**Practical No: 01**

**Aim: Implementation Of Prolog**

**Code:**

precious(gold)precious(silver).

precious(pearl).

**Output:**

****

**Code:**

female (mary).

father (surya, ramesh).

likes (surya, food).

likes (surya, car).

likes (surya, bike).

likes (surya, book).

likes (shreya, chocolate).

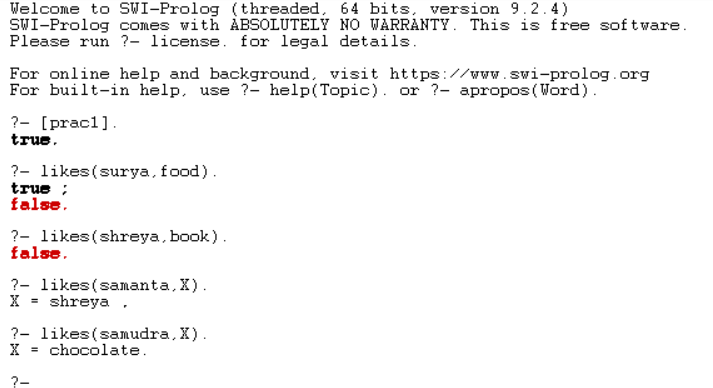
likes (sudha, food).

likes (surya, coffee).

likes (samudra, chocolate).

likes (samanta, X): likes (X, chocolate).

**Output:**

****

**Practical No: 02**

**Aim: Implementation of Water jug problem using Prolog**

**Code:**

water\_jug (X, Y): - X>4, Y<3, write ('4L jug overflow'), nl.

water\_jug (X, Y): - X>4, Y>3, write ('3L jug overflow'), nl.

water\_jug (X, Y): - X>4, Y>3, write ('Both jug overflow'), nl.

water\_jug (X, Y): - (X=: =0, Y=: =0, nl, write ('4L:0 & 3L:3 (Action: Fill 3L jug.)'), YY is 3,

water\_jug (X, YY));

(X=: =0, Y=: =0, nl, write ('4L:4 & 3L:0 (Action: Fill 4L jug.)'), XX is 4, water\_jug (XX, Y));

(X=: =2, Y=: =0, nl, write ('4L:2 & 3L:0 (Action: Goal Stage reached...)'));

(X=: =4, Y=: =0, nl, write ('4L:1 & 3L:3 (Action: Pour water from 4L to 3L jug.)'), XX is X-3,

YY is 3, water\_jug (XX, YY));

(X=: =0, Y=: =3, nl, write ('4L:3 & 3L:0 (Action: Pour water from 3L to 4L jug.)'), XX is 3, YY

is 0, water\_jug (XX, YY));

(X=: =1, Y=: =3, nl, write ('4L:1 & 3L:0 (Action: Empty 3L jug.)'), YY is 0, water\_jug (X, YY));

(X=: =3, Y=: =0, nl, write ('4L:3 & 3L:3 (Action: Fill 3L jug.)'), YY is 3, water\_jug (X, YY));

(X=: =3, Y=: =3, nl, write ('4L:4 & 3L:2 (Action: Pour water from 3L jug to 4L jug until 4L jug is full.)'), XX is X+1, YY is Y-1, water\_jug (XX, YY));

(X=: =1, Y=: =0, nl, write ('4L:0 & 3L:1 (Action: Pour water from 4L jug to 3L jug.)'), XX is Y,

YY is X, water\_jug (XX, YY));

(X=:=0,Y=:=1, nl, write ('4L:4 & 3L:1 (Action: Pour water from 4L.)'), XX is 4,

water\_jug (XX, Y));

(X=: =4, Y=: =1, nl, write ('4L:2 & 3L:3 (Action: Pour water from 4L jug to 3L jug until 3L jug

is full.)'), XX is X-2, YY is Y+2, water\_jug (XX, YY));

(X=: =2, Y=: =3, nl, write ('4L:2 & 3L:0 (Action: Empty 3L jug.)'), YY is 0,

water\_jug (X, YY));

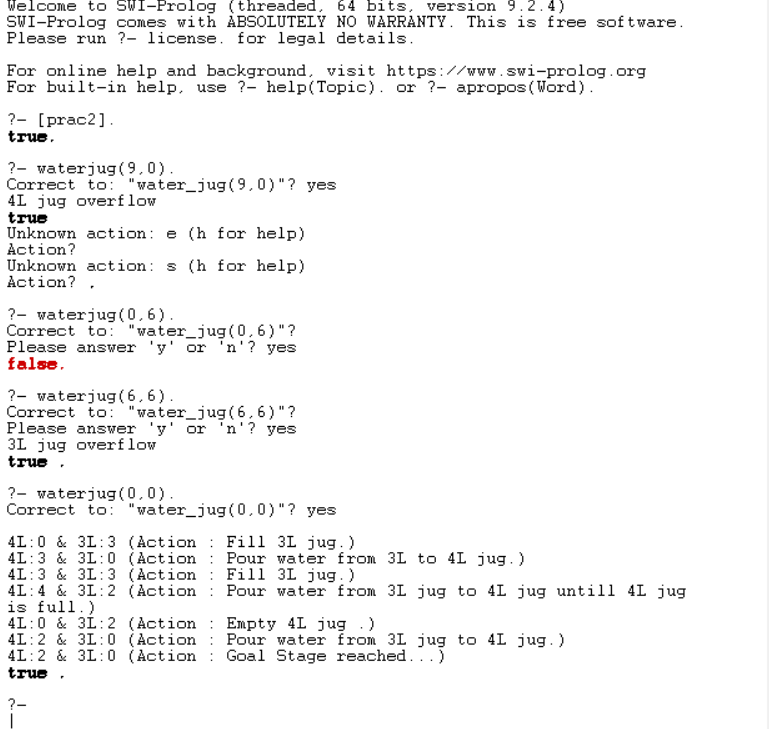
(X=: = 4, Y=: =2, nl, write ('4L:0 & 3L:2 (Action: Empty 4L jug.)'), XX is 0,

water\_jug (XX, Y));

(X=: =0, Y=: =2, nl, write ('4L:2 & 3L:0 (Action: Pour water from 3L jug to 4L jug.)'), XX is Y,

YY is X, water\_jug (XX, YY)).

