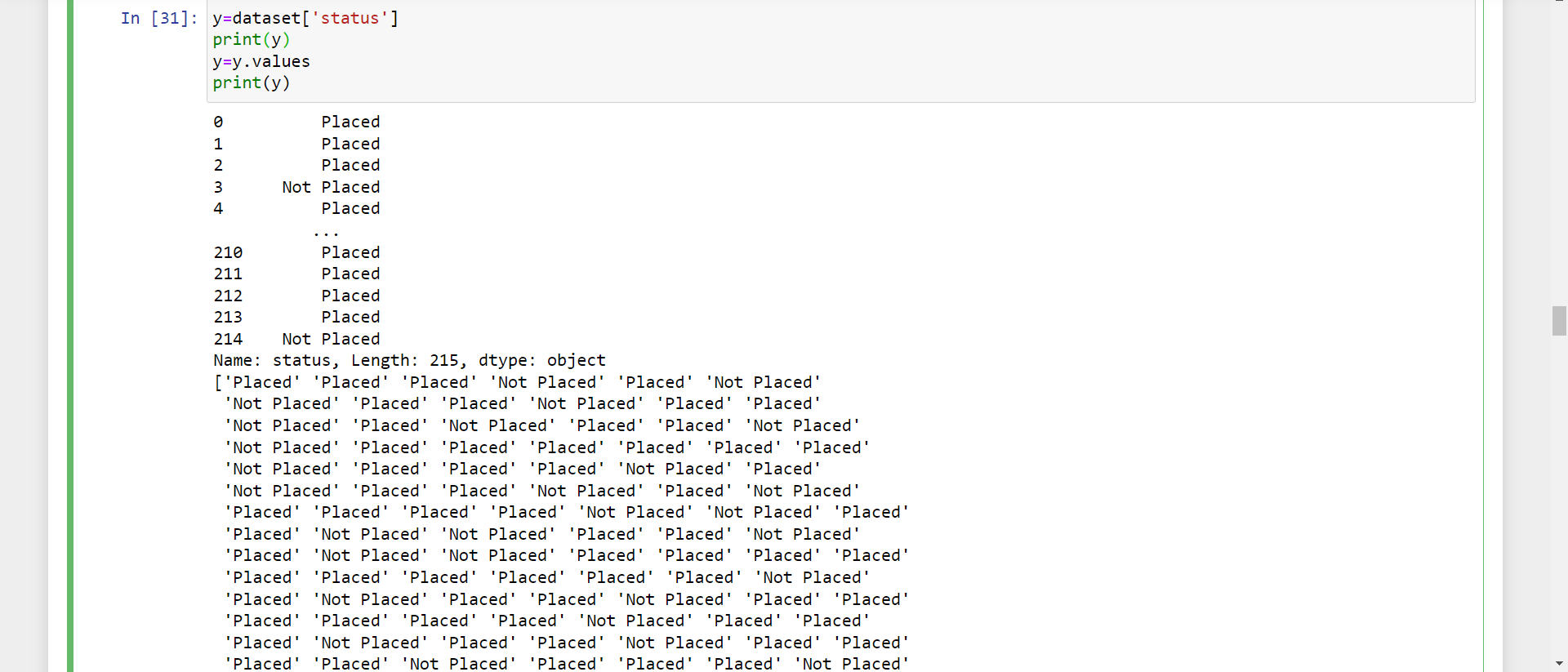


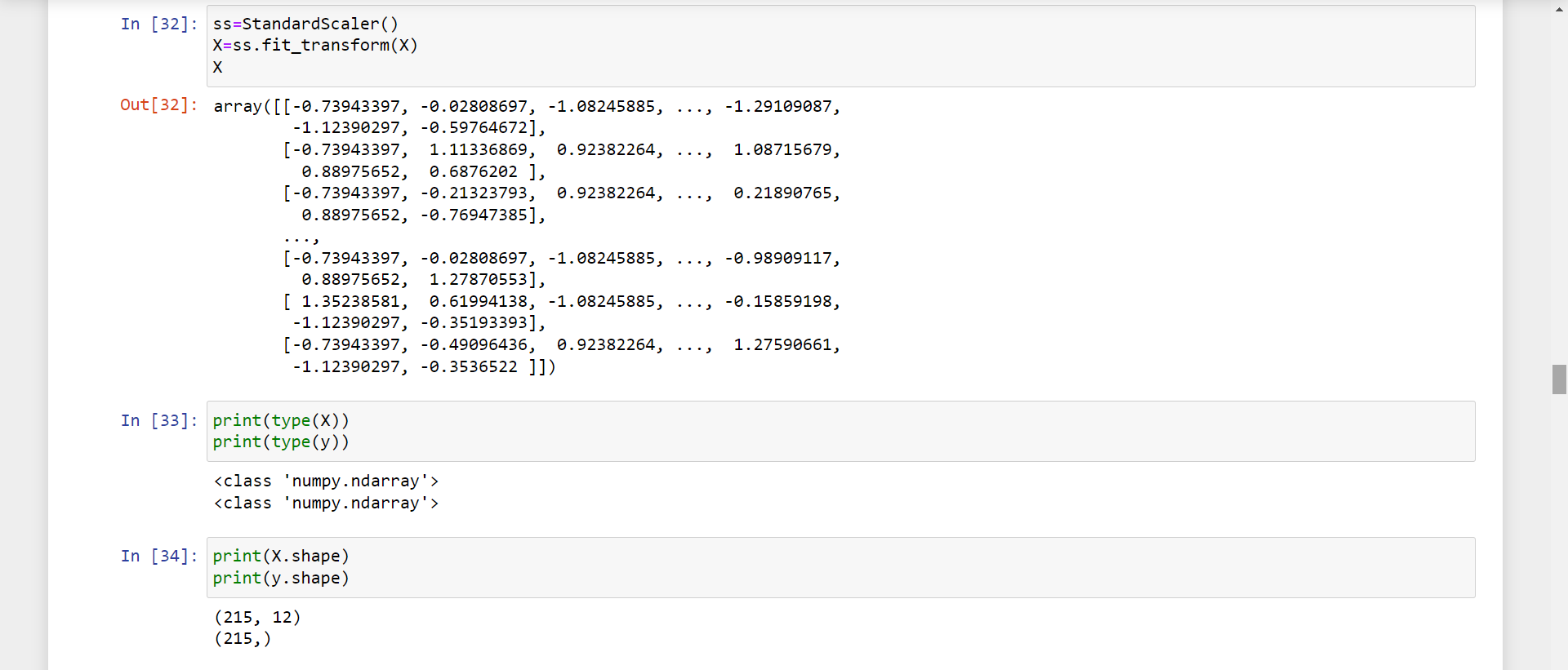








****

****

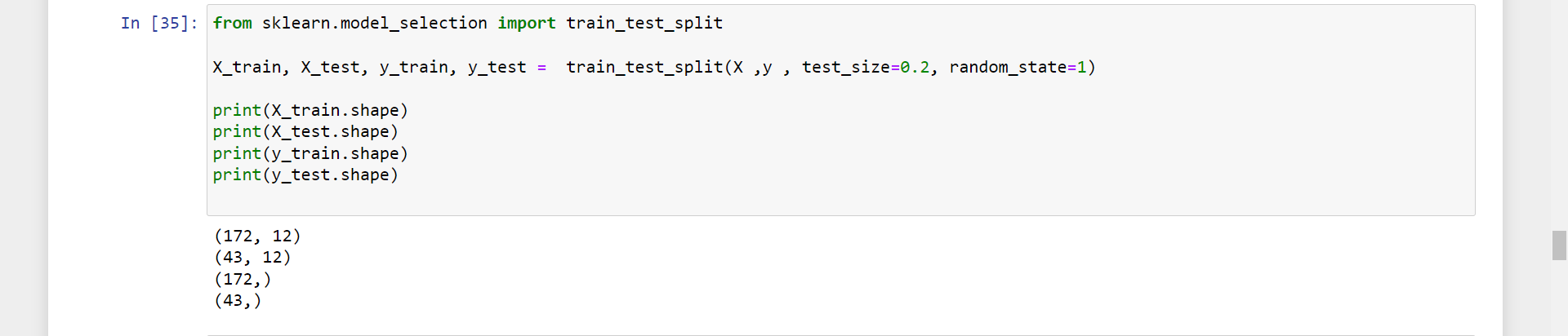
**2.3 Data Splitting**

In machine learning, data splitting is typically done to avoid [overfitting](https://www.techtarget.com/whatis/definition/overfitting). That is an instance where a machine learning model fits its training data too well and fails to reliably fit additional data.

The original data in a machine learning model is typically taken and split into three or four sets. The three sets commonly used are the training set, the dev set and the testing set:

1. The training set is the portion of data used to train the model. The model should observe and learn from the training set, optimizing any of its parameters.
2. The dev set is a data set of examples used to change learning process parameters. It is also called the *cross-validation* or *model*[validation set](https://www.techtarget.com/whatis/definition/validation-set). This set of data has the goal of ranking the model's accuracy and can help with model selection.
