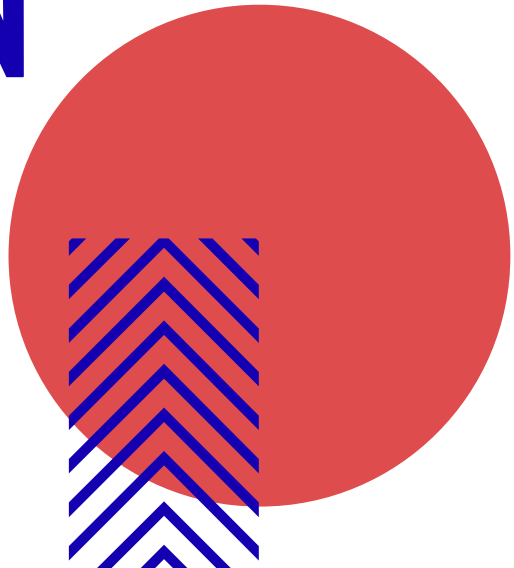
MODELOS DE CLASIFICACIÓN

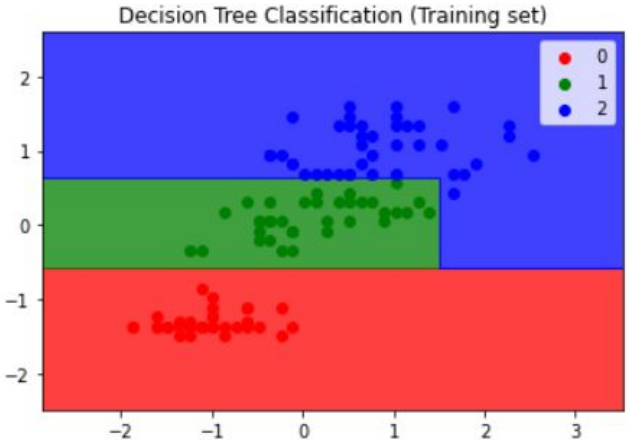
Roberto Lacayo
Fabricio Juarez
Anesveth Maatens

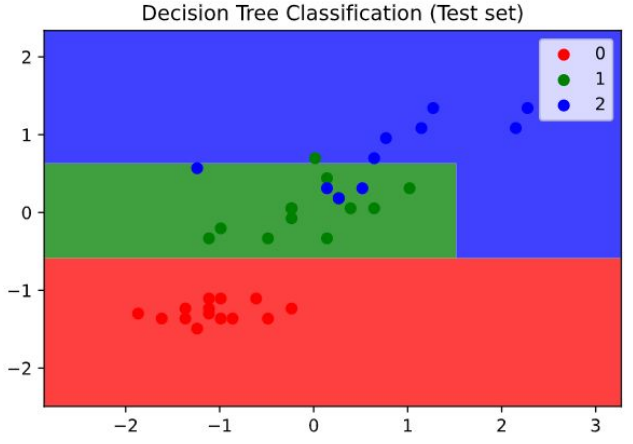


DECISION TREE IRIS

```
[ ]  # Making the Confusion Matrix Me sirve para saber que tan exactos son los datos
     from sklearn.metrics import confusion_matrix
     cm = confusion_matrix(y_test, y_pred)
     print(cm)
     print("acurrancy:")
     print((14+13+6)/(14+13+4+6+1))
```

```
[[14  0  0]
 [ 0 13  1]
 [ 0  4  6]]
acurrancy:
0.868421052631579
```