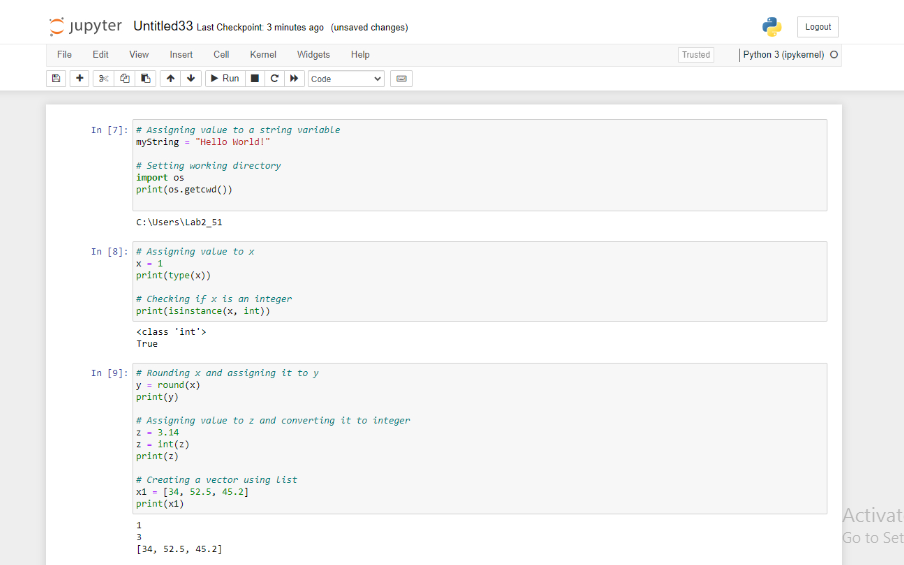
**Output:**

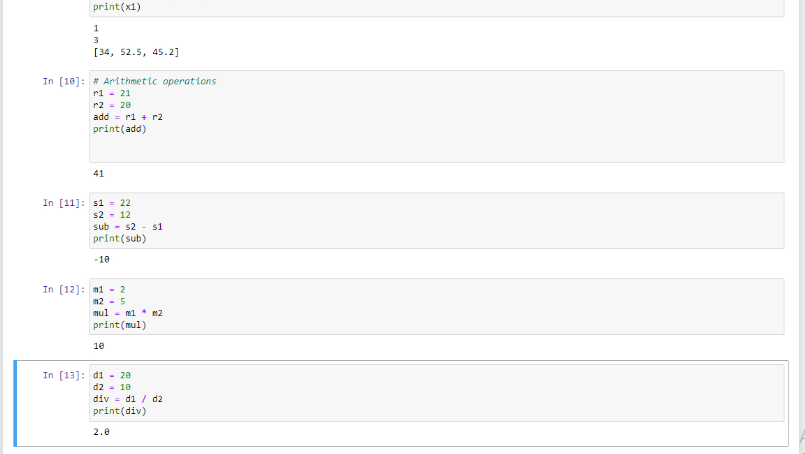
****

**Practical No:3**

**Aim: Introduction to Python Programming**

**Output:**

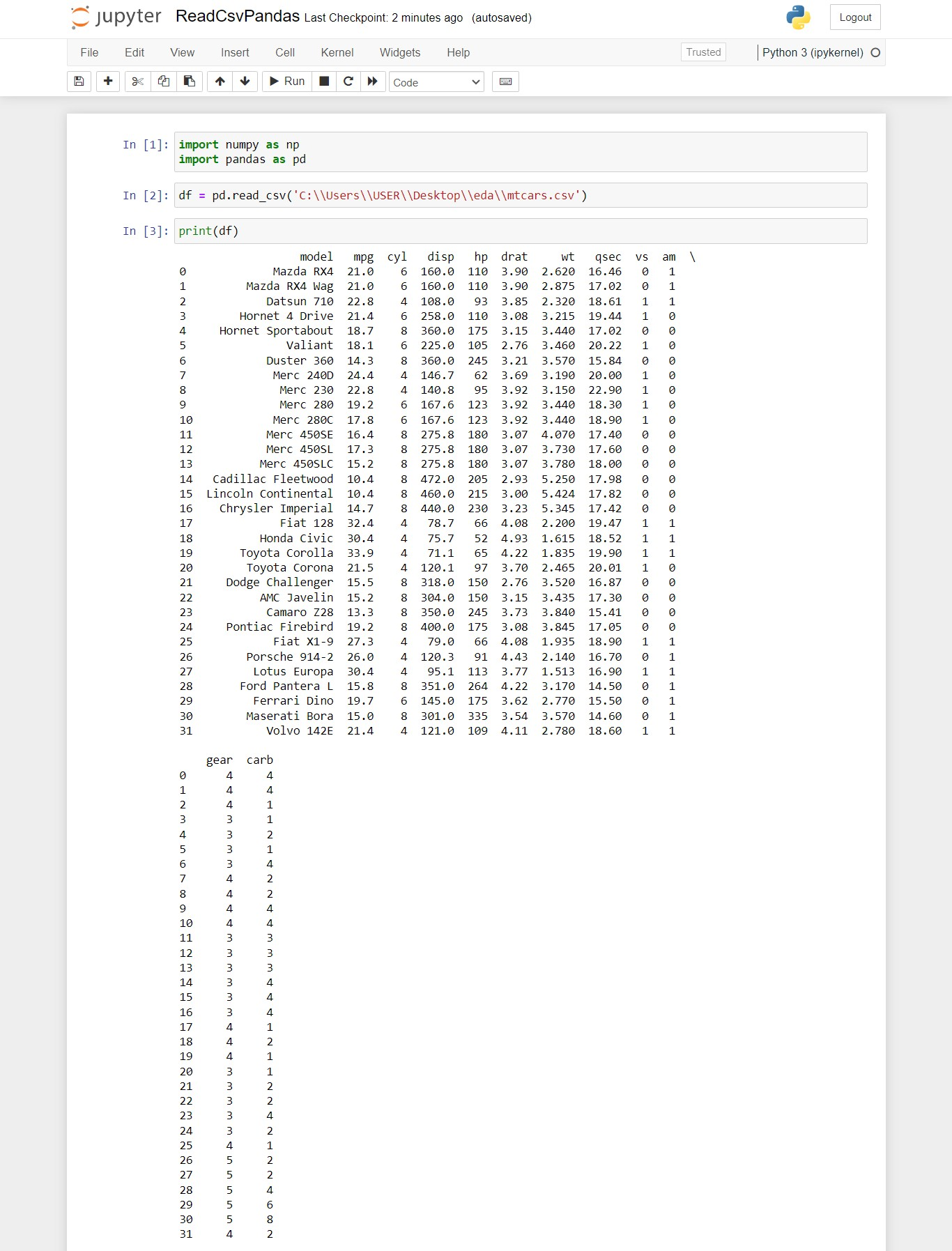


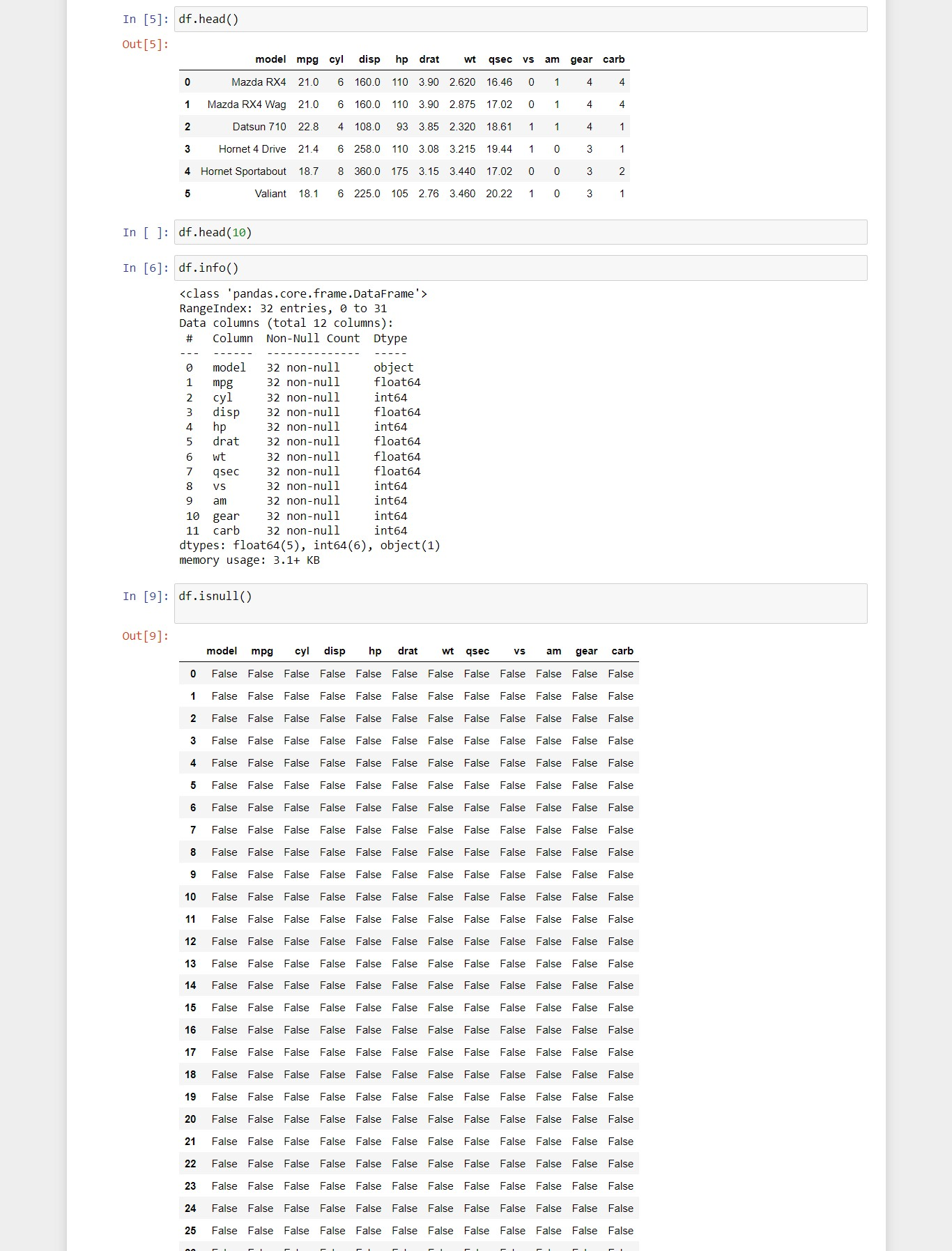


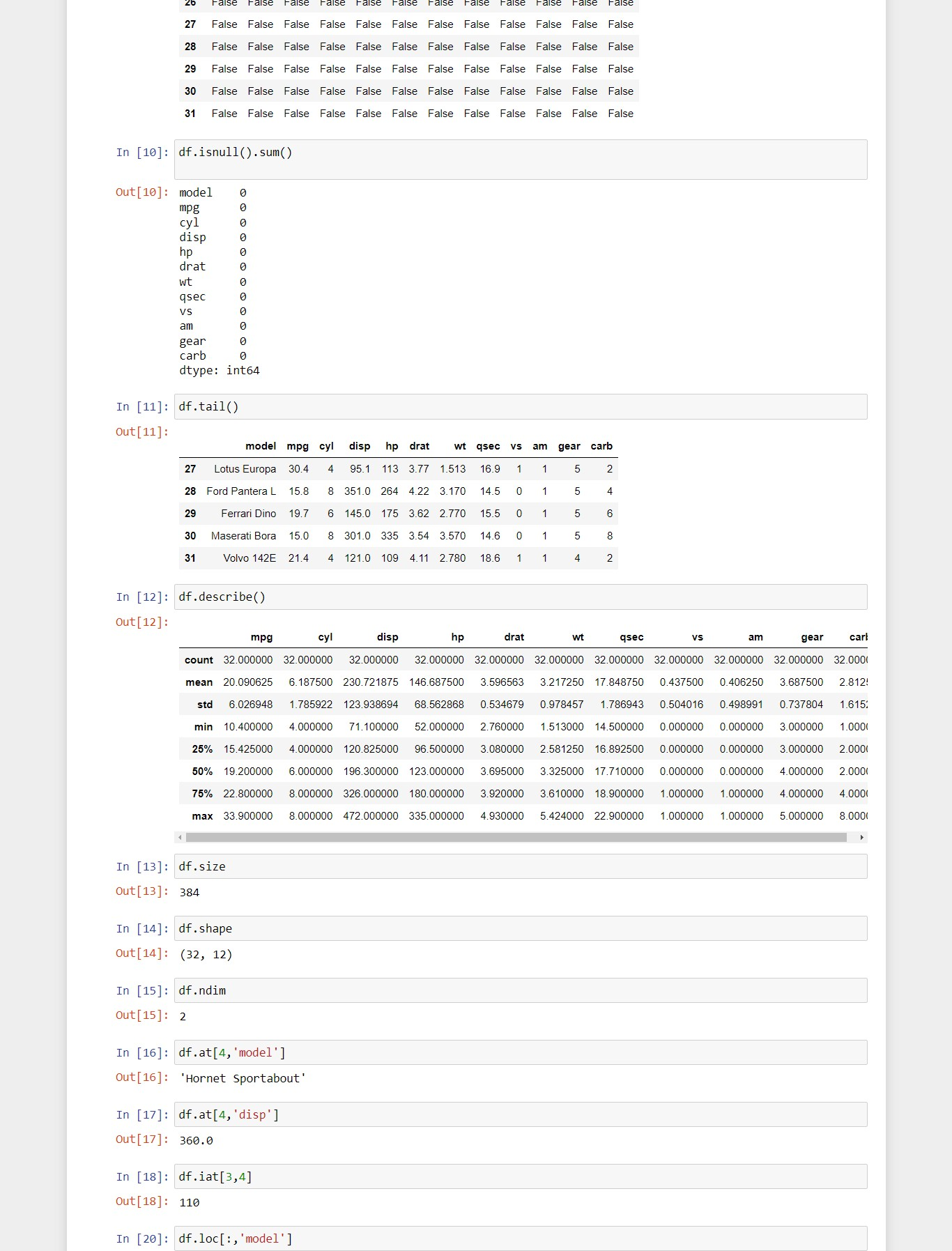
**Practical No:4**

**Aim: Introduction to Python Libraries**

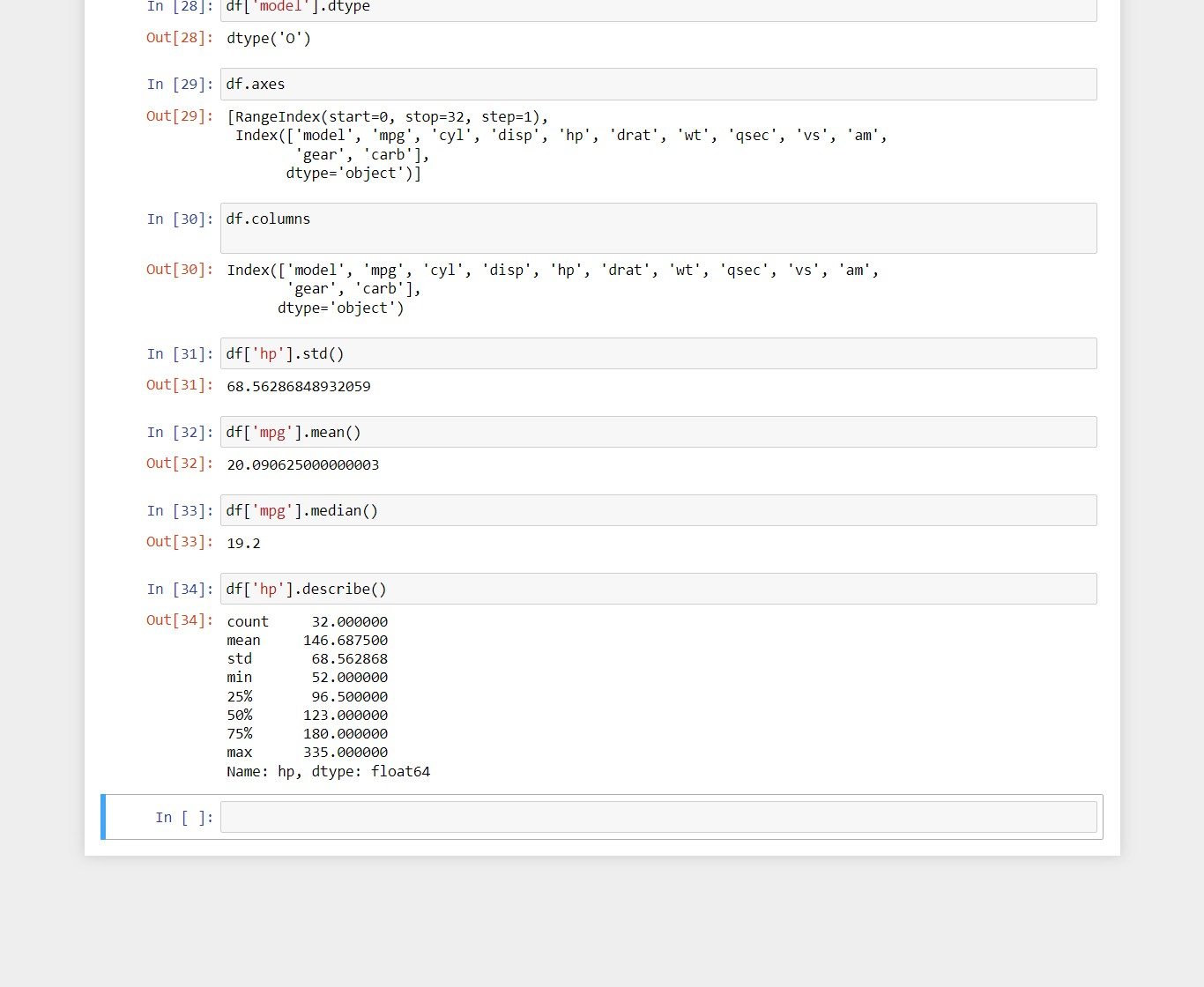