3. The testing set is the portion of data that is tested in the final model and is compared against the previous sets of data. The testing set acts as an evaluation of the final mode and algorithm.

Data should be split so that data sets can have a high amount of training data. For example, data might be split at an 80-20 or a 70-30 ratio of training vs. testing data. The exact ratio depends on the data, but a 70-20-10 ratio for training, dev and test splits is optimal for small data sets.

****

**2.4 Training the Data**

**1. Logistic Regression**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

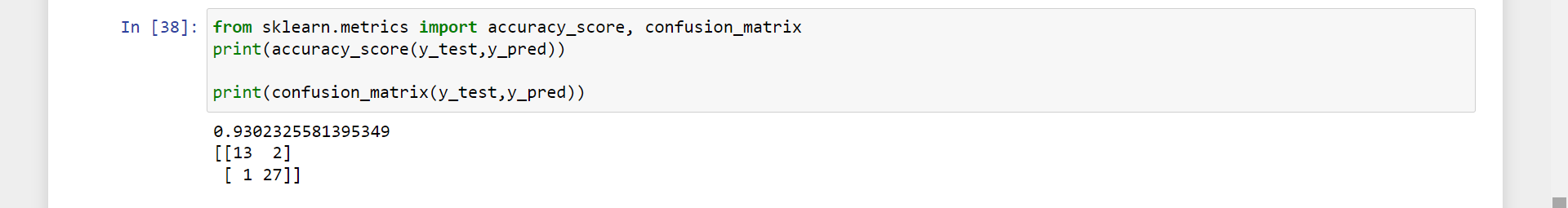
Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

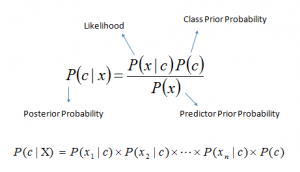
****

****

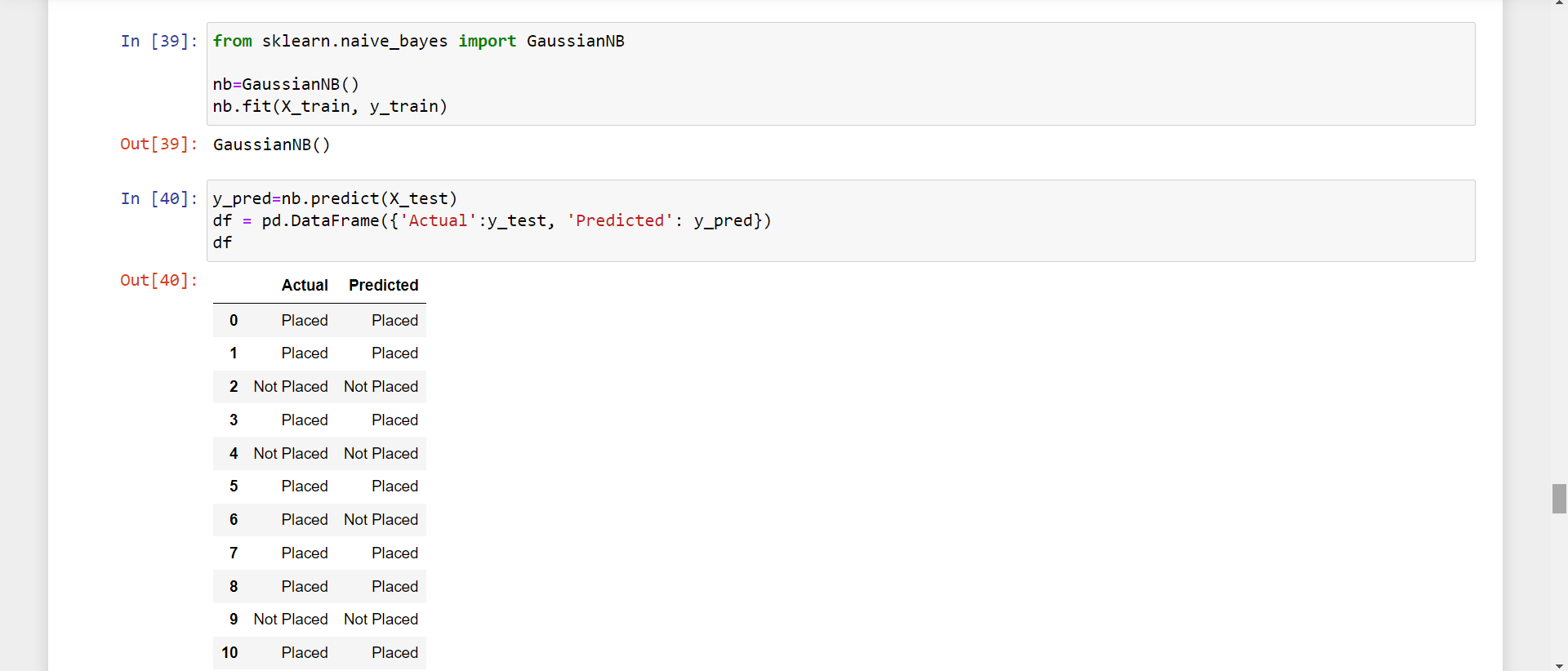
**The accuracy score for Logistic Regression is 93%**