NAIVE BAYES IRIS

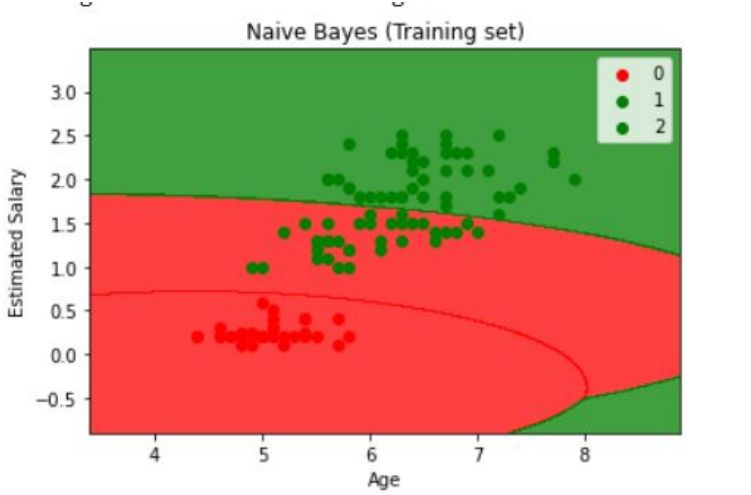
Making the Confusion Matrix

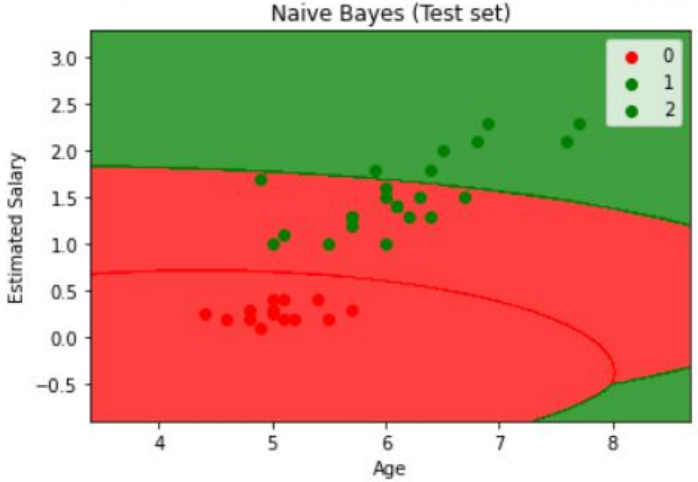
```
[ ]  from sklearn.metrics import confusion_matrix
     cm = confusion_matrix(y_test, y_pred)
     print(cm)
```

```
[[14  0  0]
 [ 0 13  1]
 [ 0  4  6]]
```

```
[ ]  accuracy1 = ((13+14+6)/(14+13+6+4+1))
     print(accuracy1)
     from sklearn.metrics import accuracy_score
     accuracy2 = accuracy_score(y_test, y_pred)
     print(accuracy2)
```

```
0.868421052631579
0.868421052631579
```