**Output:**







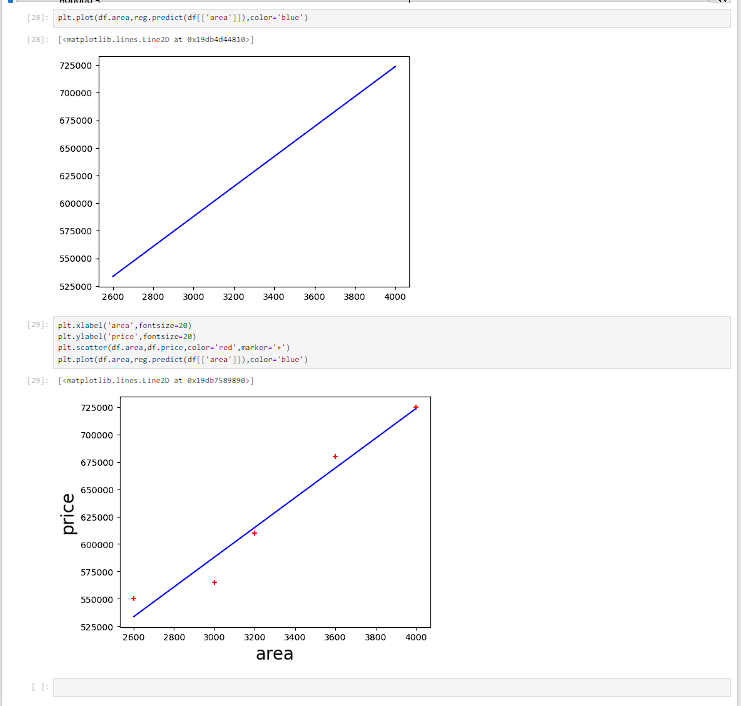
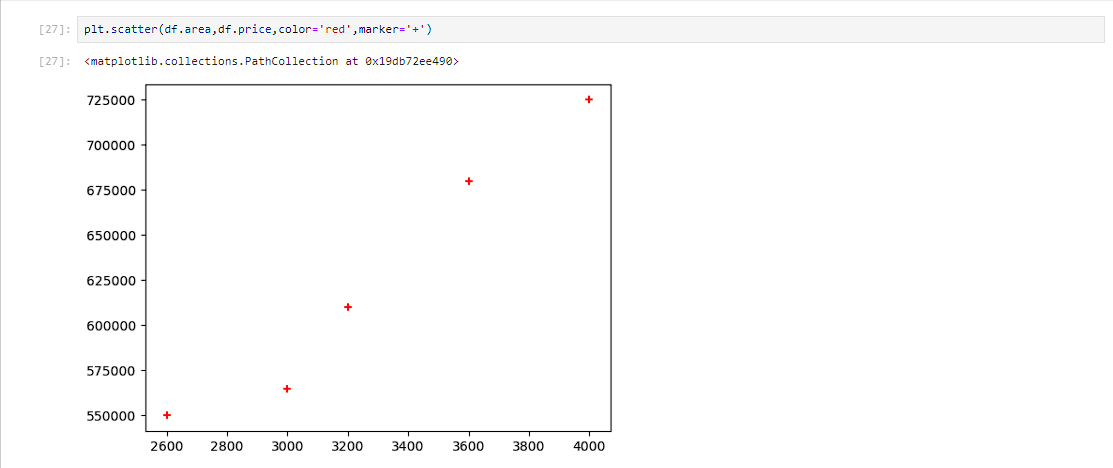
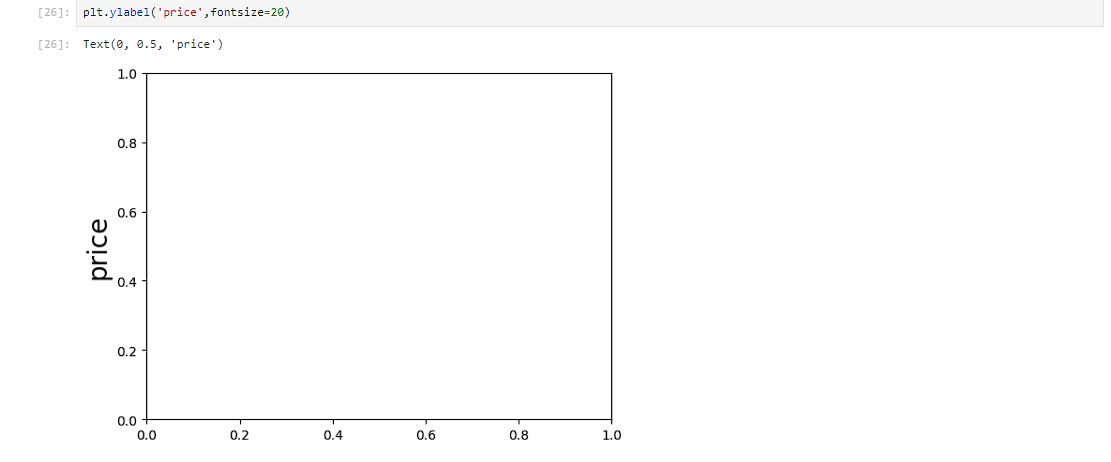
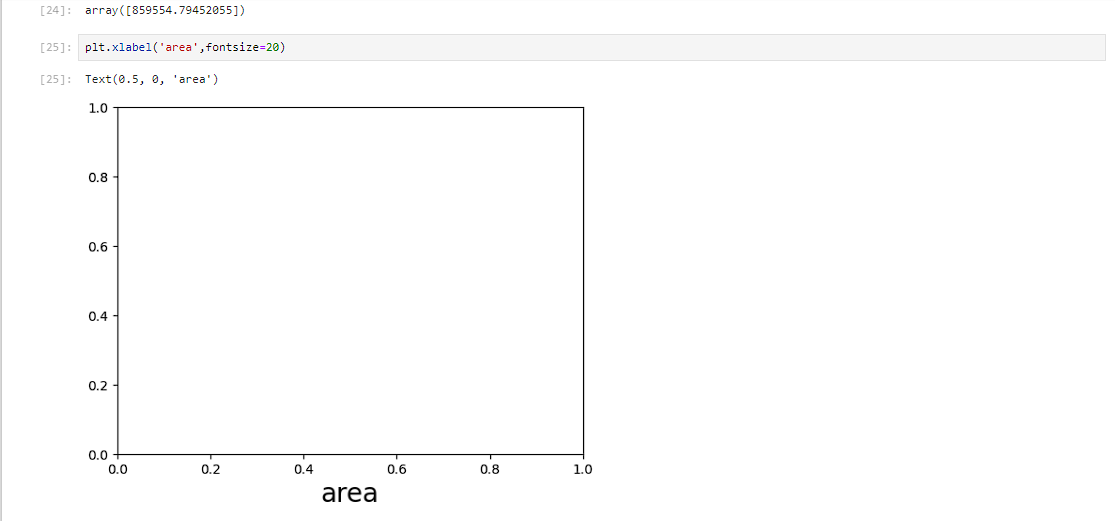
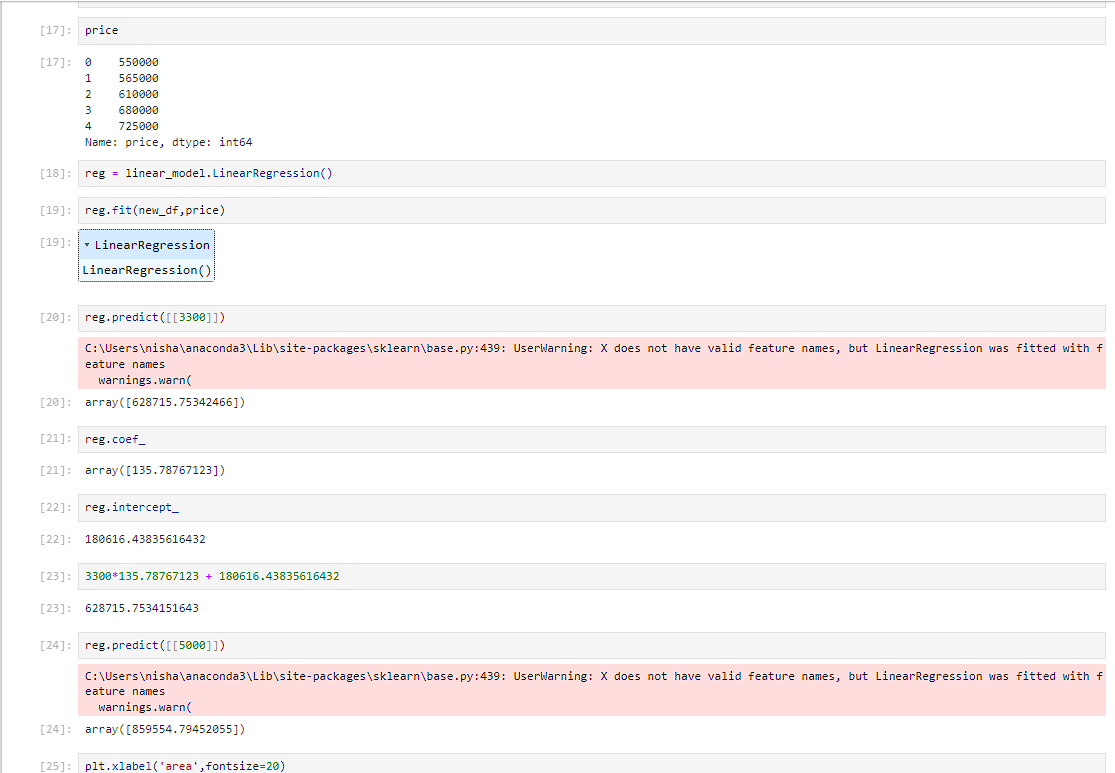
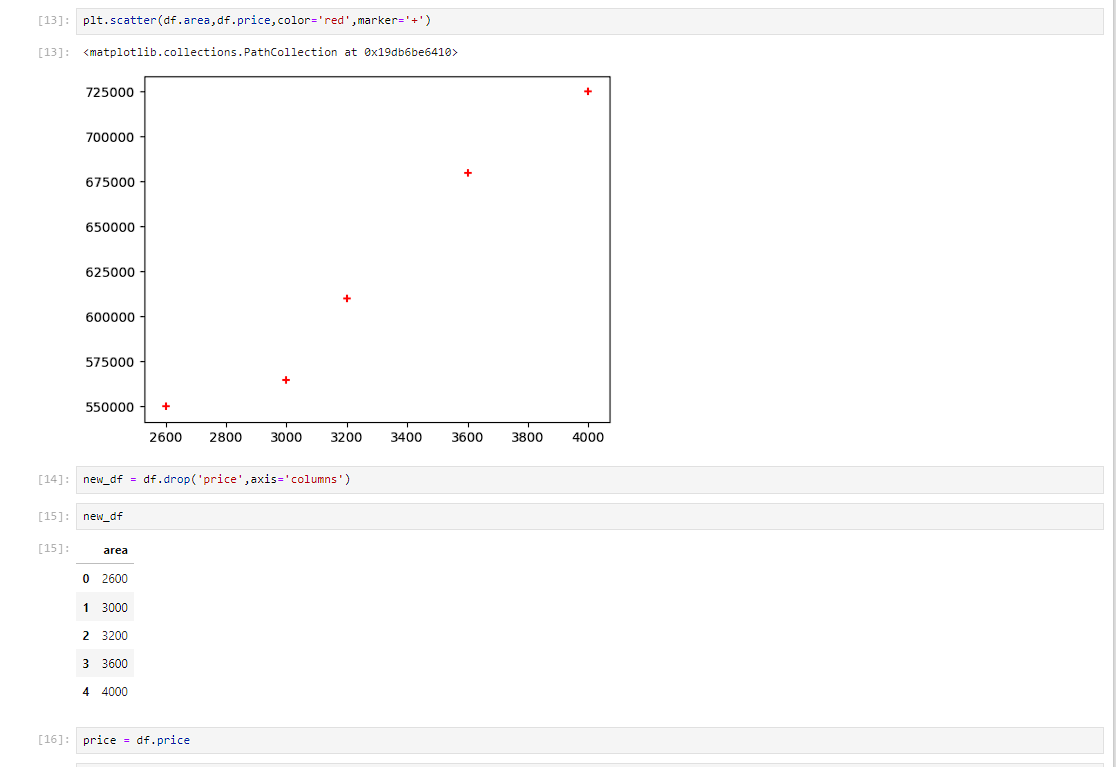
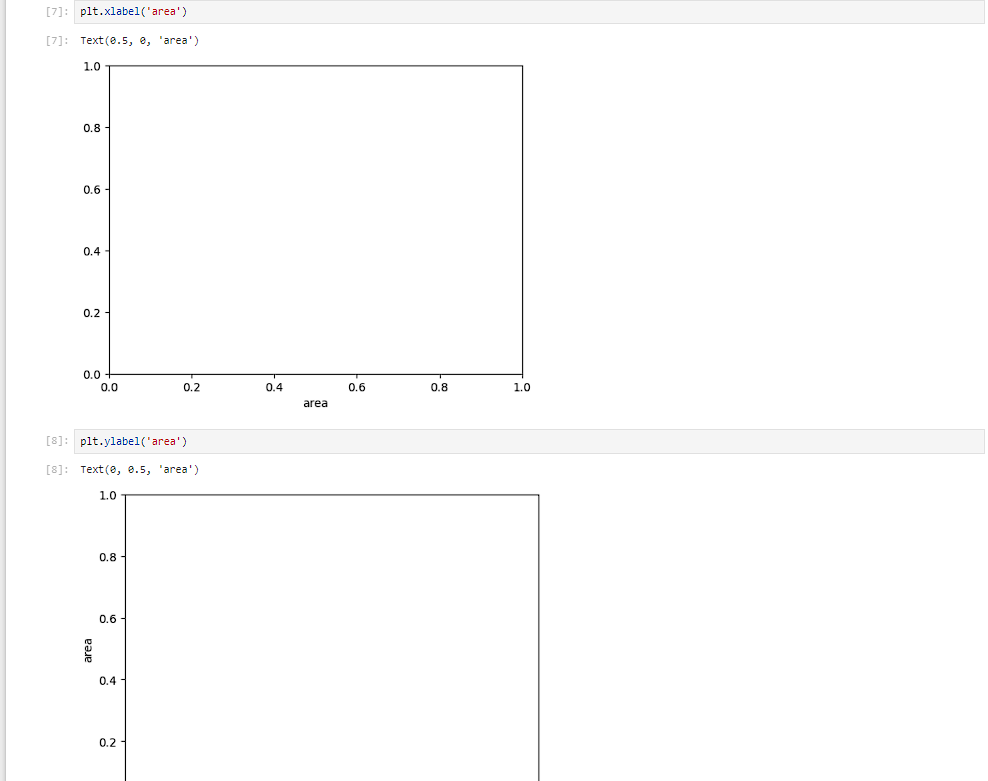