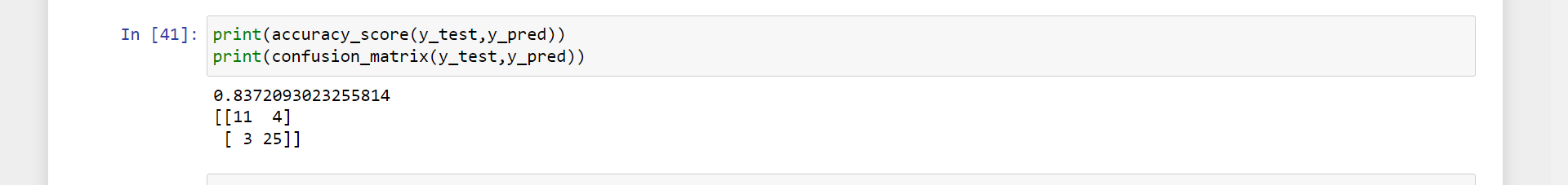
**2) Naïve Bayes:**

Naive Bayes model is quiet easy to build and specially useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Bayes theorem supply a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



Above,  
P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).  
P(c) is the prior probability of class.  
P(x|c) is the likelihood which is the probability of predictor given class.  
P(x) is the prior probability of predictor.





**Accuracy Score of Naïve Bayes is 83 %**

**3) KNeighborsClassifier:**

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps –

Step 1 − For applying any algorithm, we require a dataset. So throughout the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to select the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following –

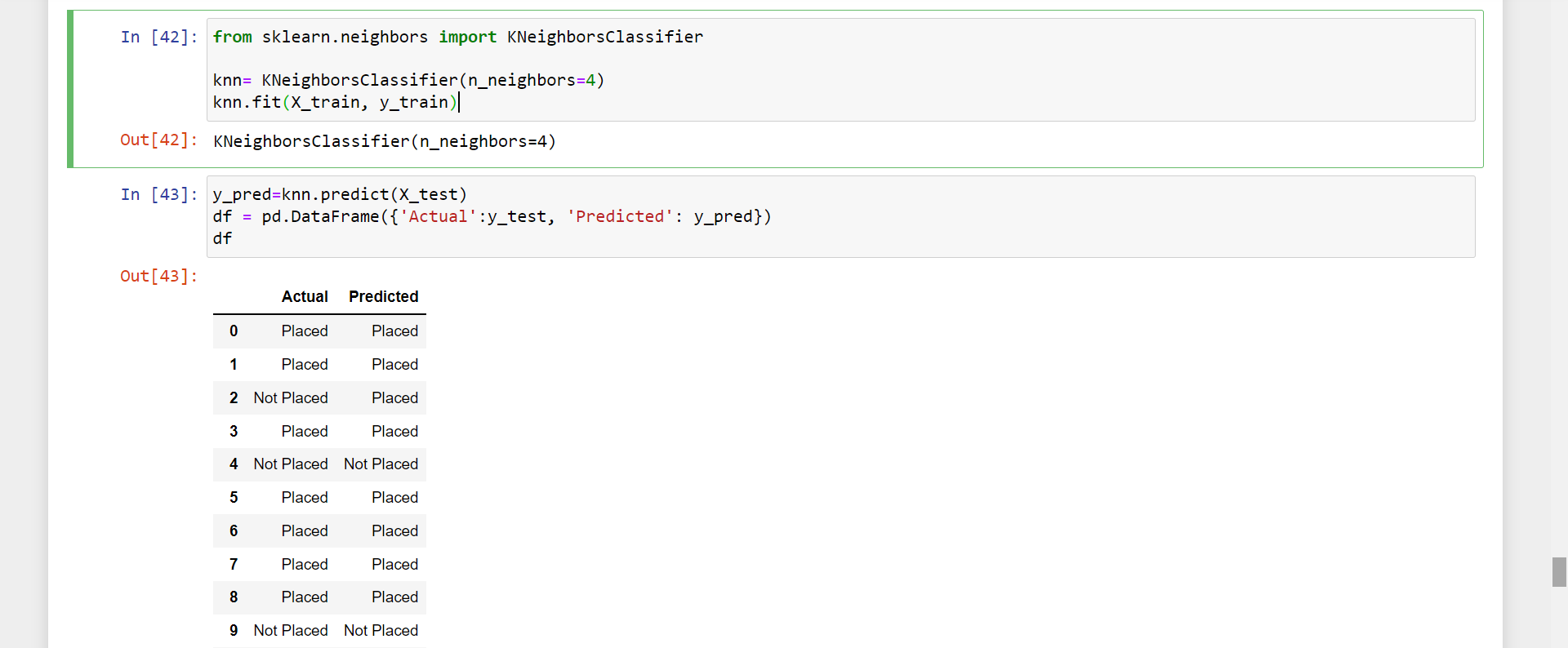
3.1 − Compute the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most used method to calculate distance is Euclidean.