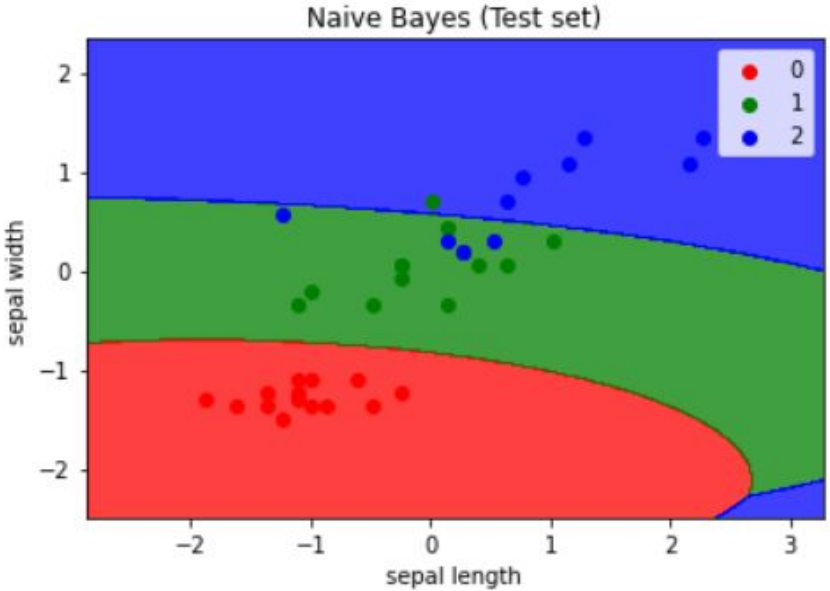


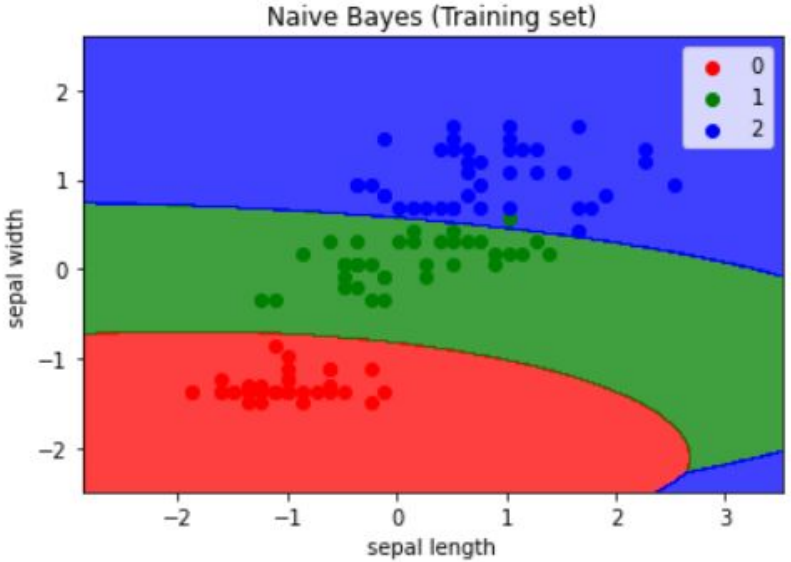


KNN IRIS

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.868421052631579
```

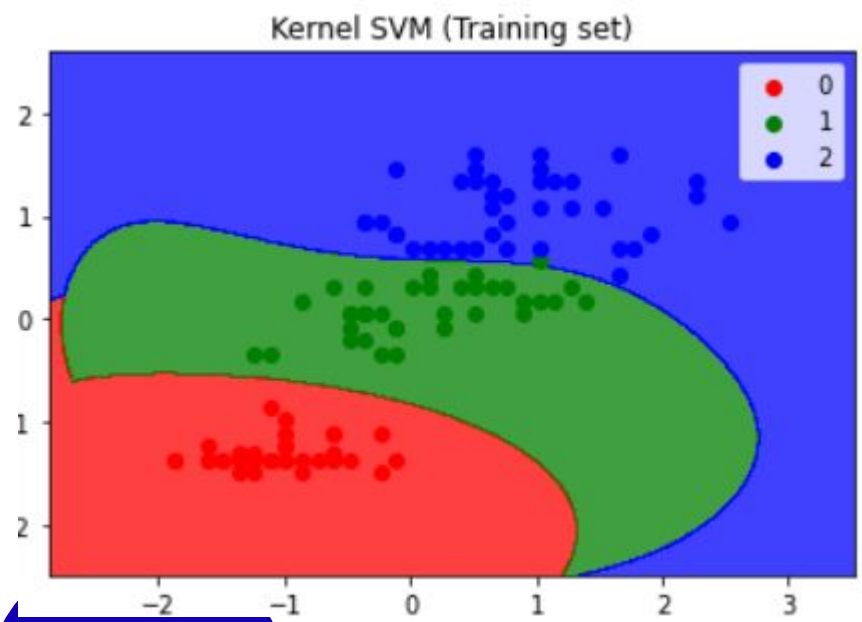


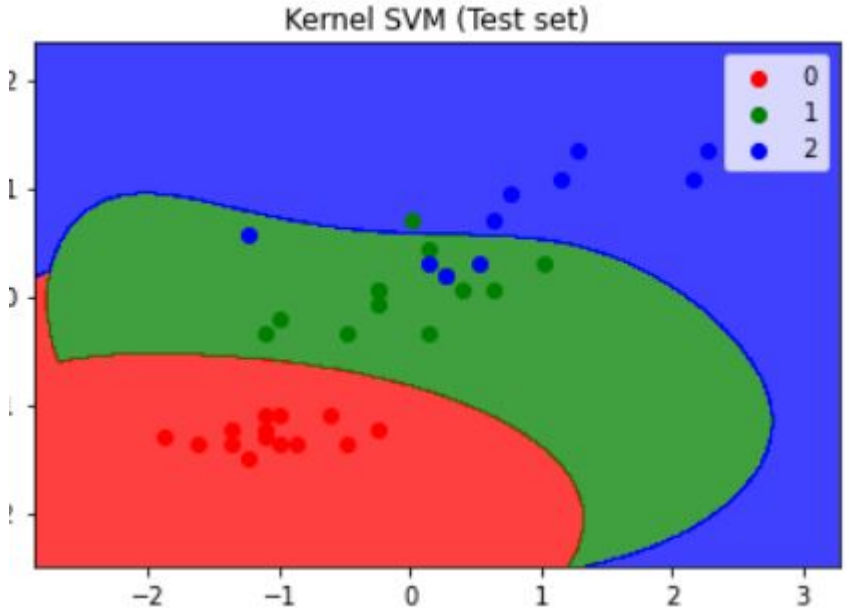


SUPPORT VECTOR MACHINE IRIS

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.868421052631579
```