**Practical No:5**

**Aim: Program to Implement Linear Regression**

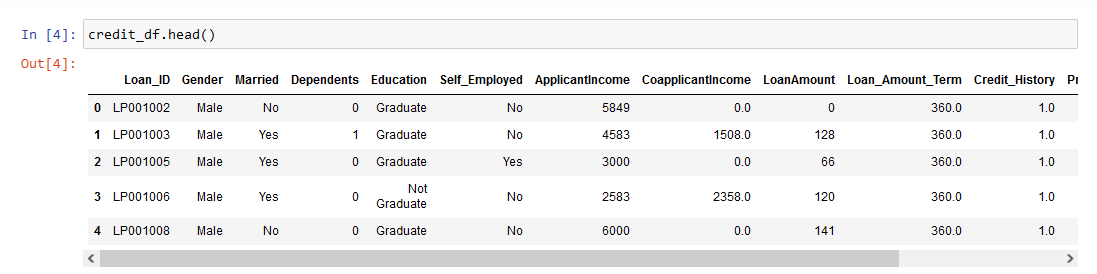
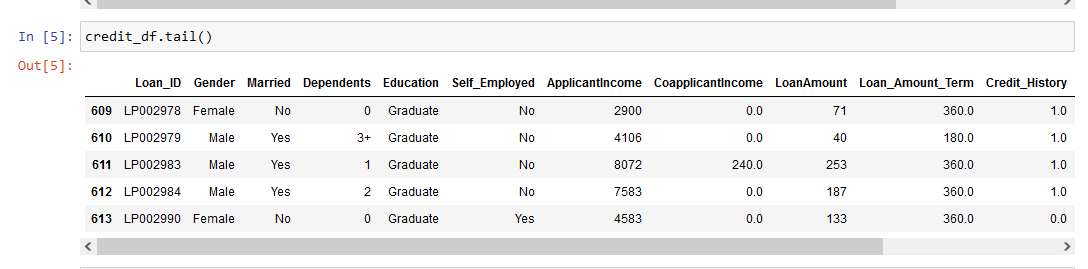
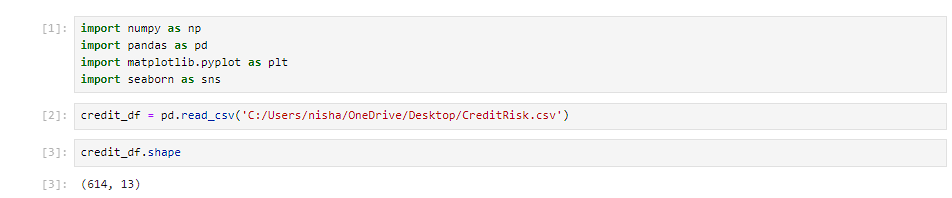
**Output:**

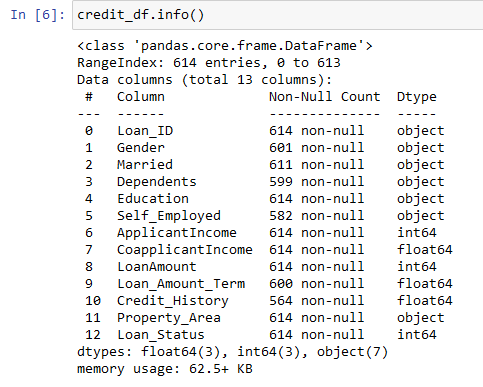


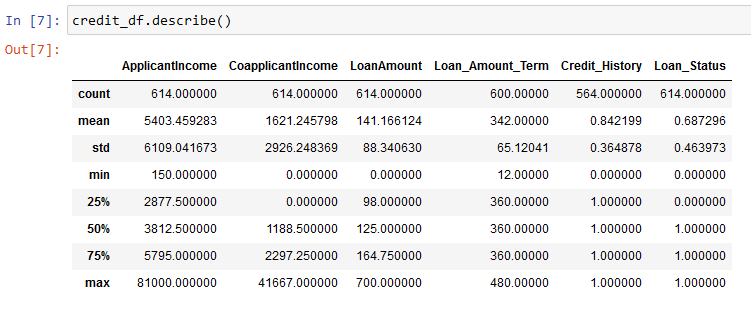
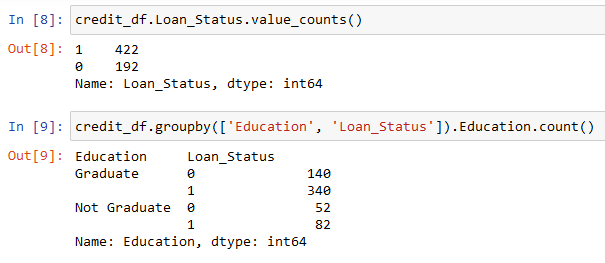
**Practical No:6**

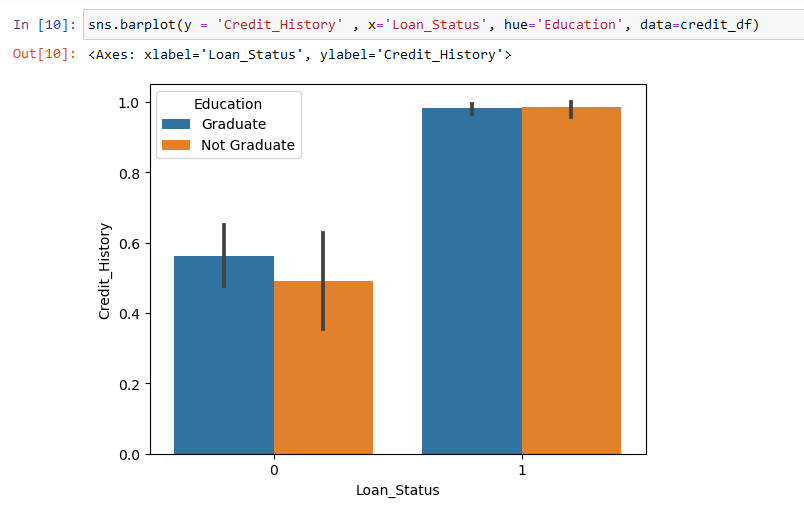
**Aim: Program to Implement Logistic Regression**