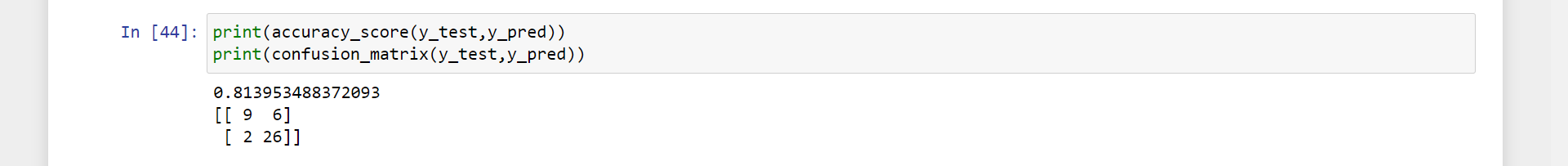
3.2 − Now, based on the distance value, sort them in ascending order.

3.3 − − Next, it will select the top K rows from the sorted array.

3.4 − Now, it will allocate a class to the test point based on the most frequent class of these rows.

Step 4 − End

****

****

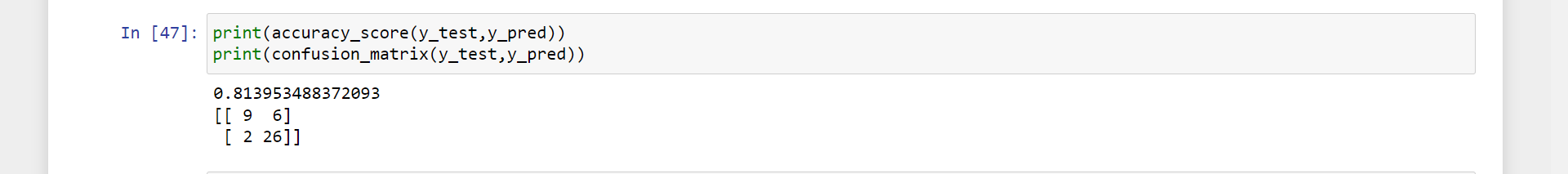
**Accuracy Score of KNN = 81%**

**4) Decision Tree Classifier:**

Decision Tree: In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record’s attribute. Based on the comparison, we follow the branch corresponding to that value and jump to the next node. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data).

1. Put down the best attribute of the dataset at the root of the tree.
2. Break the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
3. Repeat step 1 and step 2 on each subset till you find leaf nodes in all the branches of the tree

****

****

**Accuracy Score of Decision Tree Classifier: 81%**

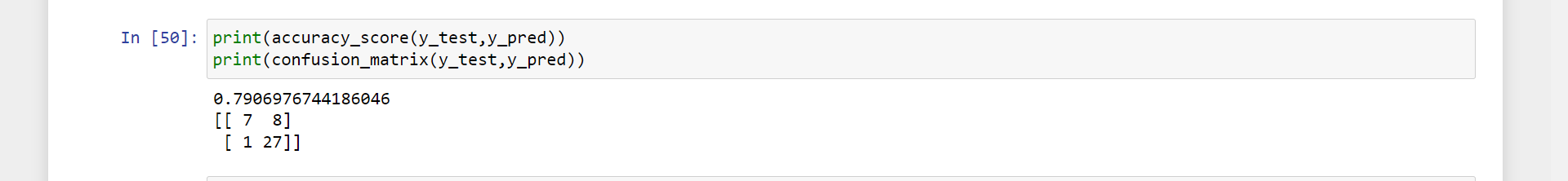
**5) Random Forest Classifier**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting**.**

****

****

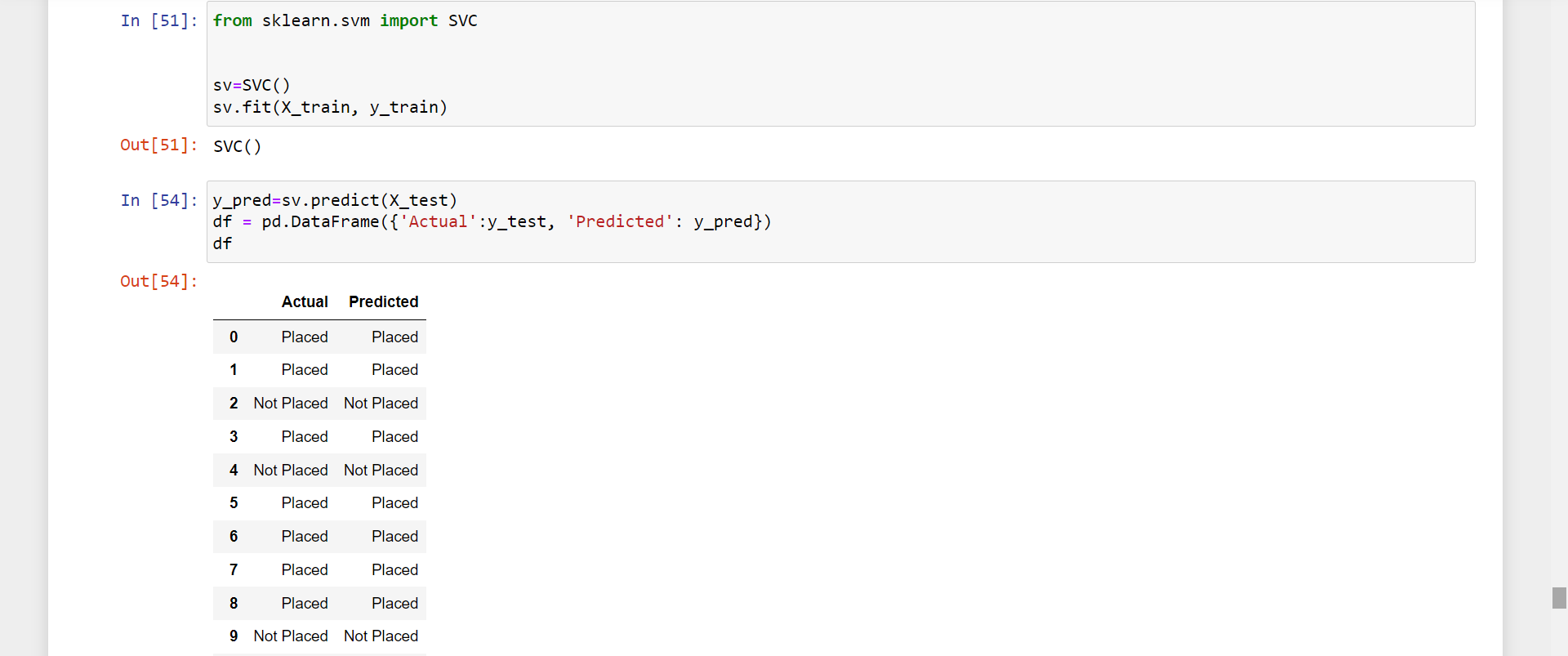
**Accuracy Score of Random Forest : 79%**