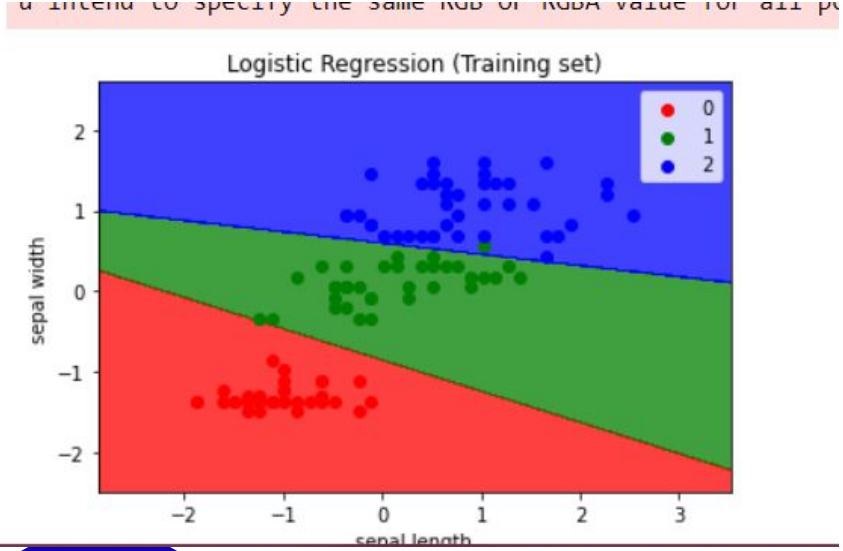


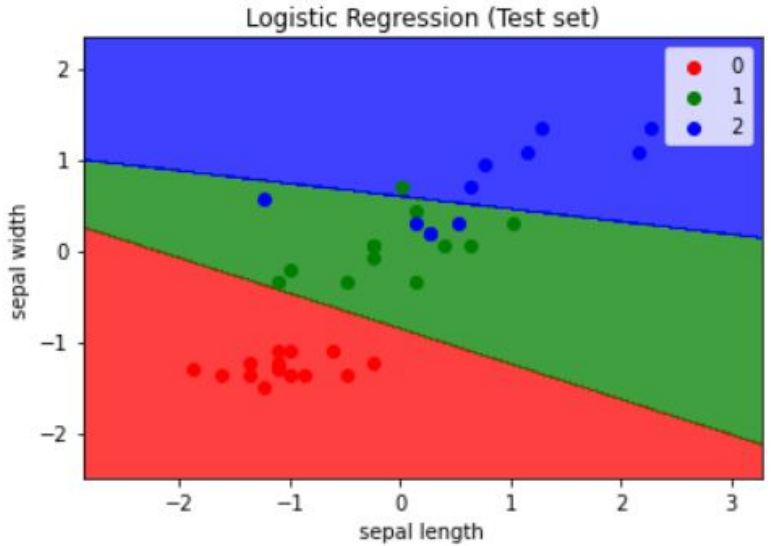


LOGISTIC REGRESSION IRIS

```
In [17]: from sklearn.metrics import accuracy_score
         print(accuracy_score(y_test, y_pred))
```

```
0.868421052631579
```