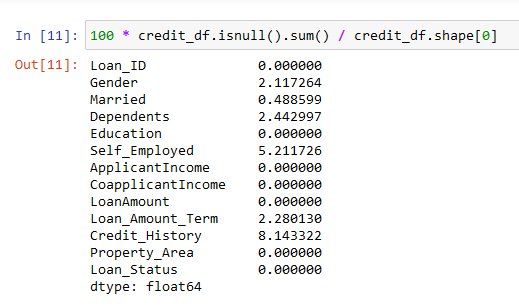
**Output:**

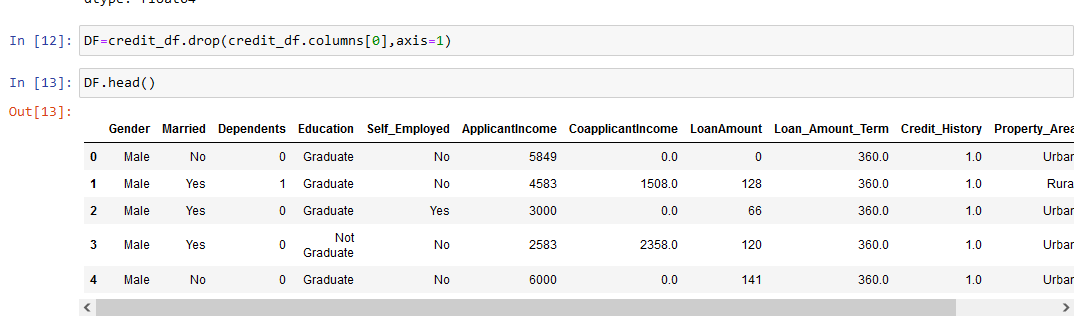
****

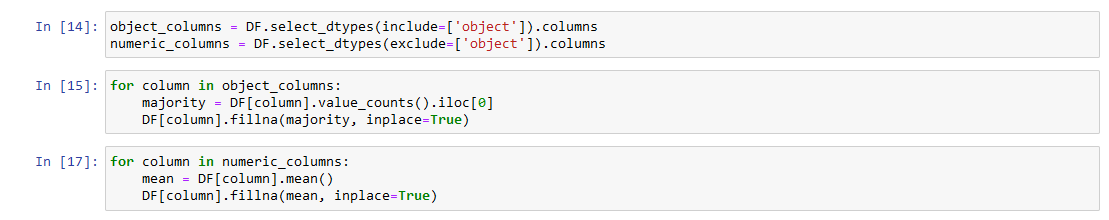


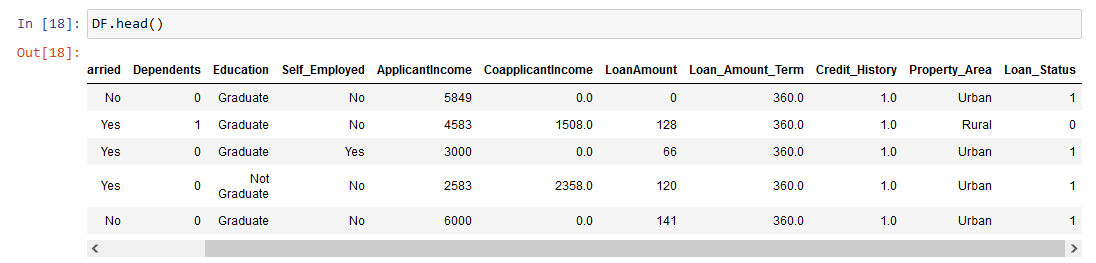




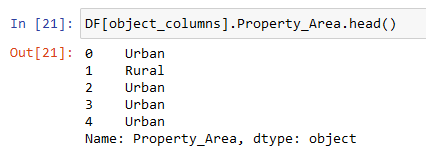


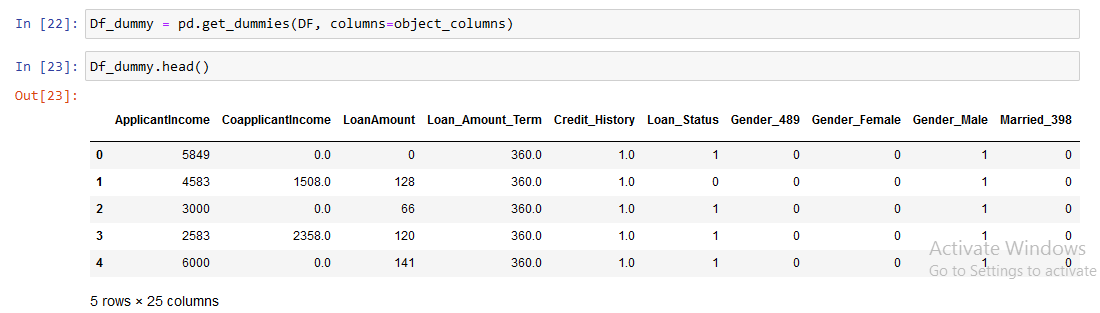


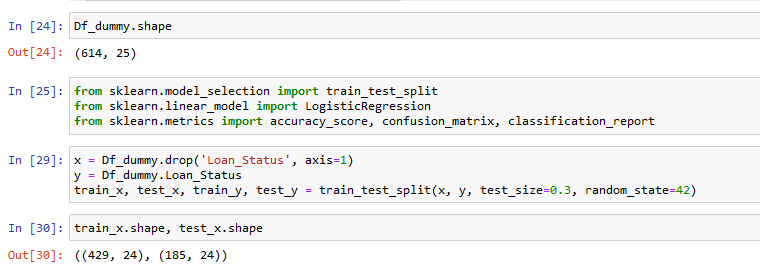


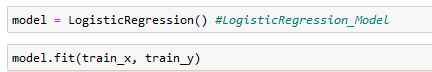


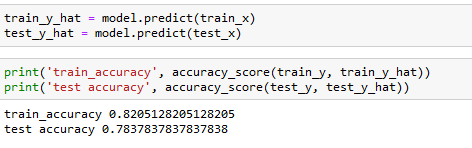


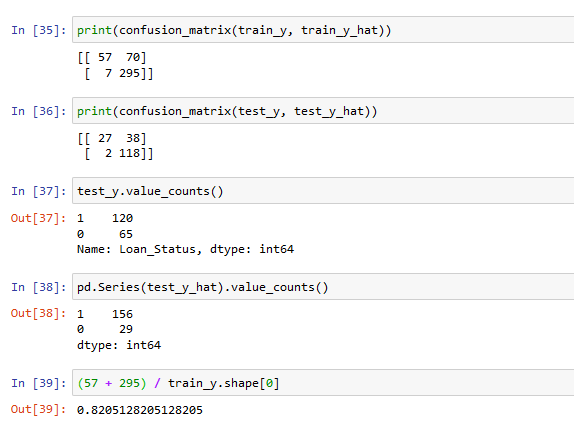


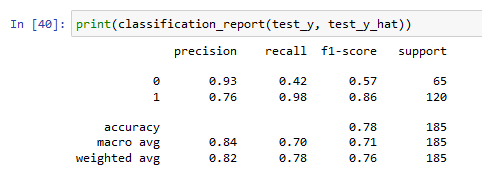


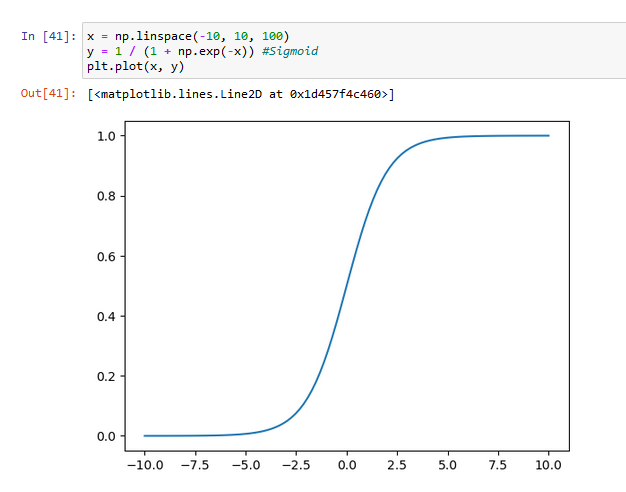




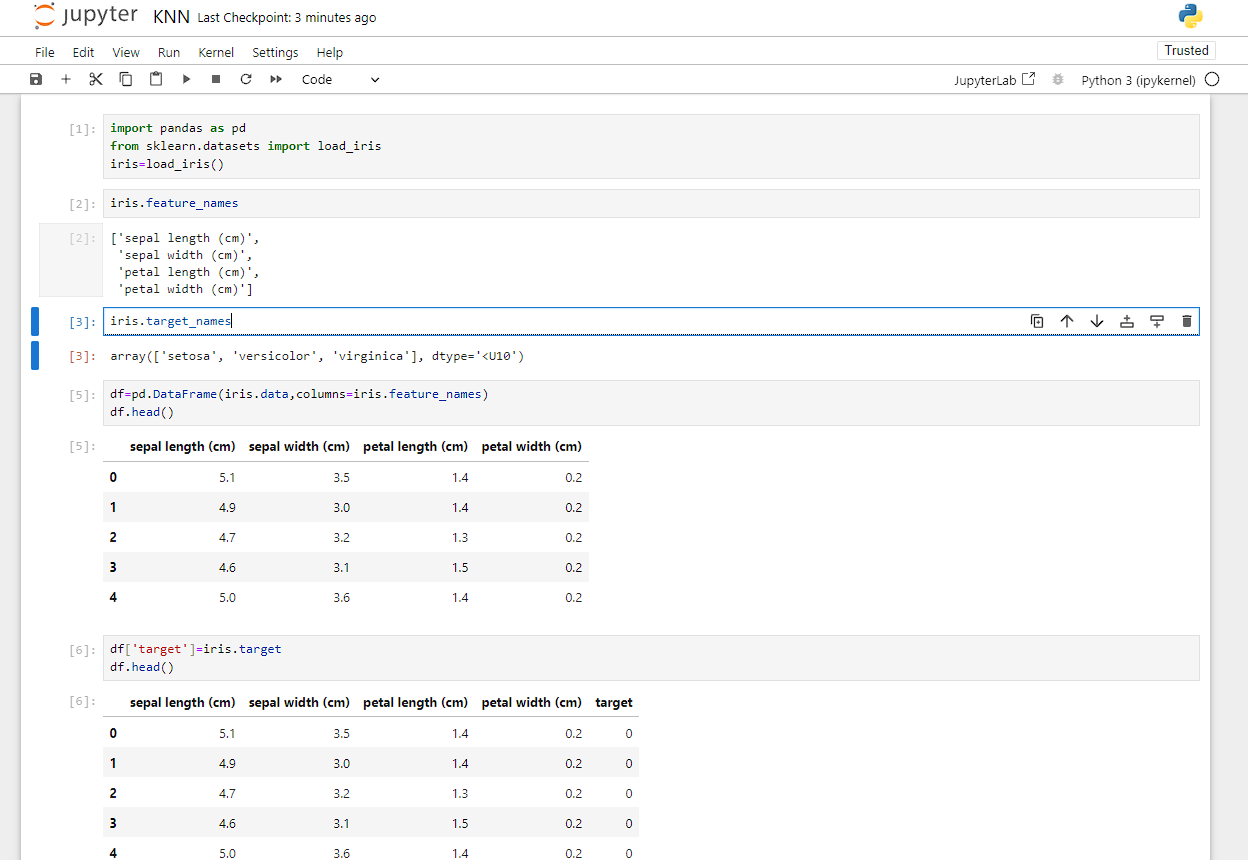
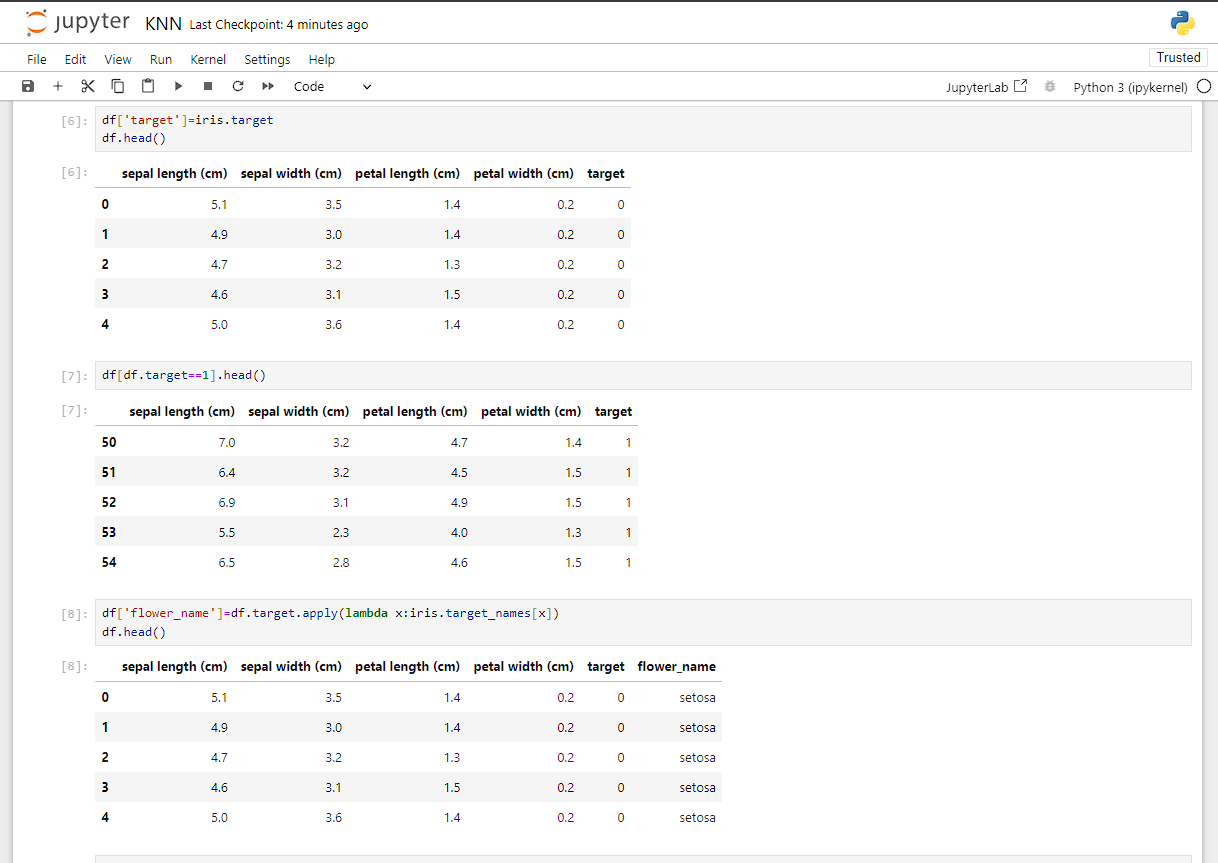


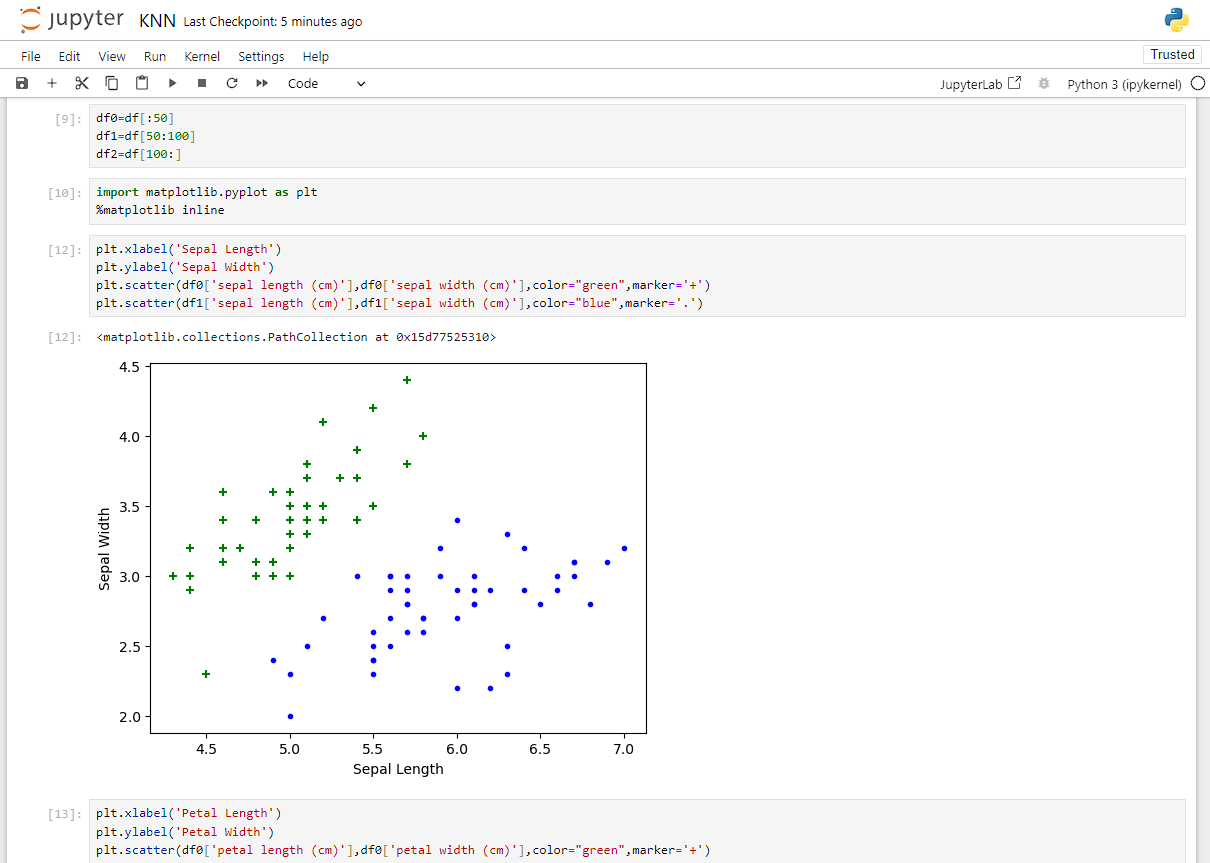
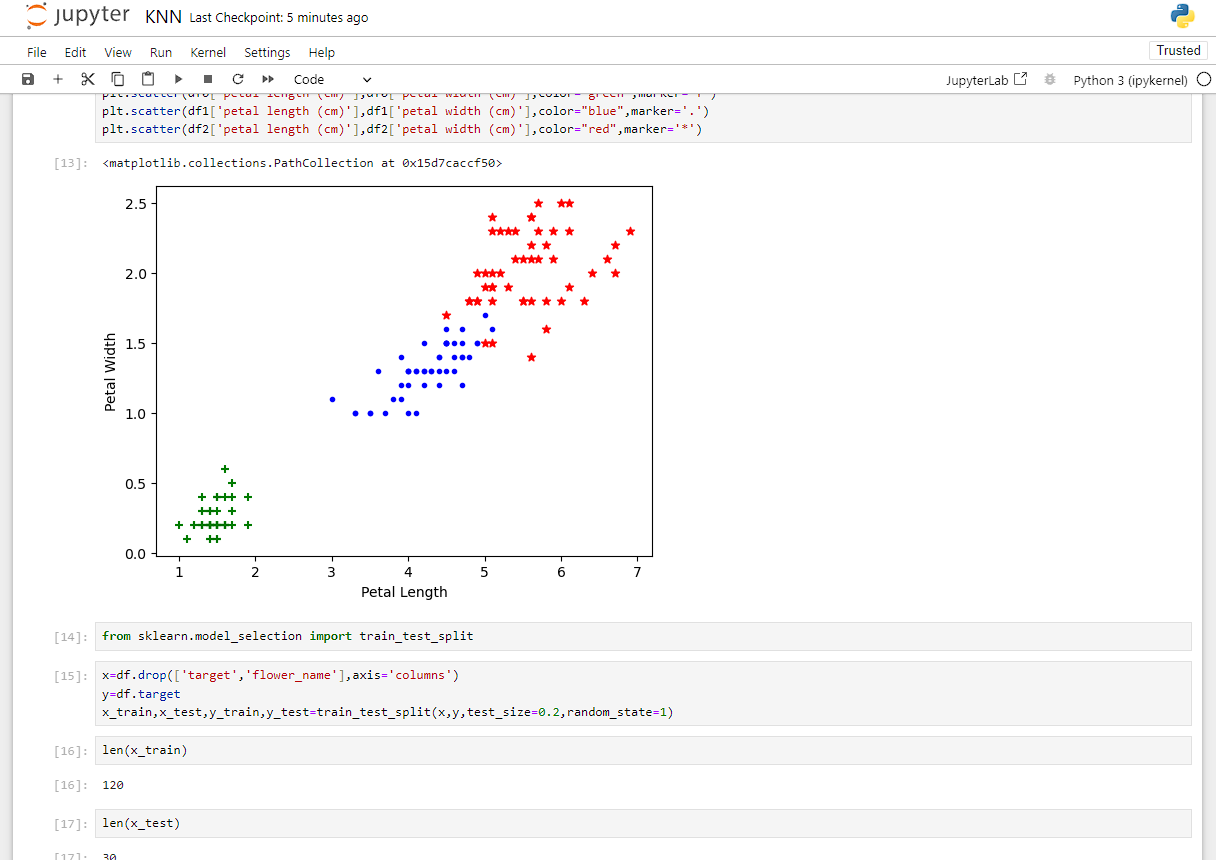


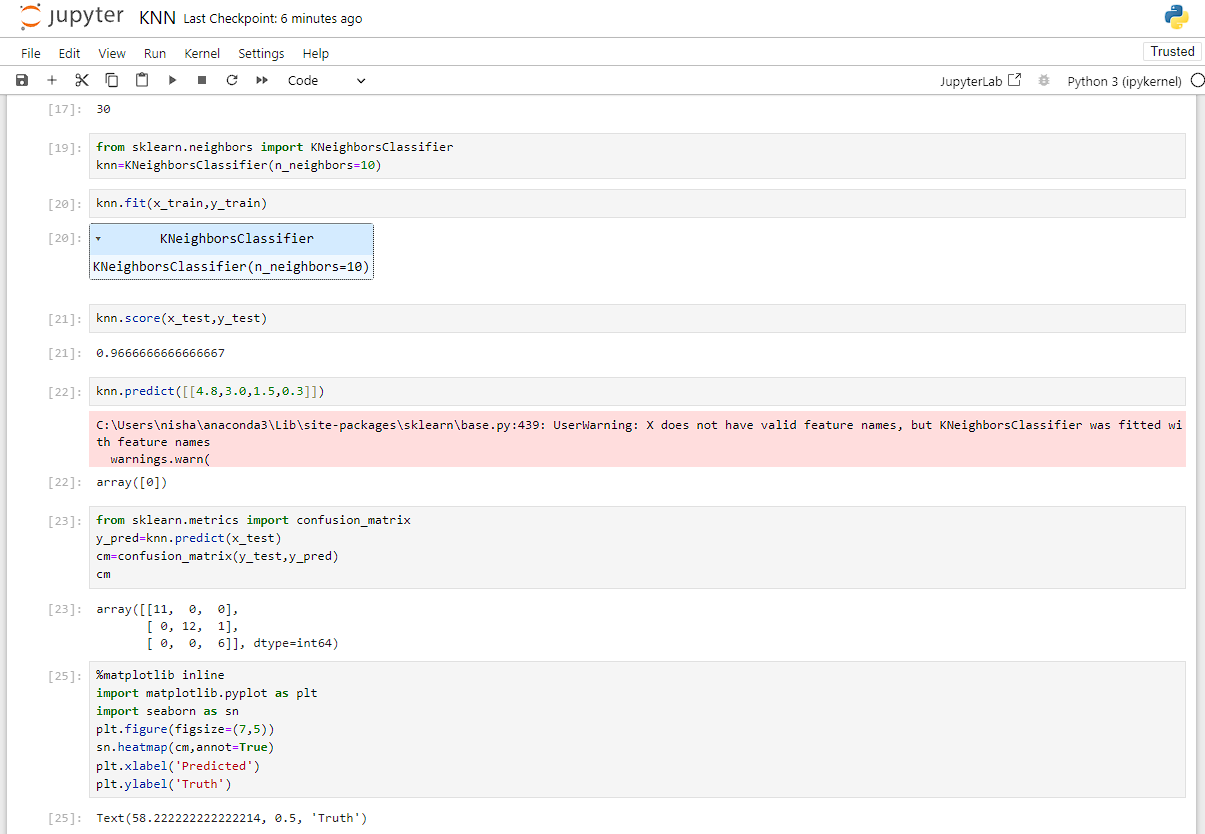
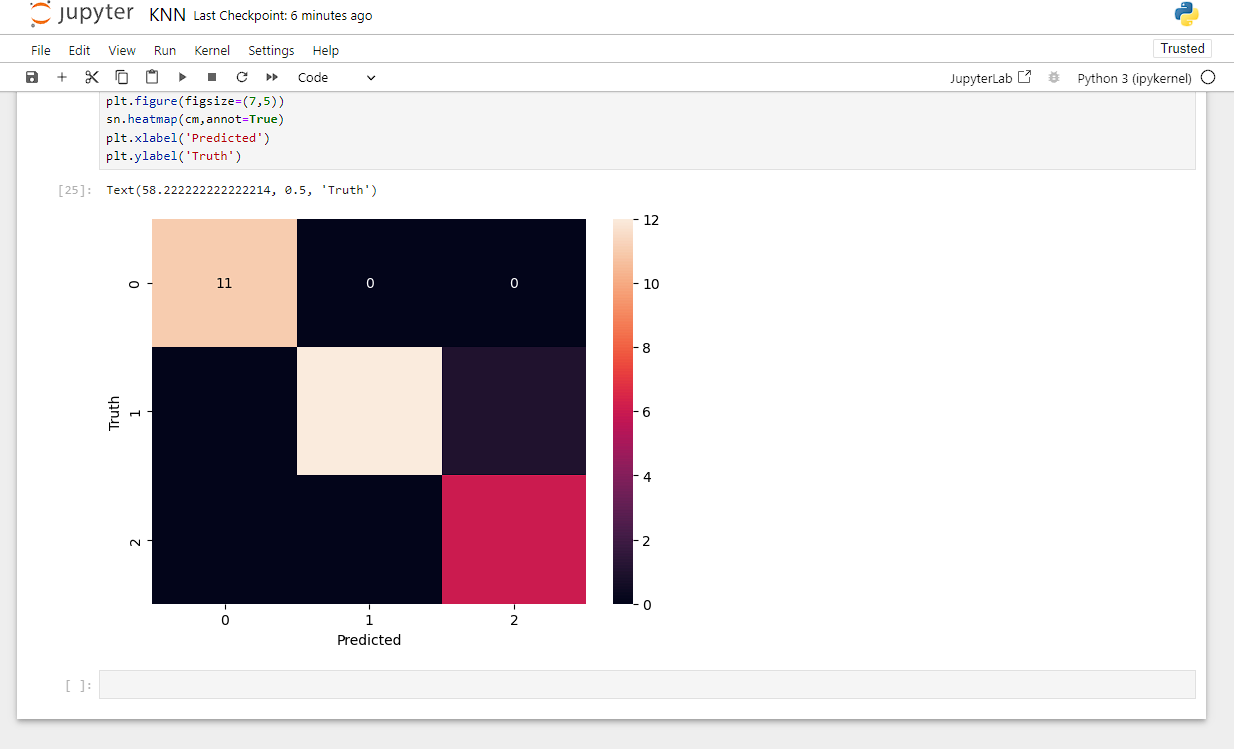