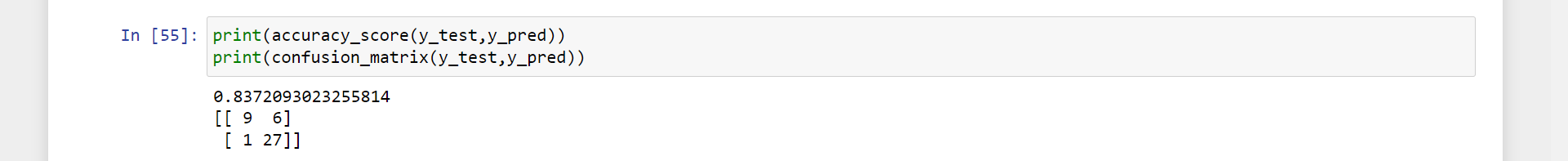
**6) Support Vector Classifier:**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

****

****

**Accuracy of SVC : 84%**

**CODING**

{

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"id": "d159901b",

"metadata": {},

"outputs": [],

"source": [

"import pandas as pd\n",

"import numpy as np"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "71f3e699",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>sl\_no</th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" <th>salary</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>Others</td>\n",

" <td>91.00</td>\n",

" <td>Others</td>\n",

" <td>Commerce</td>\n",

" <td>58.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" <td>270000.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2</td>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>Central</td>\n",

" <td>78.33</td>\n",

" <td>Others</td>\n",

" <td>Science</td>\n",

" <td>77.48</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>Yes</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" <td>200000.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>3</td>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>Central</td>\n",

td>Arts</td>\n",

" <td>64.00</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" <td>250000.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>4</td>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>Central</td>\n",

" <td>52.00</td>\n",

" <td>Central</td>\n",

" <td>Science</td>\n",

" <td>52.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>5</td>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>Central</td>\n",

" <td>73.60</td>\n",

" <td>Central</td>\n",

" <td>Commerce</td>\n",

" <td>73.30</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" <td>425000.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" sl\_no gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p \\\n",

"0 1 M 67.00 Others 91.00 Others Commerce 58.00 \n",

"1 2 M 79.33 Central 78.33 Others Science 77.48 \n",

"2 3 M 65.00 Central 68.00 Central Arts 64.00 \n",

"3 4 M 56.00 Central 52.00 Central Science 52.00 \n",

"4 5 M 85.80 Central 73.60 Central Commerce 73.30 \n",

"\n",

" degree\_t workex etest\_p specialisation mba\_p status salary \n",

"0 Sci&Tech No 55.0 Mkt&HR 58.80 Placed 270000.0 \n",

"1 Sci&Tech Yes 86.5 Mkt&Fin 66.28 Placed 200000.0 \n",

"2 Comm&Mgmt No 75.0 Mkt&Fin 57.80 Placed 250000.0 \n",

"3 Sci&Tech No 66.0 Mkt&HR 59.43 Not Placed NaN \n",

"4 Comm&Mgmt No 96.8 Mkt&Fin 55.50 Placed 425000.0 "

]

},

"execution\_count": 2,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset= pd.read\_csv('Placement\_Data\_Full\_Class.csv')\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "888b6fa0",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Index(['sl\_no', 'gender', 'ssc\_p', 'ssc\_b', 'hsc\_p', 'hsc\_b', 'hsc\_s',\n",

" 'degree\_p', 'degree\_t', 'workex', 'etest\_p', 'specialisation', 'mba\_p',\n",

" 'status', 'salary'],\n",

" dtype='object')"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.columns\n"

]

},

},

{

"cell\_type": "code",

"execution\_count": 4,

"id": "76a0d597",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(215, 15)"

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "a015dc60",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"sl\_no 0\n",

"gender 0\n",

"ssc\_p 0\n",

"ssc\_b 0\n",

"hsc\_p 0\n",

"hsc\_b 0\n",

"hsc\_s 0\n",

"degree\_p 0\n",

"degree\_t 0\n",

"workex 0\n",

"etest\_p 0\n",

"specialisation 0\n",

"mba\_p 0\n",

"status 0\n",

"salary 67\n",

"dtype: int64"

]

},

"execution\_count": 5,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.isnull().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "e40c68ac",

"metadata": {},

"outputs": [],

"source": [

"dataset.drop(['sl\_no','salary'],axis=1,inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "7e2ed95e",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Index(['gender', 'ssc\_p', 'ssc\_b', 'hsc\_p', 'hsc\_b', 'hsc\_s', 'degree\_p',\n",

" 'degree\_t', 'workex', 'etest\_p', 'specialisation', 'mba\_p', 'status'],\n",

" dtype='object')"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.columns"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "be126343",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"gender 0\n",

"ssc\_p 0\n",

"ssc\_b 0\n",

"hsc\_p 0\n",

"hsc\_b 0\n",

"hsc\_s 0\n",

"degree\_p 0\n",

"degree\_t 0\n",

"workex 0\n",

"etest\_p 0\n",

"specialisation 0\n",

"mba\_p 0\n",

"status 0\n",

"dtype: int64"

]

},

"execution\_count": 8,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.isnull().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "aa825377",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Mkt&HR', 'Mkt&Fin'], dtype=object)"

]

},

"execution\_count": 9,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['specialisation'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"id": "8945d9e9",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[95, 120]\n"

]

}

],

"source": [

"#Visualization\n",

"\n",

"total=[sum(dataset['specialisation']=='Mkt&HR'),\n",

" sum(dataset['specialisation']=='Mkt&Fin')]\n",

"\n",

"print(total)\n"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "2277abd1",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<function matplotlib.pyplot.show(close=None, block=None)>"

]

},

"execution\_count": 11,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png":

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

,

"source": [

"import matplotlib.pyplot as plt\n",

"\n",

"spec=['Mkt&HR', 'Mkt&Fin']\n",

"plt.bar(spec,total,color=['green','red'])\n",

"plt.show"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "a6bc5ff1",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"([<matplotlib.patches.Wedge at 0x2861c879580>,\n",