**Practical No:7**

**Aim: Implementation Of KNN Classification.**

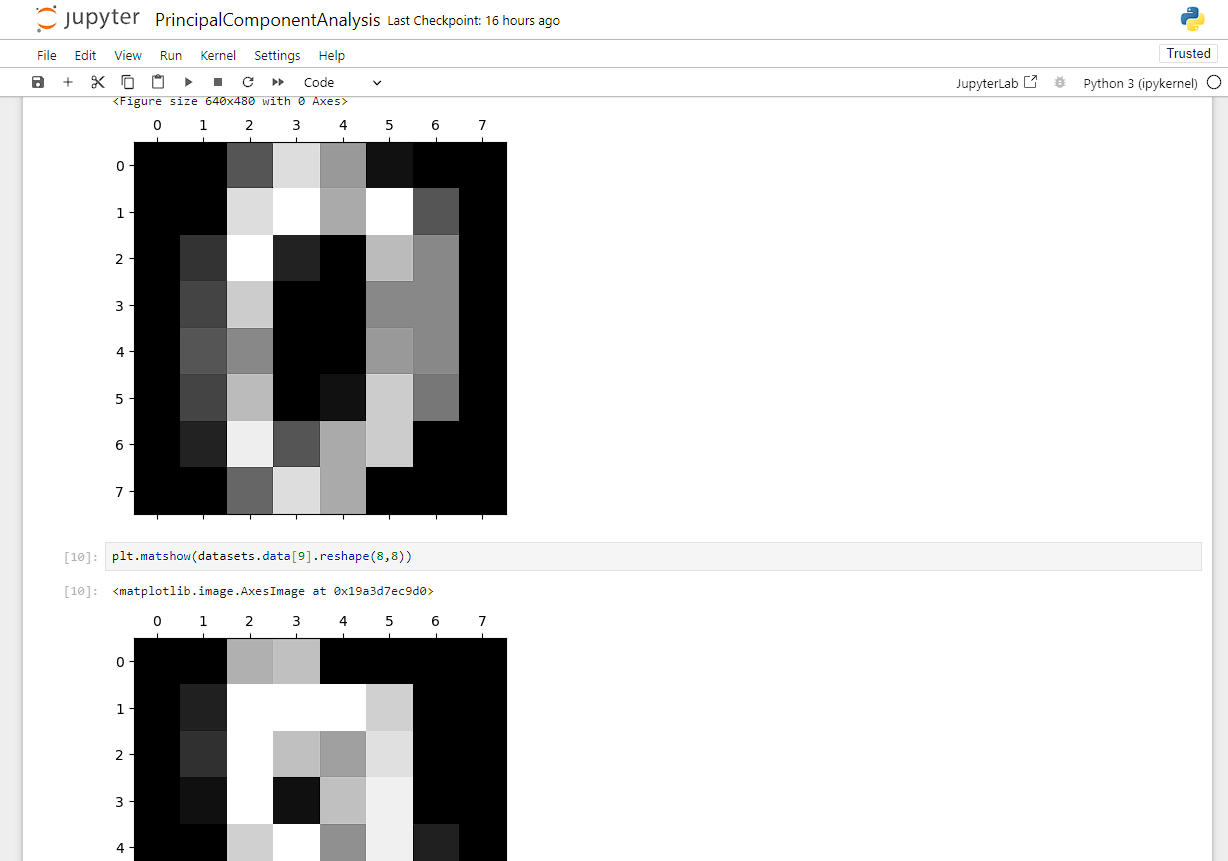
****

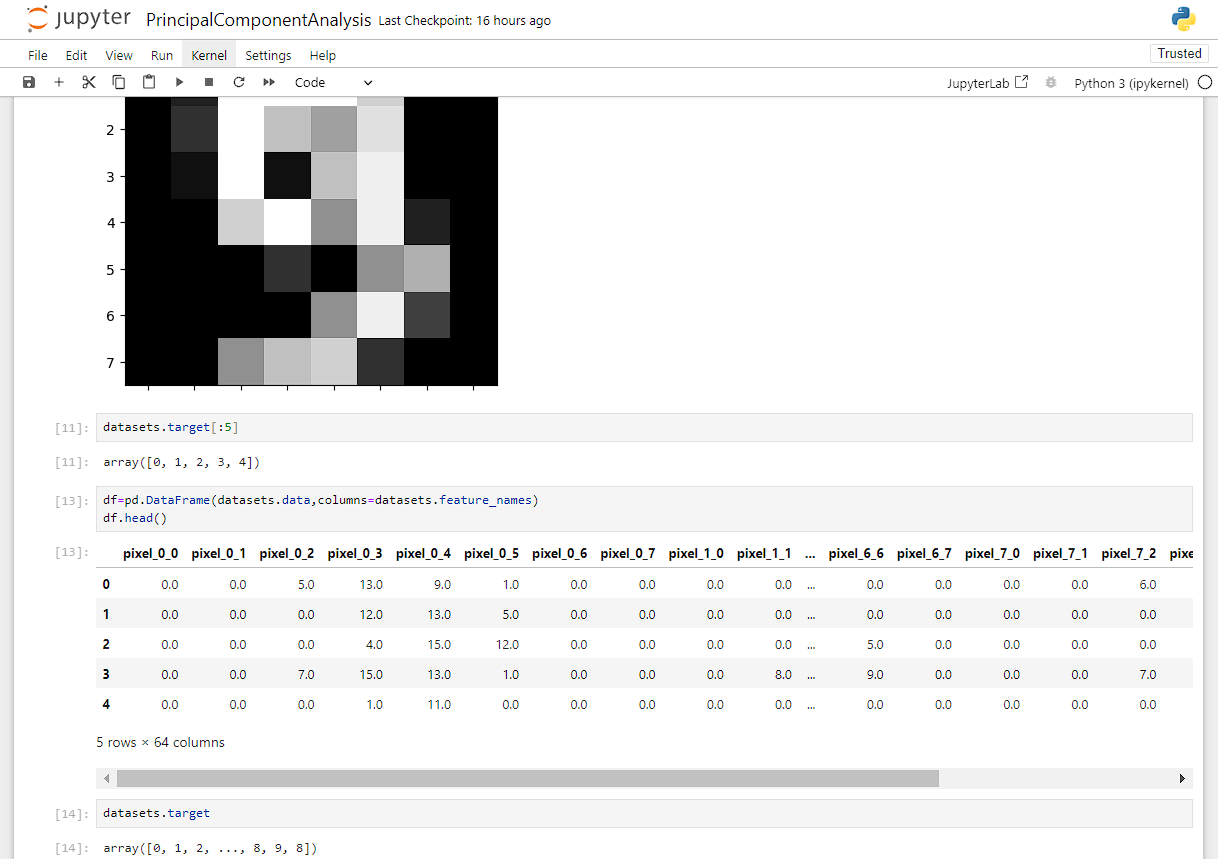
****

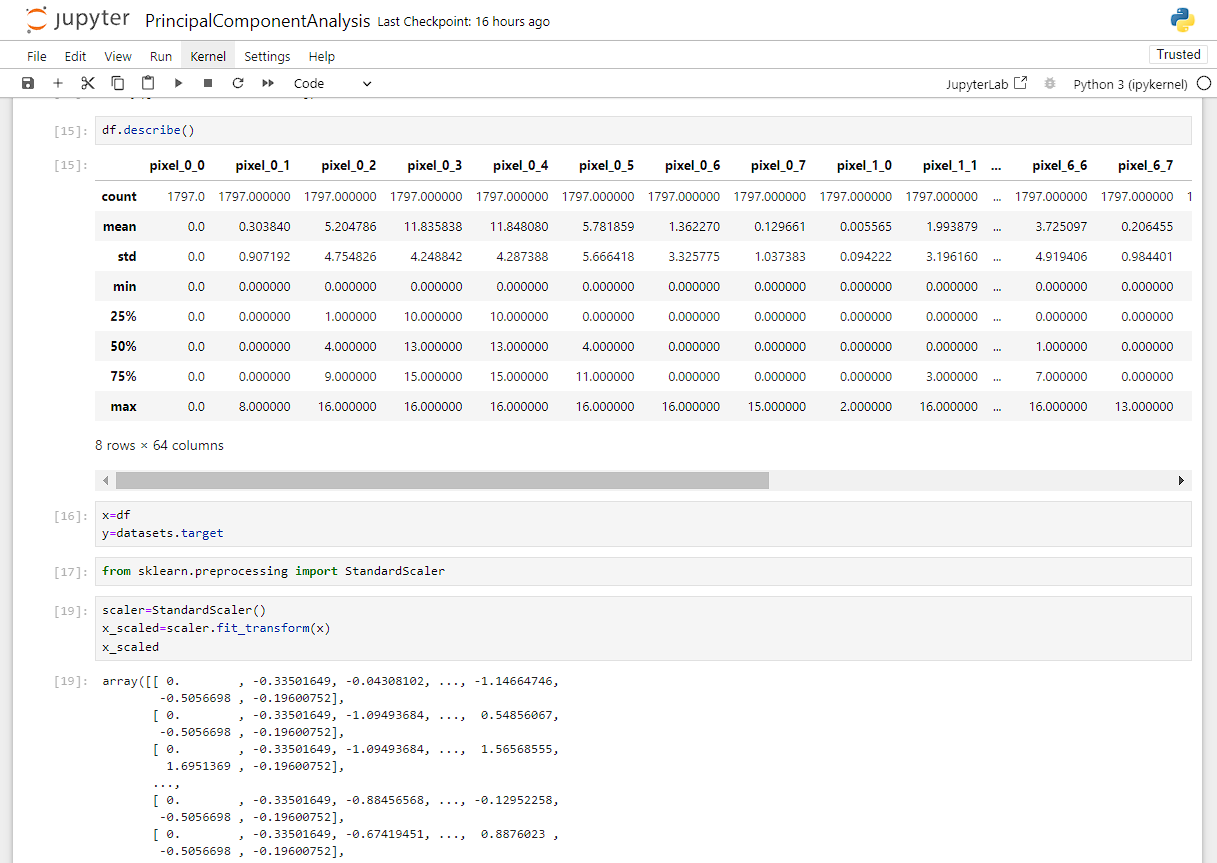
**Practical No:8**

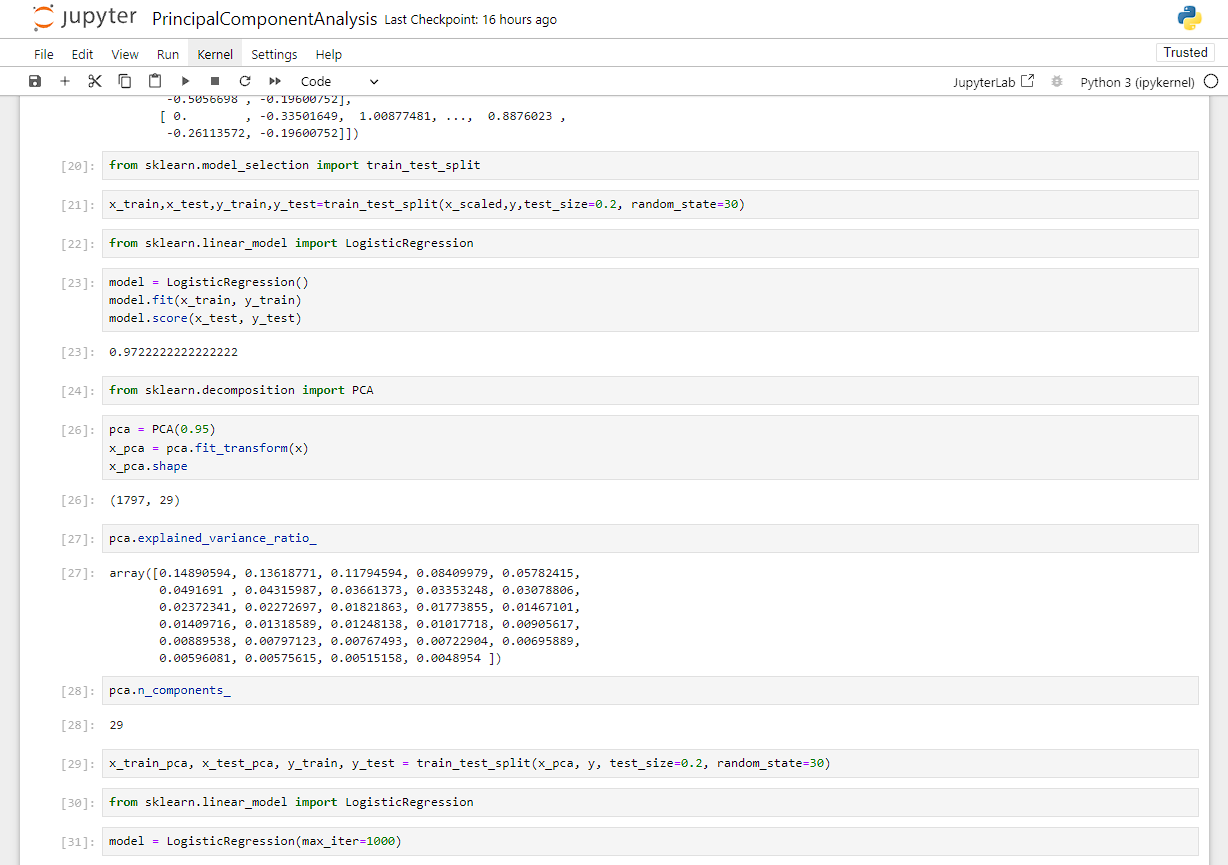
**Aim: Program to Implement Principal Component Analysis**

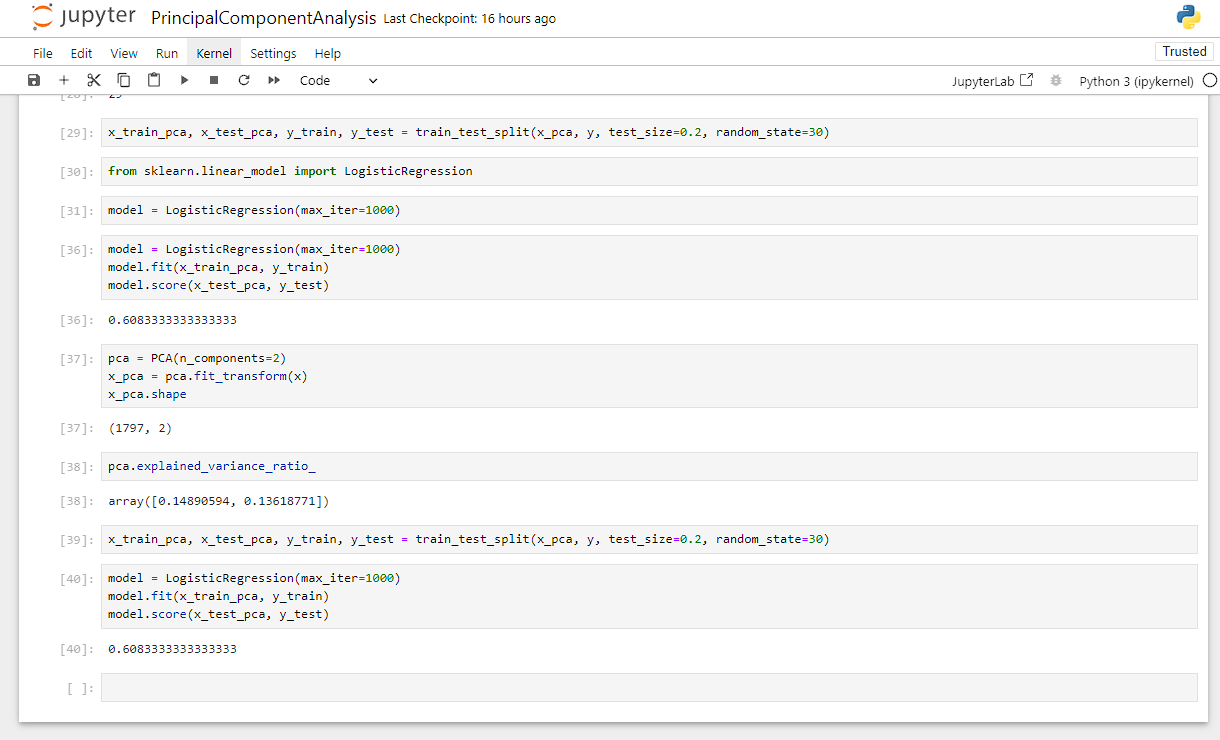
**Output:**

****

****

****

****

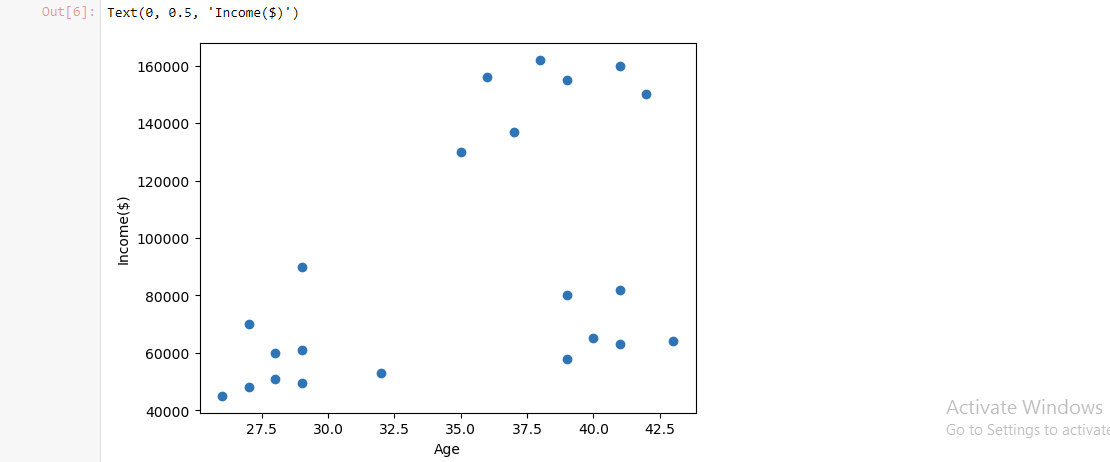
****

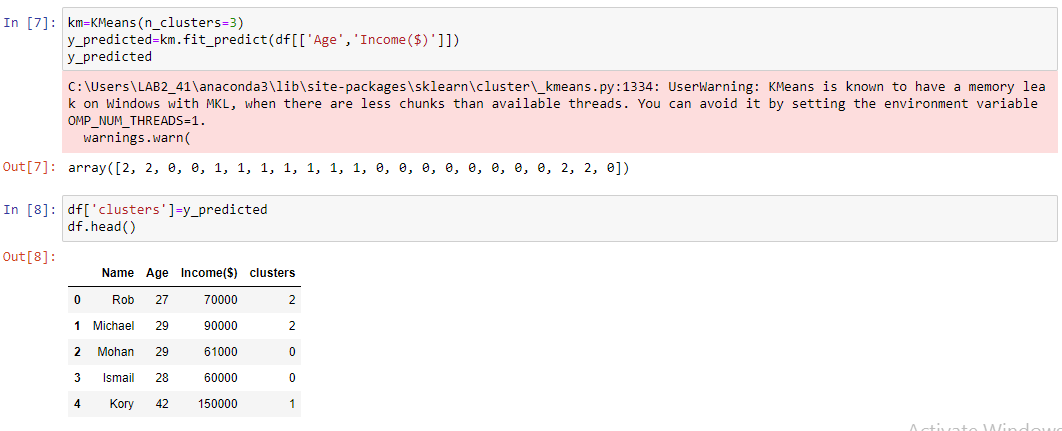
**Practical No: 9**

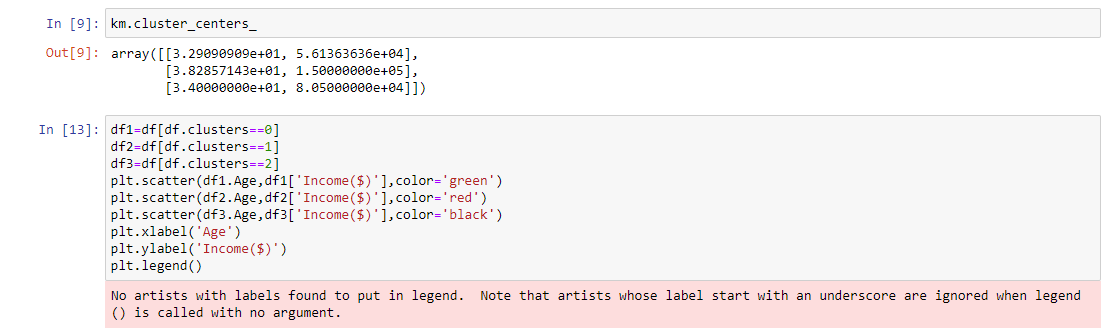
**Aim: Program to Implement K-Means Algorithm**