" <matplotlib.patches.Wedge at 0x2861c879d00>],\n",

" [Text(0.3996010580231873, 2.163404491635059, 'Mkt&HR'),\n",

" Text(-0.3996008554706382, -2.1634045290484014, 'Mkt&Fin')],\n",

" [Text(0.21796421346719305, 1.180038813619123, '44.19%'),\n",

" Text(-0.21796410298398447, -1.1800388340264005, '55.81%')])"

]

},

"execution\_count": 12,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png":

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plt.pie(total, labels=spec, radius=2, colors=['green','red'],autopct='%1.2f%%')"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"id": "d0710f20",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 215 entries, 0 to 214\n",

"Data columns (total 13 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 gender 215 non-null object \n",

" 1 ssc\_p 215 non-null float64\n",

" 2 ssc\_b 215 non-null object \n",

" 3 hsc\_p 215 non-null float64\n",

" 4 hsc\_b 215 non-null object \n",

" 5 hsc\_s 215 non-null object \n",

" 6 degree\_p 215 non-null float64\n",

" 7 degree\_t 215 non-null object \n",

" 8 workex 215 non-null object \n",

" 9 etest\_p 215 non-null float64\n",

" 10 specialisation 215 non-null object \n",

" 11 mba\_p 215 non-null float64\n",

" 12 status 215 non-null object \n",

"dtypes: float64(5), object(8)\n",

"memory usage: 22.0+ KB\n"

]

}

],

"source": [

"#Encoding\n",

"\n",

"dataset.info()"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"id": "c86db6f3",

"metadata": {},

"outputs": [],

"source": [

"#ssc\_b,hsc\_b,hsc\_s,degree\_t,workex,specialisation"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"id": "c143bcab",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Others', 'Central'], dtype=object)"

]

},

"execution\_count": 15,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['ssc\_b'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"id": "67012aaf",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

<th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>Others</td>\n",

" <td>Commerce</td>\n",

" <td>58.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>Others</td>\n",

" <td>Science</td>\n",

" <td>77.48</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>Yes</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>Central</td>\n",

" <td>Arts</td>\n",

" <td>64.00</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>Central</td>\n",

" <td>Science</td>\n",

" <td>52.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>Central</td>\n",

" <td>Commerce</td>\n",

" <td>73.30</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 Others Commerce 58.00 Sci&Tech No \n",

"1 M 79.33 2 78.33 Others Science 77.48 Sci&Tech Yes \n",

"2 M 65.00 2 68.00 Central Arts 64.00 Comm&Mgmt No \n",

"3 M 56.00 2 52.00 Central Science 52.00 Sci&Tech No \n",

"4 M 85.80 2 73.60 Central Commerce 73.30 Comm&Mgmt No \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 Mkt&HR 58.80 Placed \n",

"1 86.5 Mkt&Fin 66.28 Placed \n",

"2 75.0 Mkt&Fin 57.80 Placed \n",

"3 66.0 Mkt&HR 59.43 Not Placed \n",

"4 96.8 Mkt&Fin 55.50 Placed "

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['ssc\_b']=dataset['ssc\_b'].map({'Others':1,'Central':2})\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"id": "703f550d",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Others', 'Central'], dtype=object)"

]

},

"execution\_count": 17,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['hsc\_b'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"id": "39cb139b",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

}\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>Commerce</td>\n",

" <td>58.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>Science</td>\n",

" <td>77.48</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>Yes</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>Arts</td>\n",

" <td>64.00</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>Science</td>\n",

" <td>52.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>Commerce</td>\n",

" <td>73.30</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 1 Commerce 58.00 Sci&Tech No \n",

"1 M 79.33 2 78.33 1 Science 77.48 Sci&Tech Yes \n",

"2 M 65.00 2 68.00 2 Arts 64.00 Comm&Mgmt No \n",

"3 M 56.00 2 52.00 2 Science 52.00 Sci&Tech No \n",

"4 M 85.80 2 73.60 2 Commerce 73.30 Comm&Mgmt No \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 Mkt&HR 58.80 Placed \n",

"1 86.5 Mkt&Fin 66.28 Placed \n",

"2 75.0 Mkt&Fin 57.80 Placed \n",

"3 66.0 Mkt&HR 59.43 Not Placed \n",

"4 96.8 Mkt&Fin 55.50 Placed "

]

},

"execution\_count": 18,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['hsc\_b']=dataset['hsc\_b'].map({'Others':1,'Central':2})\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 19,

"id": "0f1e6621",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Commerce', 'Science', 'Arts'], dtype=object)"

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['hsc\_s'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"id": "10d54a43",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

<th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>Yes</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>Sci&amp;Tech</td>\n",

" <td>No</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>73.30</td>\n",

" <td>Comm&amp;Mgmt</td>\n",

" <td>No</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 1 1 58.00 Sci&Tech No \n",

"1 M 79.33 2 78.33 1 2 77.48 Sci&Tech Yes \n",

"2 M 65.00 2 68.00 2 3 64.00 Comm&Mgmt No \n",

"3 M 56.00 2 52.00 2 2 52.00 Sci&Tech No \n",

"4 M 85.80 2 73.60 2 1 73.30 Comm&Mgmt No \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 Mkt&HR 58.80 Placed \n",

"1 86.5 Mkt&Fin 66.28 Placed \n",

"2 75.0 Mkt&Fin 57.80 Placed \n",

"3 66.0 Mkt&HR 59.43 Not Placed \n",

"4 96.8 Mkt&Fin 55.50 Placed "

]

},

"execution\_count": 20,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['hsc\_s']=dataset['hsc\_s'].map({'Commerce':1, 'Science':2, 'Arts':3})\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"id": "c89ff73c",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Sci&Tech', 'Comm&Mgmt', 'Others'], dtype=object)"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['degree\_t'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"id": "dfa67979",

"metadata": {},

"outputs": [],

"source": [

"dataset['degree\_t']=dataset['degree\_t'].map({'Sci&Tech':1,'Comm&Mgmt':2, 'Others':3 })"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"id": "2fdeac42",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>No</td>\n",

" <td>55.0</td>\n",

"cell\_type": "code",

"execution\_count": 21,

"id": "c89ff73c",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Sci&Tech', 'Comm&Mgmt', 'Others'], dtype=object)"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['degree\_t'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"id": "dfa67979",

"metadata": {},

"outputs": [],

"source": [

"dataset['degree\_t']=dataset['degree\_t'].map({'Sci&Tech':1,'Comm&Mgmt':2, 'Others':3 })"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"id": "2fdeac42",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>No</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>1</td>\n",

" <td>Yes</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>2</td>\n",

" <td>No</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>1</td>\n",

" <td>No</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>73.30</td>\n",

" <td>2</td>\n",

" <td>No</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 1 1 58.00 1 No \n",

"1 M 79.33 2 78.33 1 2 77.48 1 Yes \n",

"2 M 65.00 2 68.00 2 3 64.00 2 No \n",

"3 M 56.00 2 52.00 2 2 52.00 1 No \n",

"4 M 85.80 2 73.60 2 1 73.30 2 No \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 Mkt&HR 58.80 Placed \n",

"1 86.5 Mkt&Fin 66.28 Placed \n",

"2 75.0 Mkt&Fin 57.80 Placed \n",

"3 66.0 Mkt&HR 59.43 Not Placed \n",

"4 96.8 Mkt&Fin 55.50 Placed "

]

},

"execution\_count": 23,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"id": "668f7194",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['No', 'Yes'], dtype=object)"

]

},

"execution\_count": 24,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['workex'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"id": "cf95ef4a",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

"cell\_type": "code",

"execution\_count": 21,

"id": "c89ff73c",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Sci&Tech', 'Comm&Mgmt', 'Others'], dtype=object)"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['degree\_t'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"id": "dfa67979",

"metadata": {},

"outputs": [],

"source": [

"dataset['degree\_t']=dataset['degree\_t'].map({'Sci&Tech':1,'Comm&Mgmt':2, 'Others':3 })"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"id": "2fdeac42",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>No</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>1</td>\n",

" <td>Yes</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>2</td>\n",

" <td>No</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>1</td>\n",

" <td>No</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>73.30</td>\n",

" <td>2</td>\n",

" <td>No</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 1 1 58.00 1 No \n",

"1 M 79.33 2 78.33 1 2 77.48 1 Yes \n",

"2 M 65.00 2 68.00 2 3 64.00 2 No \n",

"3 M 56.00 2 52.00 2 2 52.00 1 No \n",

"4 M 85.80 2 73.60 2 1 73.30 2 No \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 Mkt&HR 58.80 Placed \n",

"1 86.5 Mkt&Fin 66.28 Placed \n",

"2 75.0 Mkt&Fin 57.80 Placed \n",

"3 66.0 Mkt&HR 59.43 Not Placed \n",

"4 96.8 Mkt&Fin 55.50 Placed "

]

},

"execution\_count": 23,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"id": "668f7194",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['No', 'Yes'], dtype=object)"

]

},

"execution\_count": 24,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['workex'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"id": "cf95ef4a",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

<th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>73.30</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 1 1 58.00 1 0 \n",

"1 M 79.33 2 78.33 1 2 77.48 1 1 \n",

"2 M 65.00 2 68.00 2 3 64.00 2 0 \n",

"3 M 56.00 2 52.00 2 2 52.00 1 0 \n",

"4 M 85.80 2 73.60 2 1 73.30 2 0 \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 Mkt&HR 58.80 Placed \n",

"1 86.5 Mkt&Fin 66.28 Placed \n",

"2 75.0 Mkt&Fin 57.80 Placed \n",

"3 66.0 Mkt&HR 59.43 Not Placed \n",

"4 96.8 Mkt&Fin 55.50 Placed "

]

},

"execution\_count": 25,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['workex']=dataset['workex'].map({'No':0,'Yes':1})\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"id": "46681b04",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Mkt&HR', 'Mkt&Fin'], dtype=object)"

]

},

"execution\_count": 26,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['specialisation'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"id": "eff18306",

"metadata": {},

"outputs": [

{

"data": {

<th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>55.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>86.5</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>75.0</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>66.0</td>\n",

" <td>Mkt&amp;HR</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>73.30</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>96.8</td>\n",

" <td>Mkt&amp;Fin</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 1 1 58.00 1 0 \n",

"1 M 79.33 2 78.33 1 2 77.48 1 1 \n",

"2 M 65.00 2 68.00 2 3 64.00 2 0 \n",

"3 M 56.00 2 52.00 2 2 52.00 1 0 \n",

"4 M 85.80 2 73.60 2 1 73.30 2 0 \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 Mkt&HR 58.80 Placed \n",

"1 86.5 Mkt&Fin 66.28 Placed \n",

"2 75.0 Mkt&Fin 57.80 Placed \n",

"3 66.0 Mkt&HR 59.43 Not Placed \n",

"4 96.8 Mkt&Fin 55.50 Placed "

]

},

"execution\_count": 25,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['workex']=dataset['workex'].map({'No':0,'Yes':1})\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"id": "46681b04",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['Mkt&HR', 'Mkt&Fin'], dtype=object)"

]

},

"execution\_count": 26,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['specialisation'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"id": "eff18306",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>M</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>55.0</td>\n",

" <td>1</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

<td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>M</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>86.5</td>\n",

" <td>2</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>M</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>75.0</td>\n",

" <td>2</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>M</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>66.0</td>\n",

" <td>1</td>\n",

" <td>59.43</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>M</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>73.30</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>96.8</td>\n",

" <td>2</td>\n",

" <td>55.50</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 M 67.00 1 91.00 1 1 58.00 1 0 \n",

"1 M 79.33 2 78.33 1 2 77.48 1 1 \n",

"2 M 65.00 2 68.00 2 3 64.00 2 0 \n",

"3 M 56.00 2 52.00 2 2 52.00 1 0 \n",

"4 M 85.80 2 73.60 2 1 73.30 2 0 \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 1 58.80 Placed \n",

"1 86.5 2 66.28 Placed \n",

"2 75.0 2 57.80 Placed \n",

"3 66.0 1 59.43 Not Placed \n",

"4 96.8 2 55.50 Placed "

]

},

"execution\_count": 27,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['specialisation']=dataset['specialisation'].map({'Mkt&HR':1,'Mkt&Fin':2})\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"id": "c789c45b",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array(['M', 'F'], dtype=object)"

]

},

"execution\_count": 28,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['gender'].unique()"

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"id": "79ded83a",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" <th>status</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>55.0</td>\n",

" <td>1</td>\n",

" <td>58.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>1</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>86.5</td>\n",

" <td>2</td>\n",

" <td>66.28</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>1</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>75.0</td>\n",

" <td>2</td>\n",

" <td>57.80</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>1</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 1 67.00 1 91.00 1 1 58.00 1 0 \n",