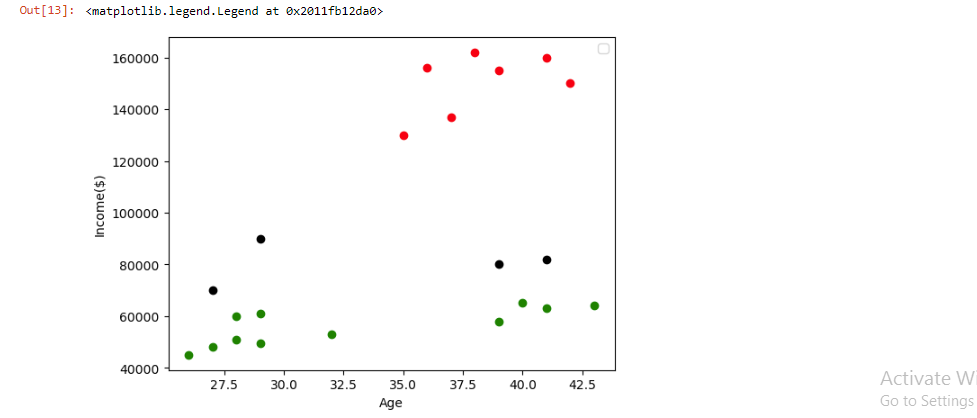
**Output:**



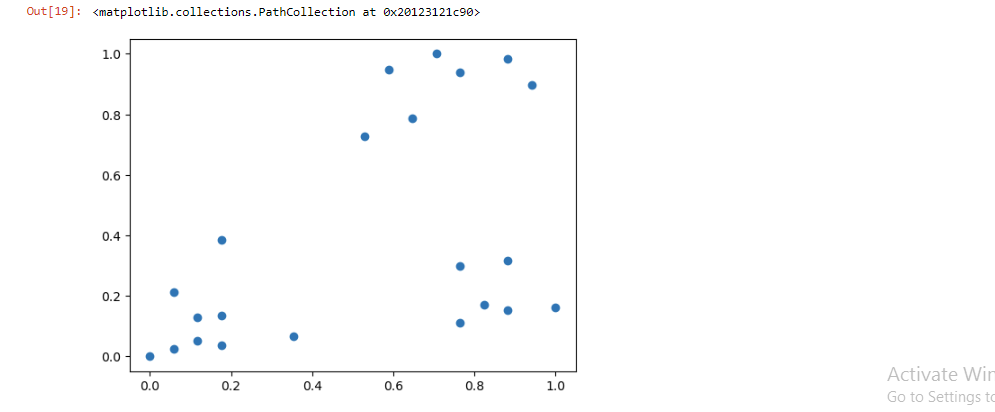


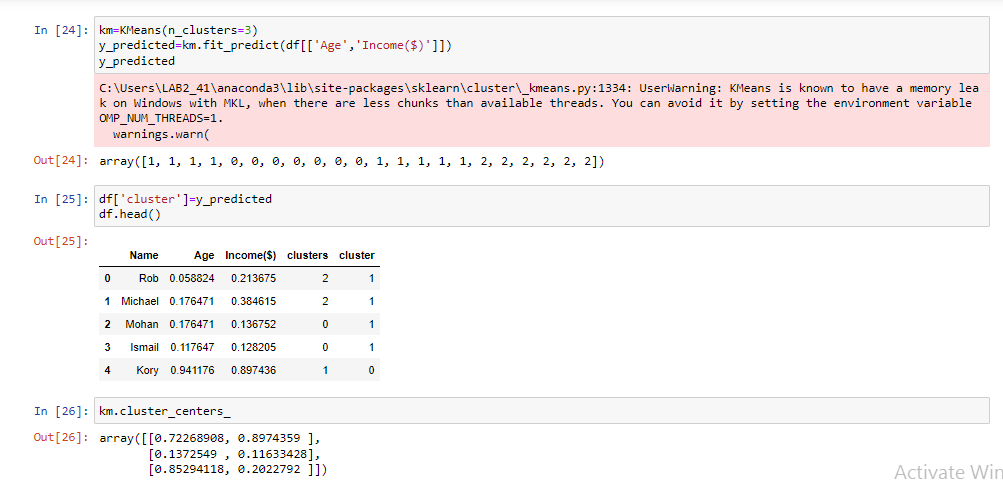


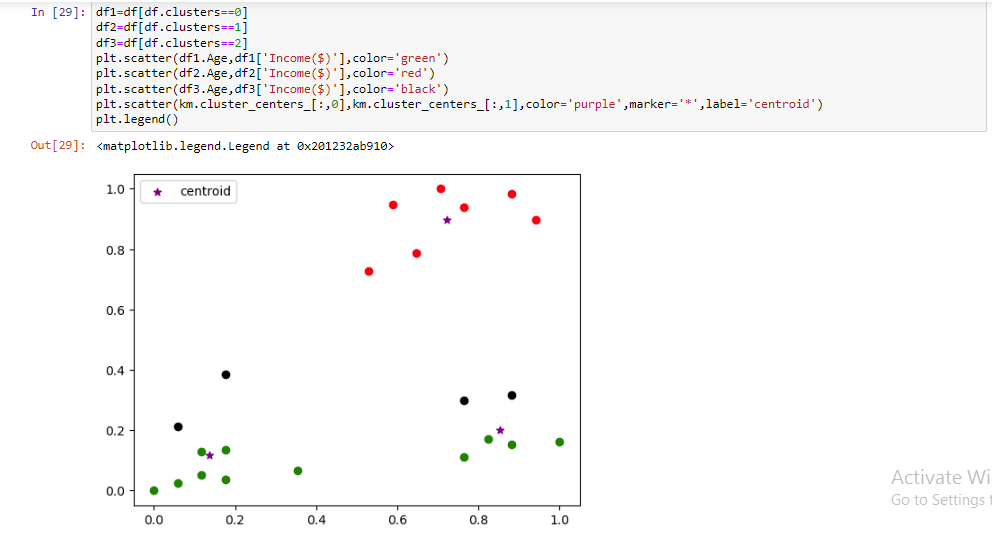




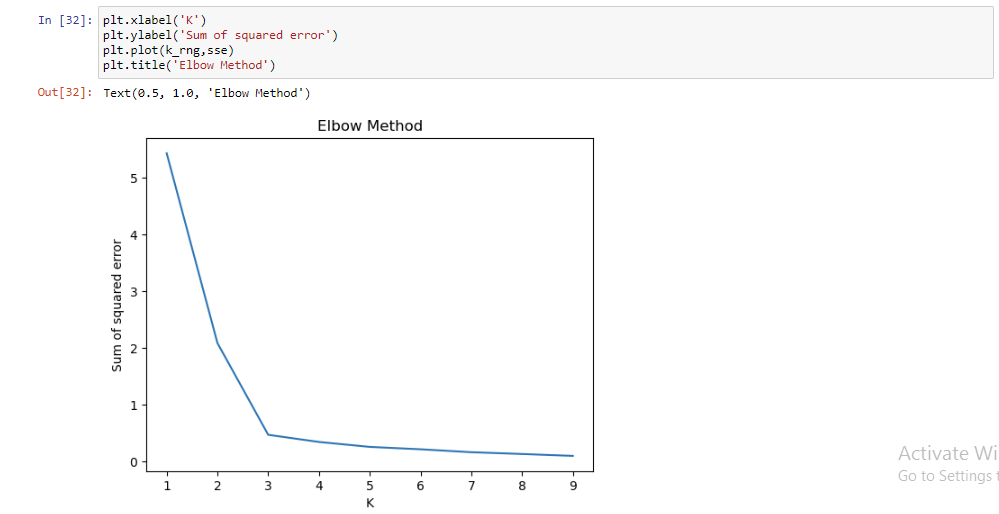








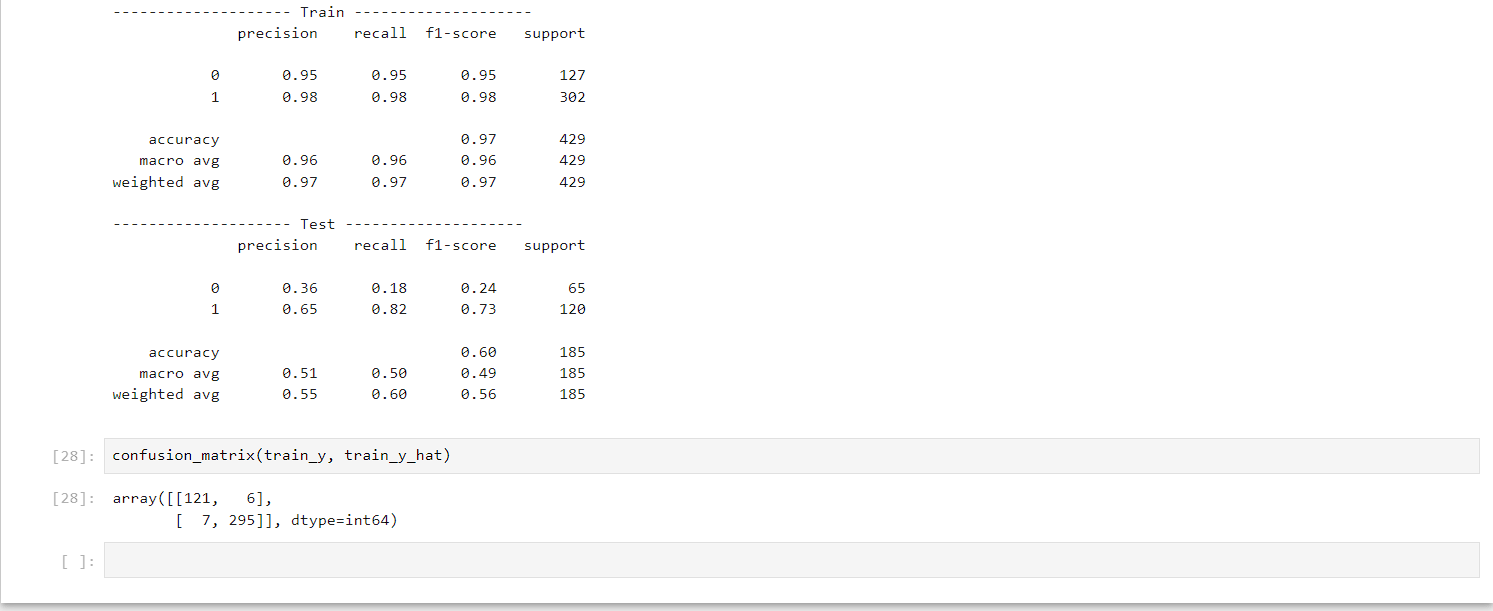
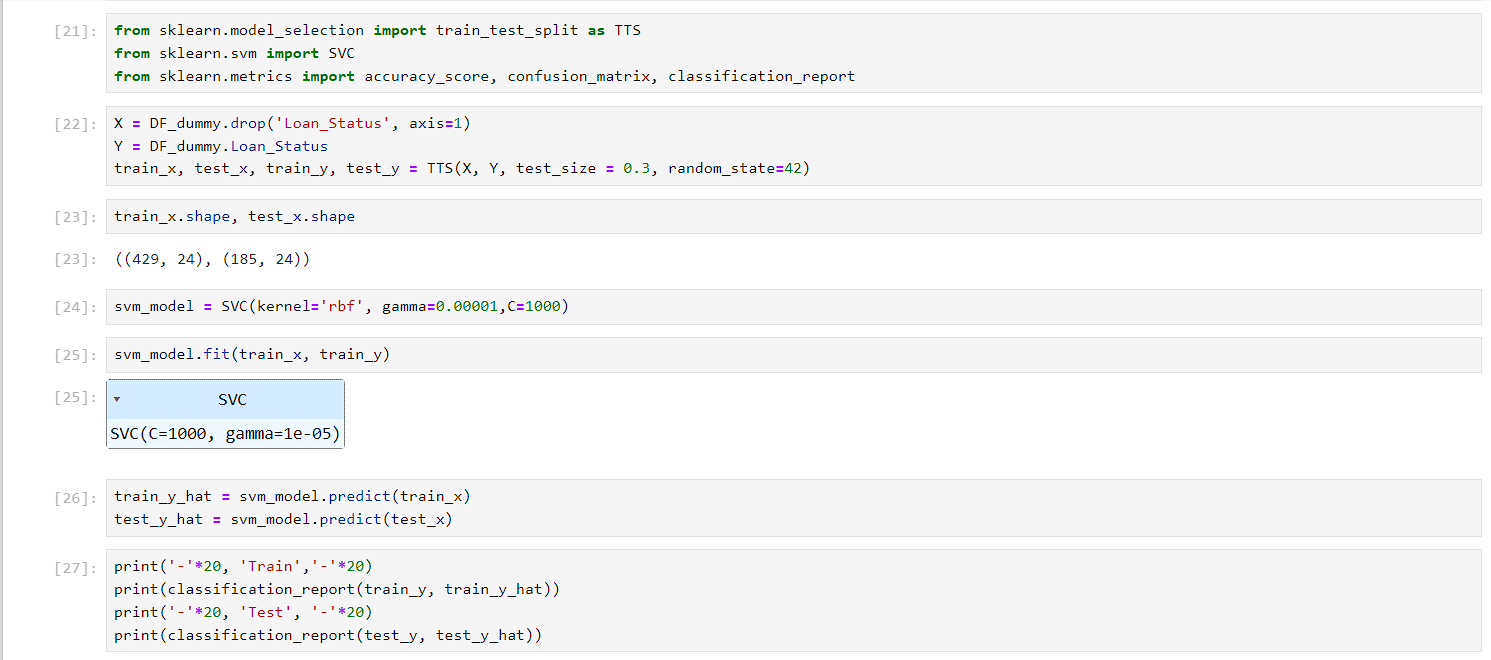
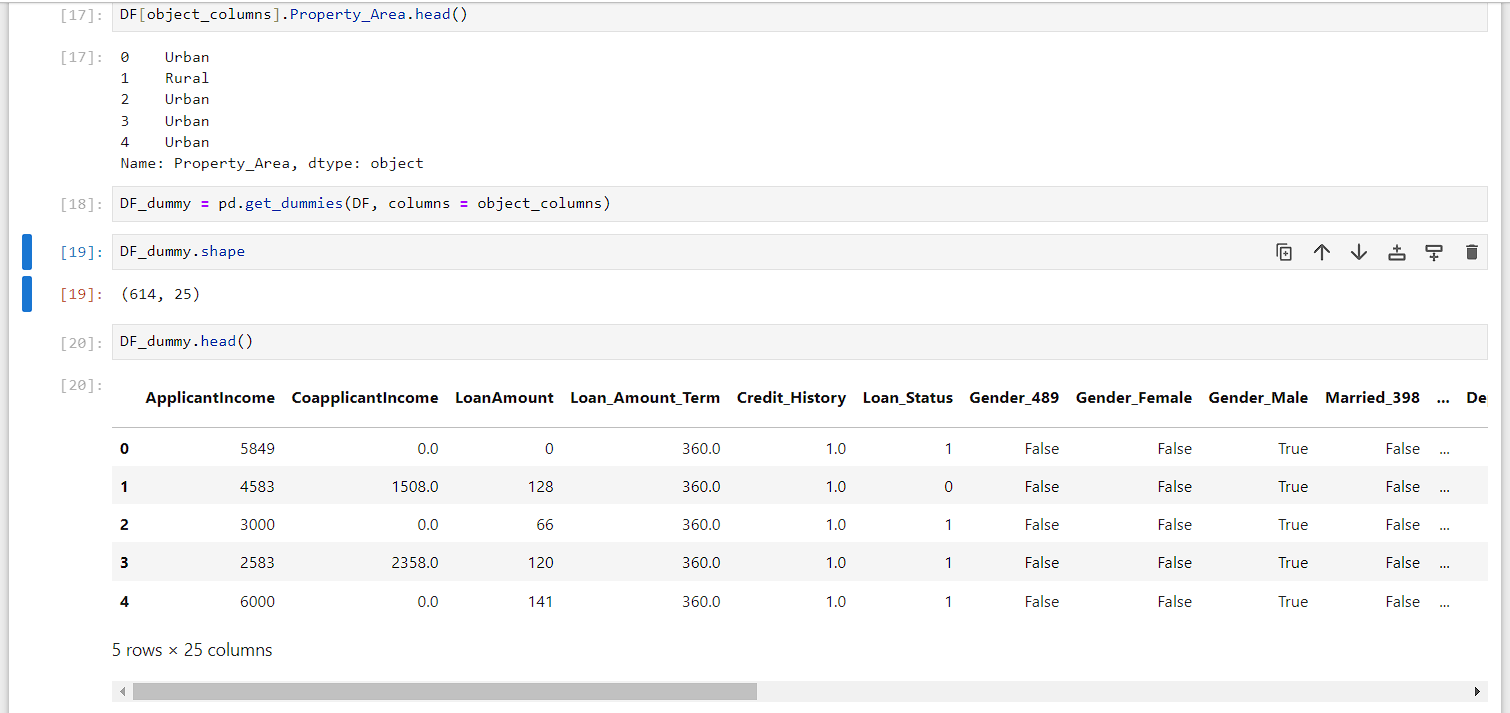
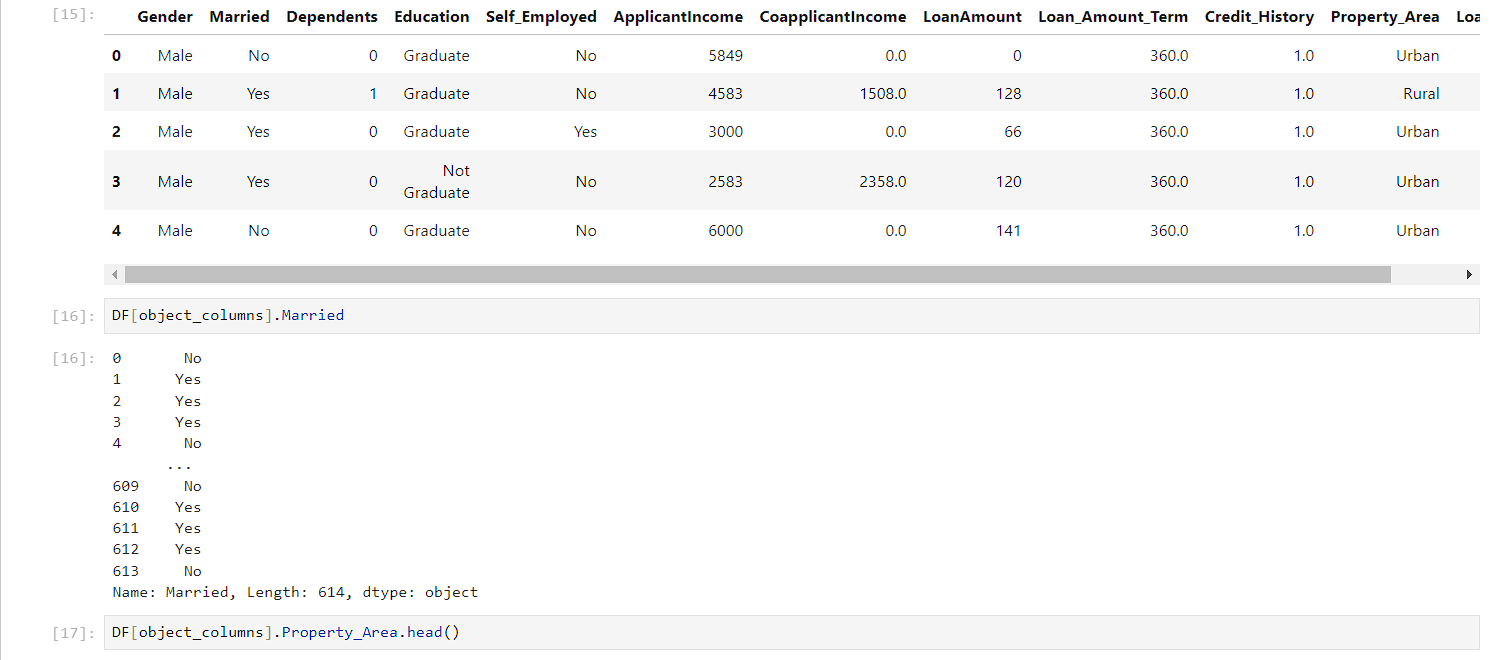
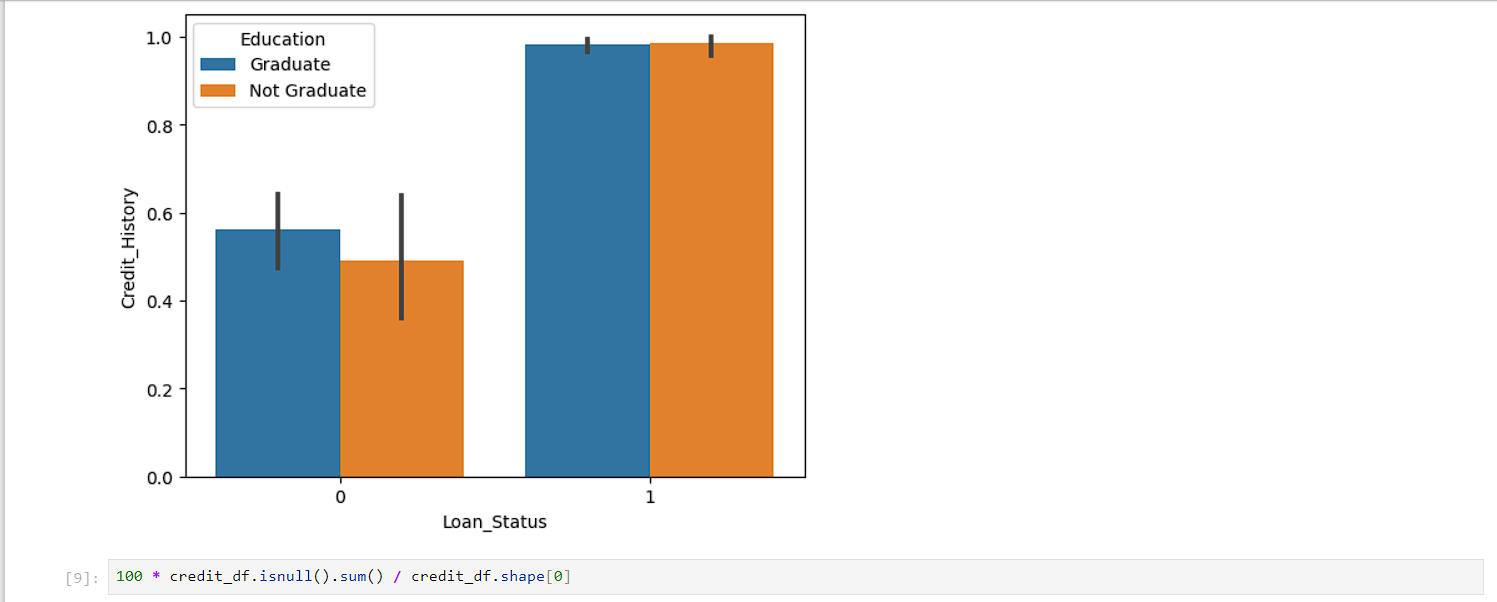
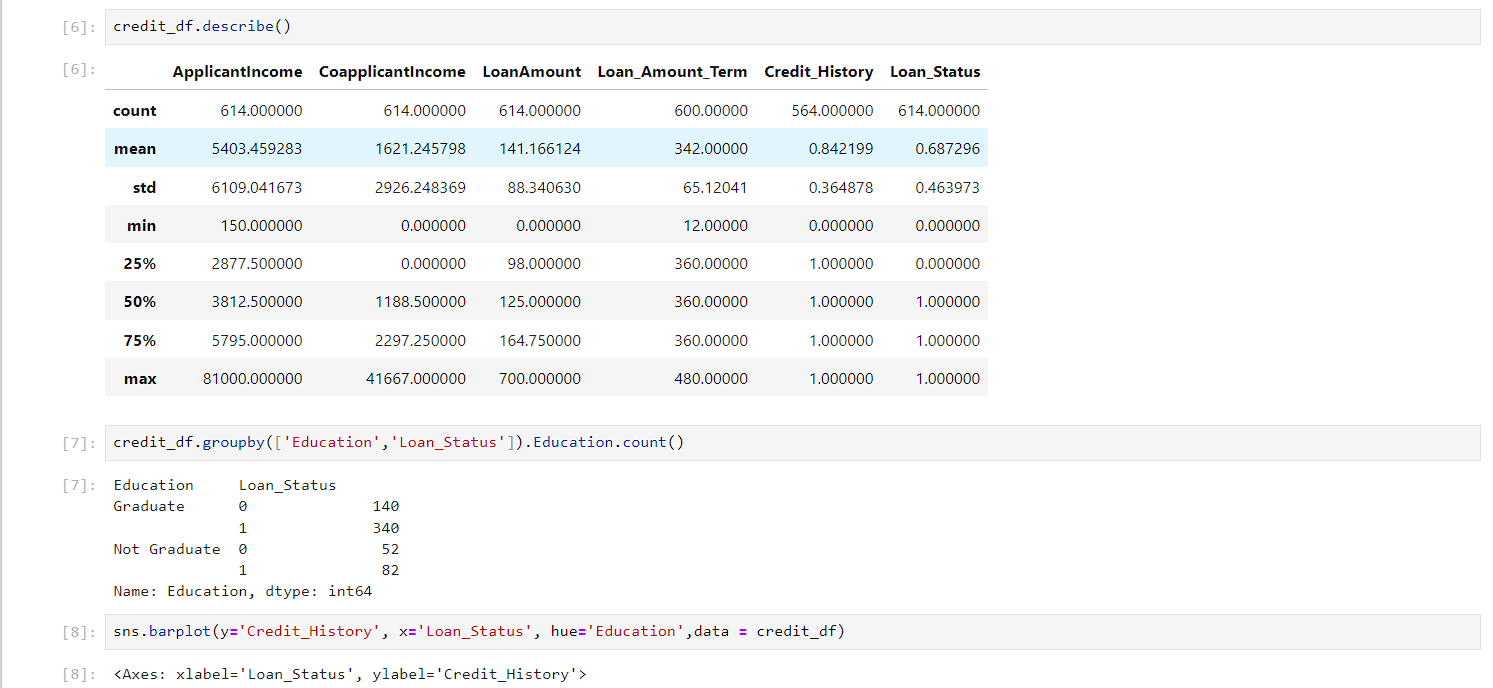
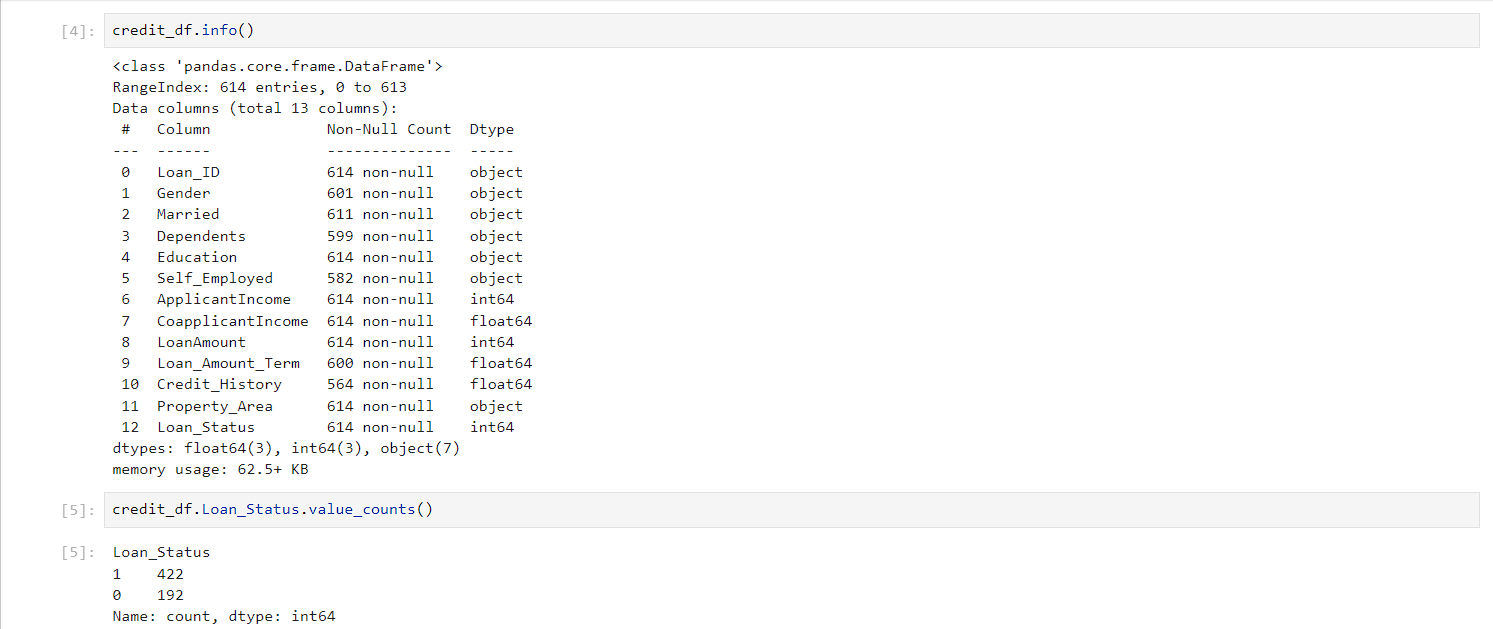
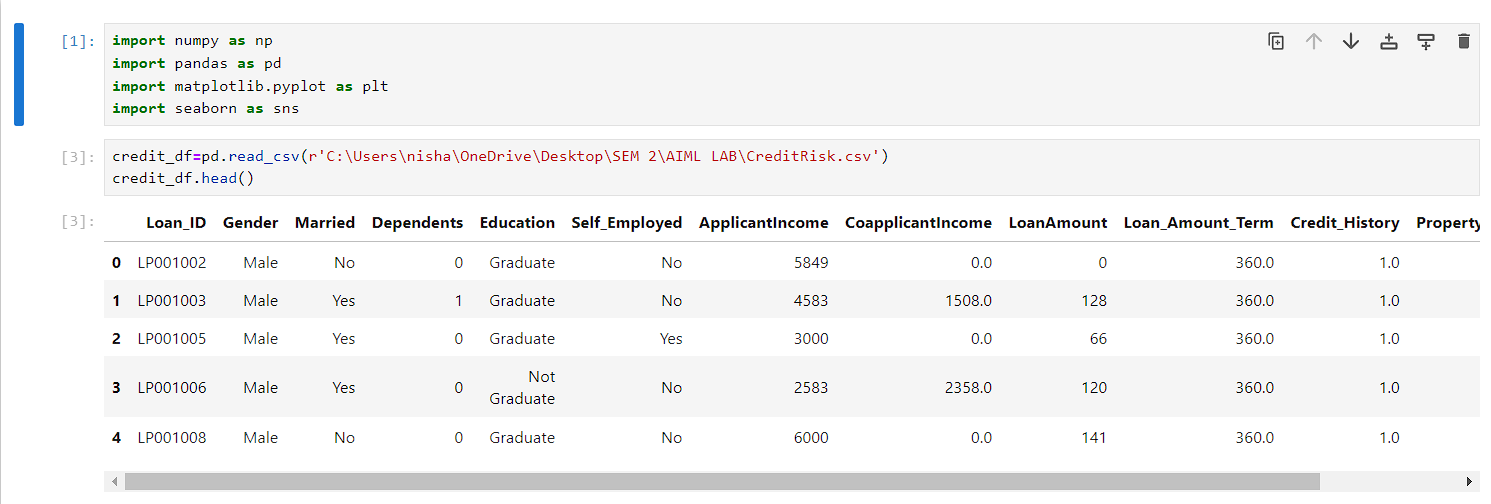