"1 1 79.33 2 78.33 1 2 77.48 1 1 \n",

"2 1 65.00 2 68.00 2 3 64.00 2 0 \n",

"3 1 56.00 2 52.00 2 2 52.00 1 0 \n",

"4 1 85.80 2 73.60 2 1 73.30 2 0 \n",

"\n",

" etest\_p specialisation mba\_p status \n",

"0 55.0 1 58.80 Placed \n",

"1 86.5 2 66.28 Placed \n",

"2 75.0 2 57.80 Placed \n",

"3 66.0 1 59.43 Not Placed \n",

"4 96.8 2 55.50 Placed "

]

},

"execution\_count": 29,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset['gender']=dataset['gender'].map({'M':1, 'F':2})\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"id": "f3f6b90e",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>gender</th>\n",

" <th>ssc\_p</th>\n",

" <th>ssc\_b</th>\n",

" <th>hsc\_p</th>\n",

" <th>hsc\_b</th>\n",

" <th>hsc\_s</th>\n",

" <th>degree\_p</th>\n",

" <th>degree\_t</th>\n",

" <th>workex</th>\n",

" <th>etest\_p</th>\n",

" <th>specialisation</th>\n",

" <th>mba\_p</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>67.00</td>\n",

" <td>1</td>\n",

" <td>91.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>58.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>55.0</td>\n",

" <td>1</td>\n",

" <td>58.80</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>1</td>\n",

" <td>79.33</td>\n",

" <td>2</td>\n",

" <td>78.33</td>\n",

" <td>1</td>\n",

" <td>2</td>\n",

" <td>77.48</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>86.5</td>\n",

" <td>2</td>\n",

" <td>66.28</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>1</td>\n",

" <td>65.00</td>\n",

" <td>2</td>\n",

" <td>68.00</td>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" <td>64.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>75.0</td>\n",

" <td>2</td>\n",

" <td>57.80</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>1</td>\n",

" <td>56.00</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>2</td>\n",

" <td>2</td>\n",

" <td>52.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>66.0</td>\n",

" <td>1</td>\n",

" <td>59.43</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>1</td>\n",

" <td>85.80</td>\n",

" <td>2</td>\n",

" <td>73.60</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>73.30</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>96.8</td>\n",

" <td>2</td>\n",

" <td>55.50</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex \\\n",

"0 1 67.00 1 91.00 1 1 58.00 1 0 \n",

"1 1 79.33 2 78.33 1 2 77.48 1 1 \n",

"2 1 65.00 2 68.00 2 3 64.00 2 0 \n",

"3 1 56.00 2 52.00 2 2 52.00 1 0 \n",

"4 1 85.80 2 73.60 2 1 73.30 2 0 \n",

"\n",

" etest\_p specialisation mba\_p \n",

"0 55.0 1 58.80 \n",

"1 86.5 2 66.28 \n",

"2 75.0 2 57.80 \n",

"3 66.0 1 59.43 \n",

"4 96.8 2 55.50 "

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"#Scaling\n",

"\n",

"from sklearn.preprocessing import StandardScaler\n",

"\n",

"X=dataset.drop(['status'],axis=1)\n",

"X.head()\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"id": "30346ad7",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"0 Placed\n",

"1 Placed\n",

"2 Placed\n",

"3 Not Placed\n",

"4 Placed\n",

" ... \n",

"210 Placed\n",

"211 Placed\n",

"212 Placed\n",

"213 Placed\n",

"214 Not Placed\n",

"Name: status, Length: 215, dtype: object\n",

"['Placed' 'Placed' 'Placed' 'Not Placed' 'Placed' 'Not Placed'\n",

" 'Not Placed' 'Placed'

" <th>11</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>12</th>\n",

" <td>Not Placed</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>13</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14</th>\n",

" <td>Not Placed</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>15</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>16</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>17</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>18</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>19</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>20</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>21</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>22</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>23</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>24</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>26</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

<tr>\n",

" <th>17</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>18</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>19</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>20</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>21</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>22</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>23</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>24</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>26</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

<tr>\n",

" <th>17</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>18</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>19</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>20</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>21</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>22</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>23</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>24</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>26</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>27</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>28</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>29</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>30</th>\n",

" <td>Not Placed</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>31</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>32</th>\n",

" <td>Not Placed</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>33</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>34</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>35</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>36</th>\n",

" <td>Not Placed</td>\n",

" <td>Not Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>37</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>38</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>39</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>40</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>41</th>\n",

" <td>Not Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" <tr>\n",

" <th>42</th>\n",

" <td>Placed</td>\n",

" <td>Placed</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Actual Predicted\n",

"0 Placed Placed\n",

"1 Placed Placed\n",

"2 Not Placed Not Placed\n",

"3 Placed Placed\n",

"4 Not Placed Not Placed\n",

"5 Placed Placed\n",

"6 Placed Not Placed\n",

"7 Placed Placed\n",

"8 Placed Placed\n",

"9 Not Placed Placed\n",

"10 Placed Placed\n",

"11 Placed Placed\n",

"12 Not Placed Not Placed\n",

"13 Placed Placed\n",

"14 Not Placed Not Placed\n",

]

},

"execution\_count": 54,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_pred=sv.predict(X\_test)\n",

"df = pd.DataFrame({'Actual':y\_test, 'Predicted': y\_pred})\n",

"df"

]

},

{

"cell\_type": "code",

"execution\_count": 55,

"id": "64ec7508",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"0.8372093023255814\n",

"[[ 9 6]\n",

" [ 1 27]]\n"

]

}

],

"source": [

"print(accuracy\_score(y\_test,y\_pred))\n",

"print(confusion\_matrix(y\_test,y\_pred))"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"id": "7314afcc",

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.8.8"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| KNN | K Nearest Neighbour |
| SVM | Support Vector Machine |
| SVC | Support Vectoe Classifier |

**CONCLUSION**

Student Placement Predictor is a system which predicts student placement status using machine learning techniques. This system is beneficial for institutions to predict student’s campus placement and placement officers can work on identifying the weakness of each student. They can also suggest improvements so that the student can overcome the weakness and supply to the best of their abilities. Algorithms like Logistic Regression and SVC will give maximum accuracy to the prediction.

The future enhancements of the project is to focus on to add some more parameters to predict more efficient placement status. We can also enhance the project by predicting some solutions or suggestions for the output generated by system.

**BIBILIOGRAPHY**

1.) <https://www.kaggle.com/>

2.) <https://www.wikipedia.org/>

3.) <https://www.geeksforgeeks.org/>