**Practical No: 10**

**Aim: Implementation Of Support Vector Machine**.

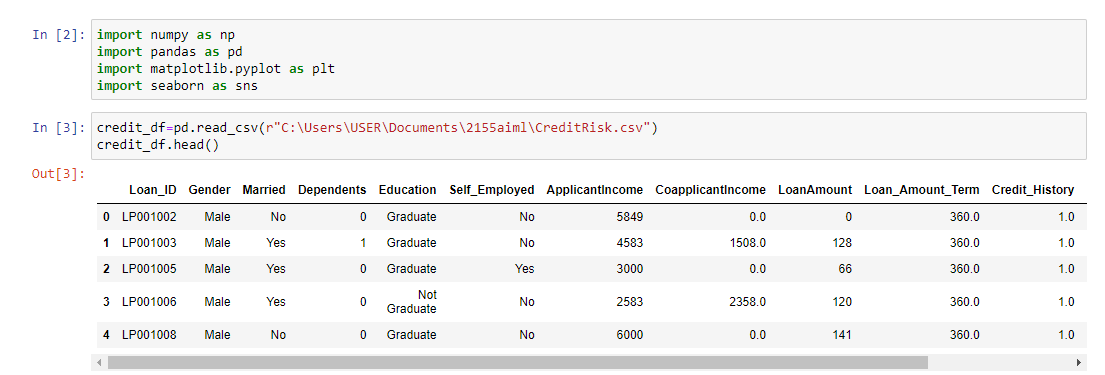
**Output:**

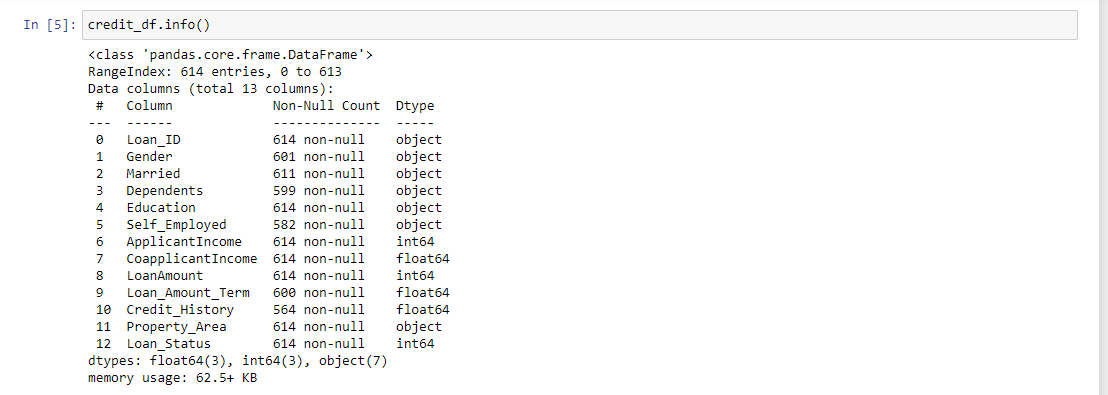


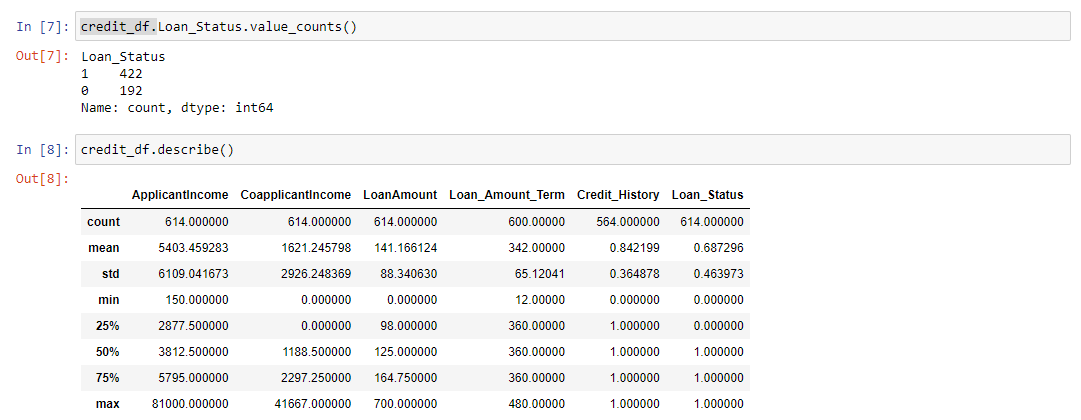
**Practical No: 11**

**Aim: Program to Implement Decision Tree**

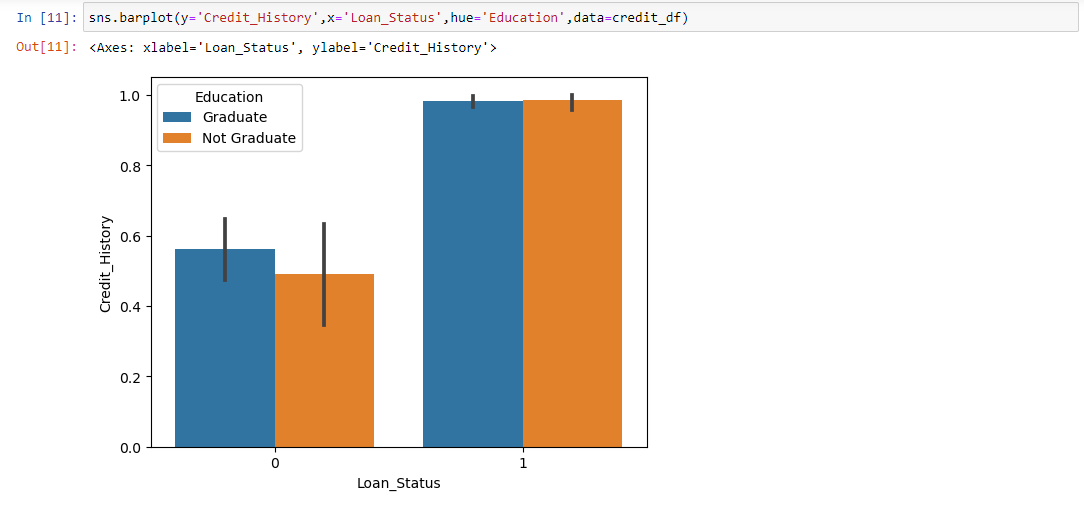
**Output:**

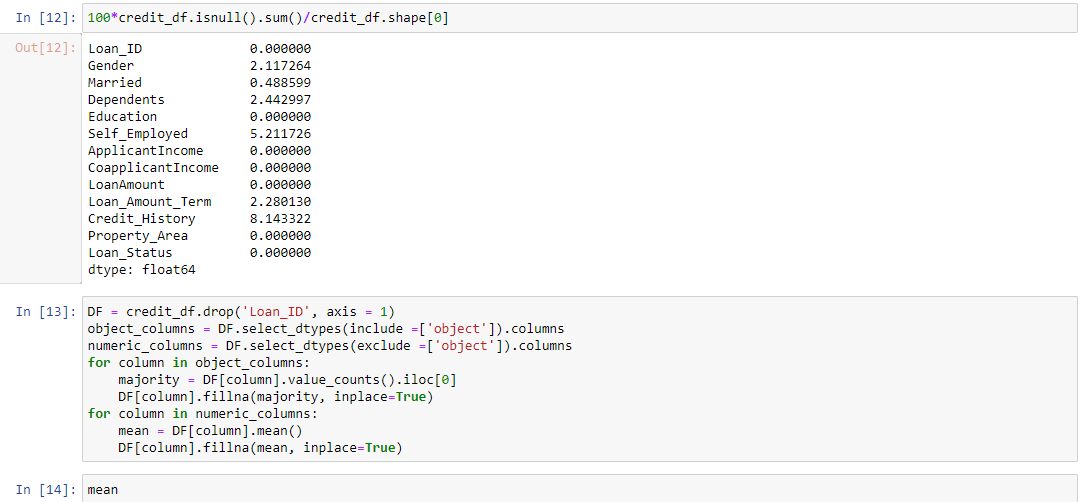


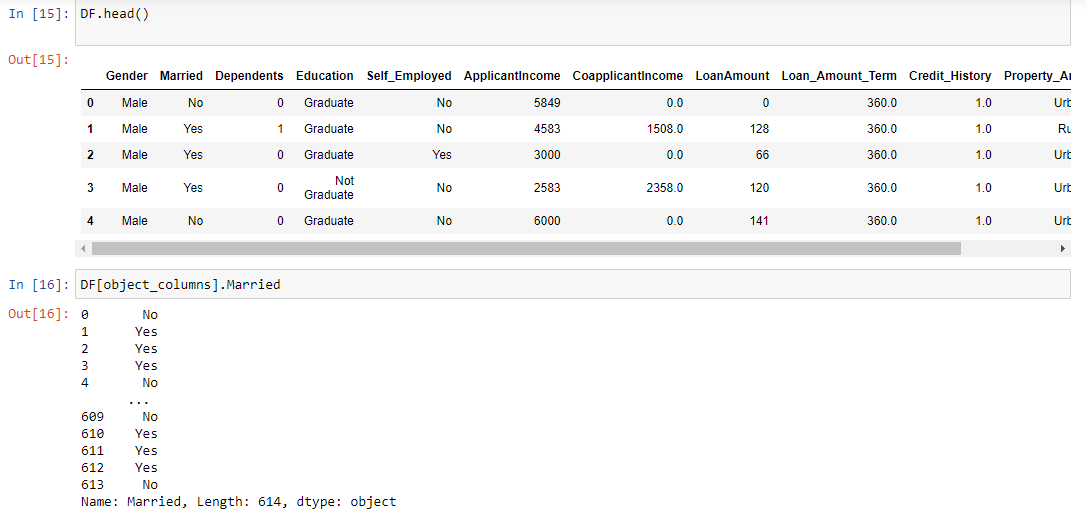


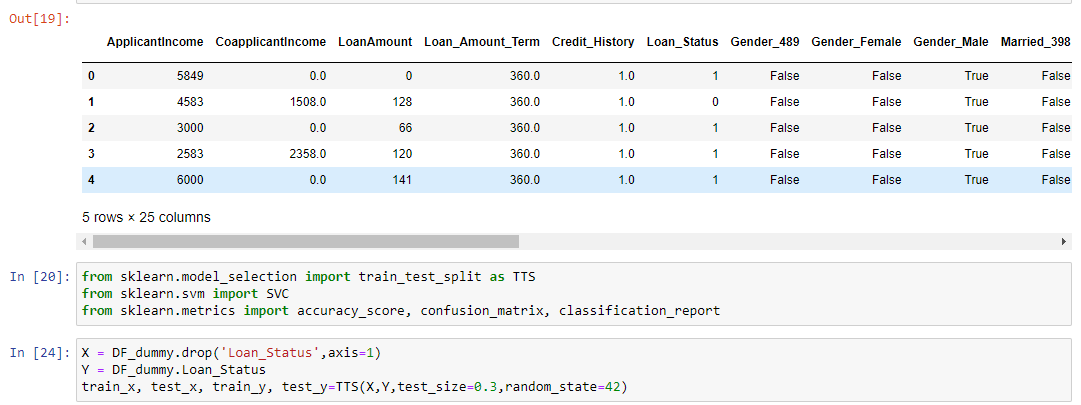


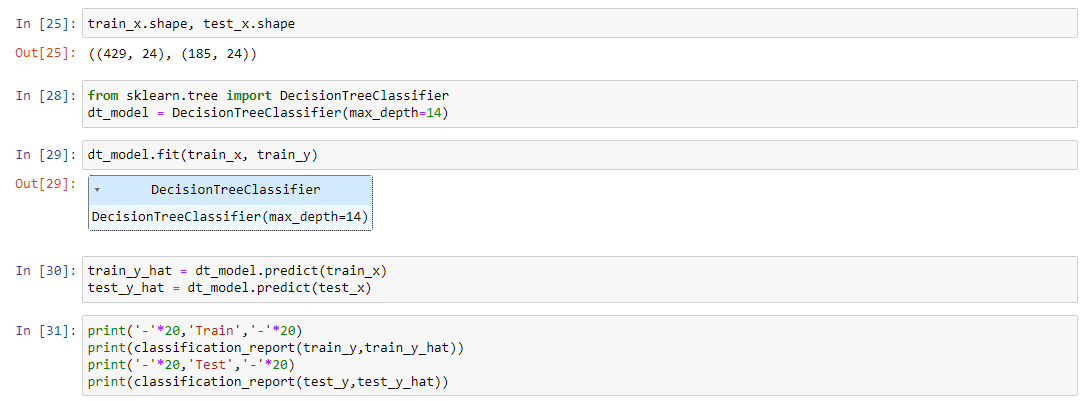


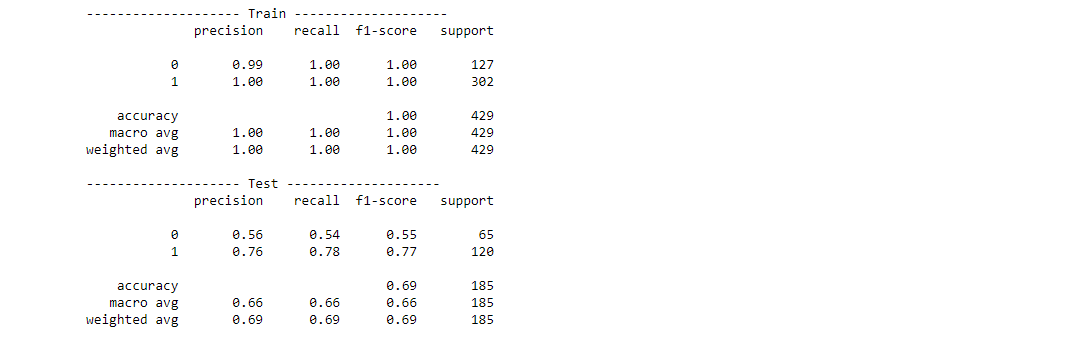


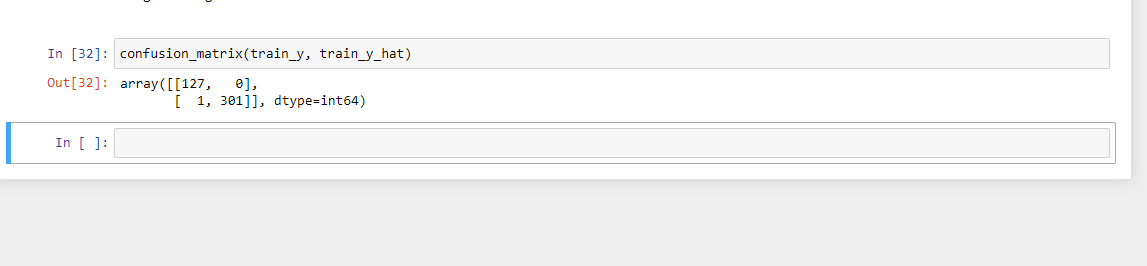












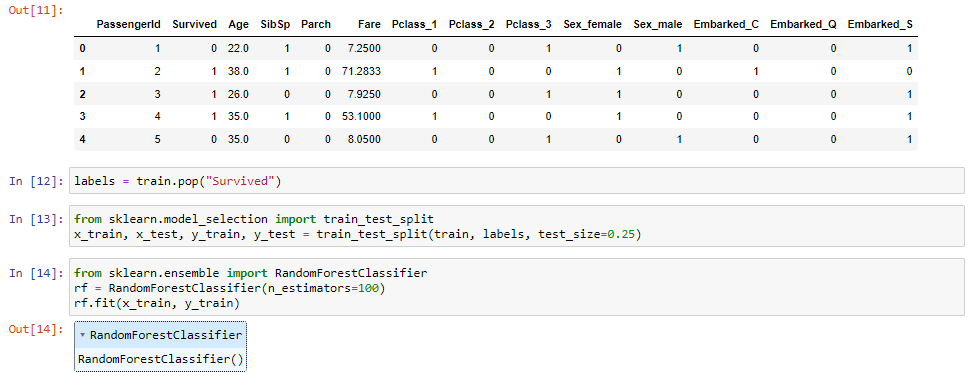
**Practical No: 12**

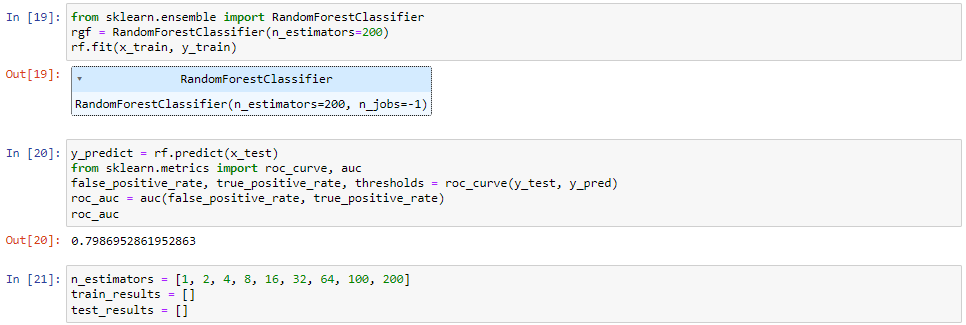
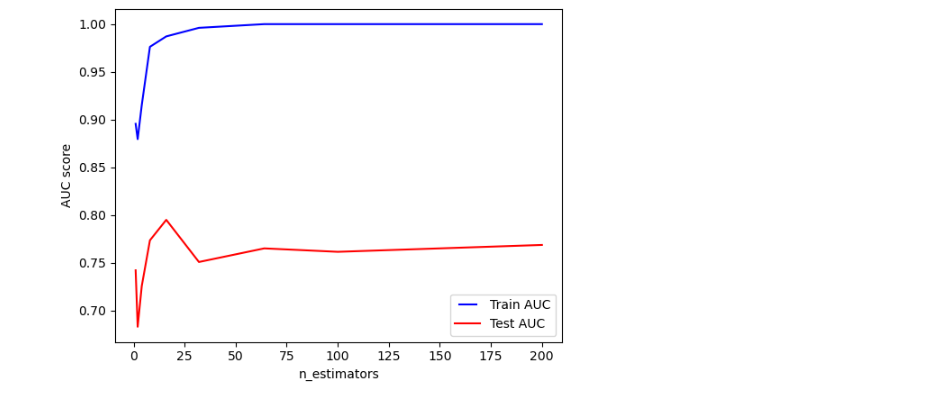
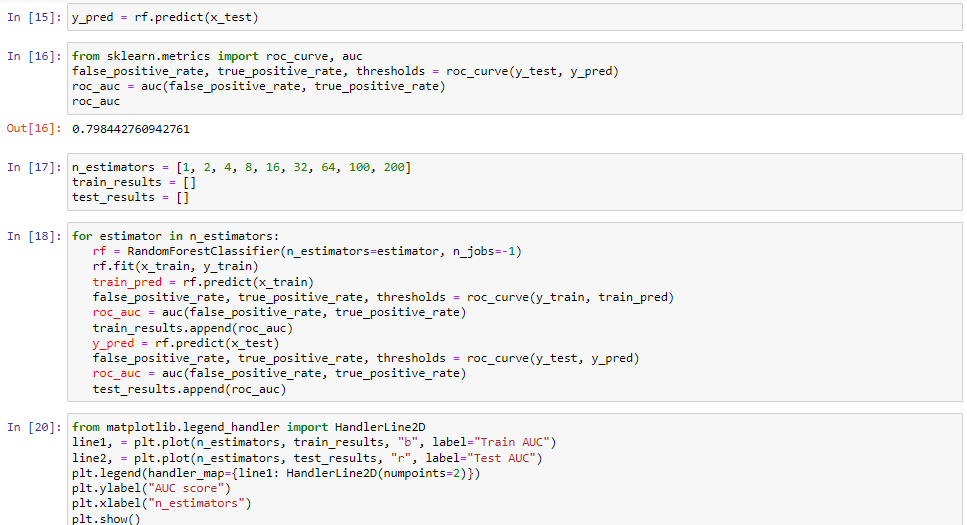
**Aim: Program to Implement Random Forest**

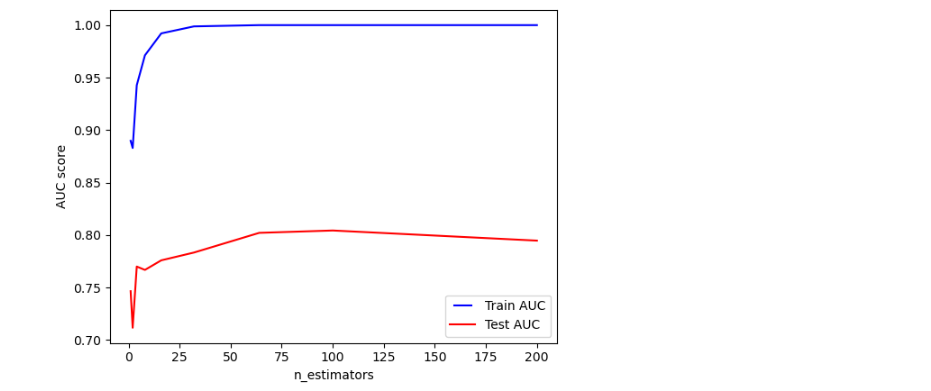
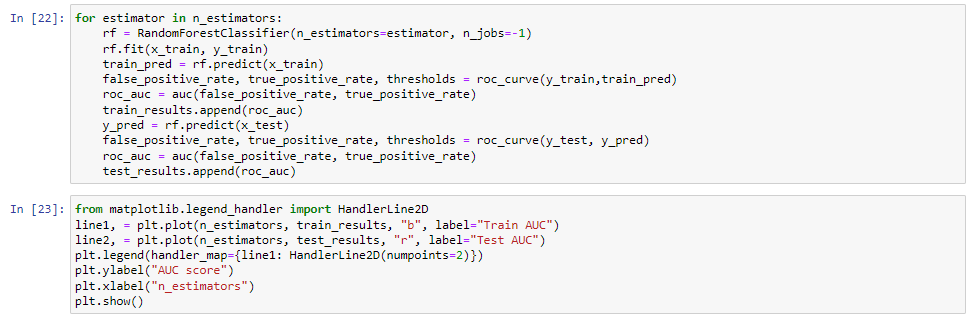
**Output:**







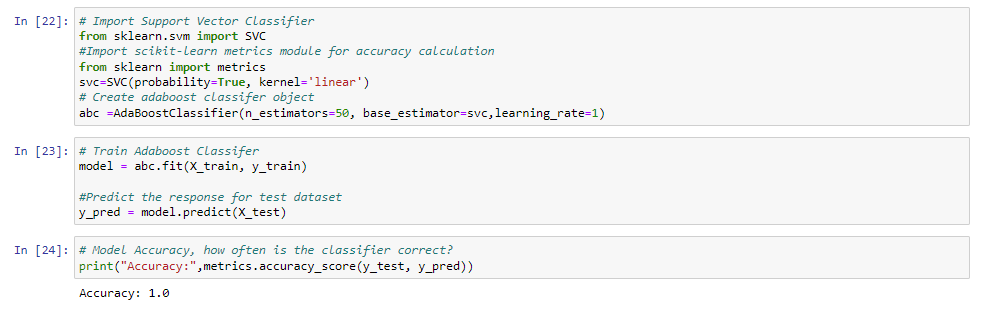




**Practical No: 13**

**Aim: Program to Implement AdaBoost**

**Output:**



**Practical No: 14**

**Aim: Program to Implement Gradient Boosting**

**Output:**