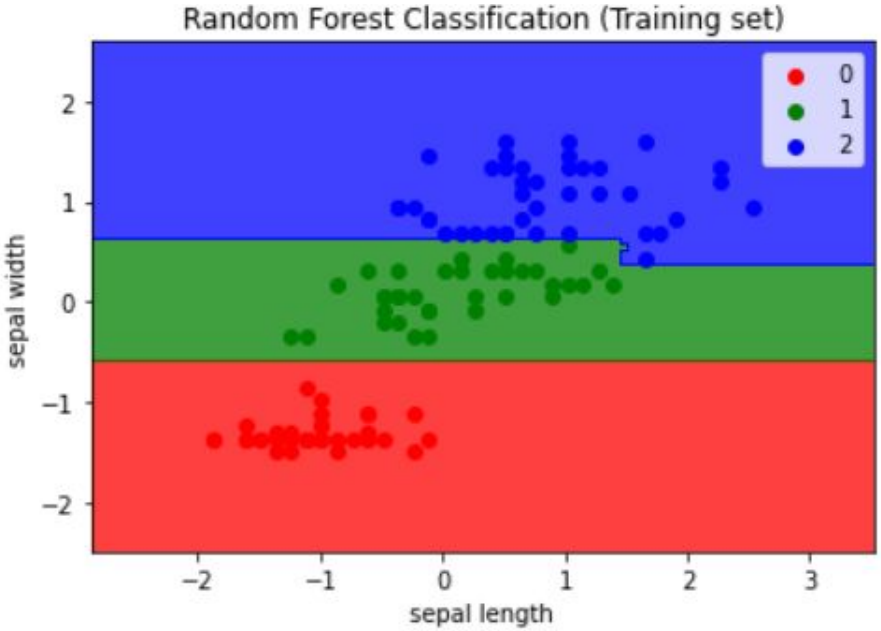


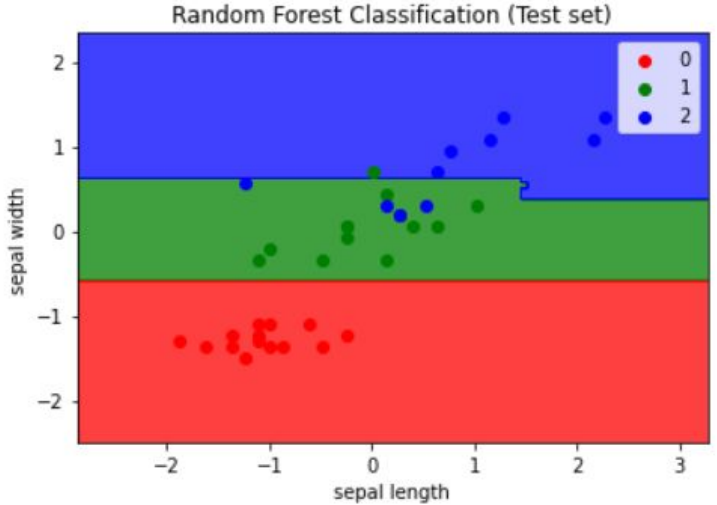


RANDOM FOREST CLASSIFIER IRIS

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.868421052631579
```





NAIVE BAYES SNA

Making the Confusion Matrix

```
[7]  from sklearn.metrics import confusion_matrix
     cm = confusion_matrix(y_test, y_pred)
```

```
[8]  print(cm)
```

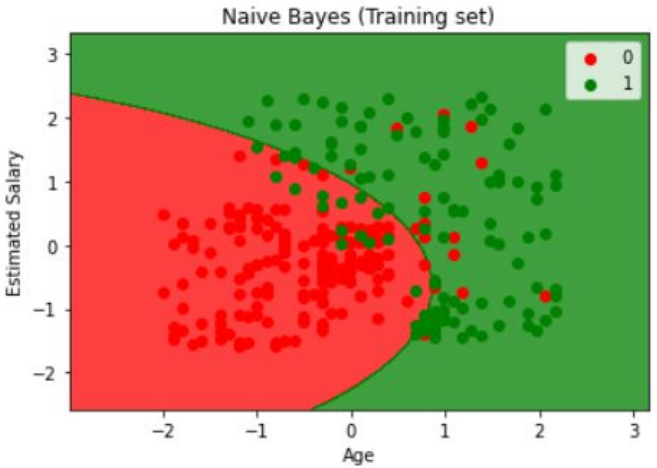
```
[[65  3]
 [ 7 25]]
```

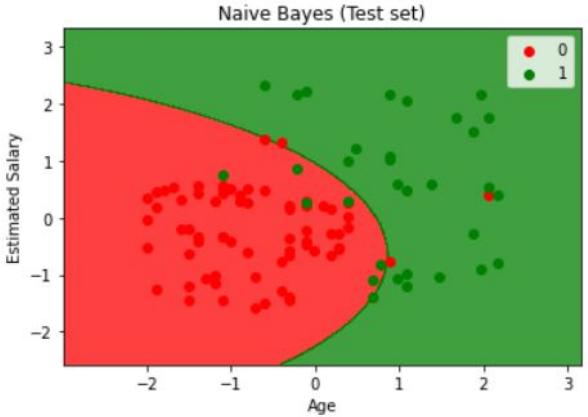
```
[10] ac = ((65+25)/(65+25+7+3))
     print(ac)
```

```
0.9
```

```
[6]: from sklearn.metrics import accuracy_score
     print(accuracy_score(y_test,y_pred))
```

```
0.9
```

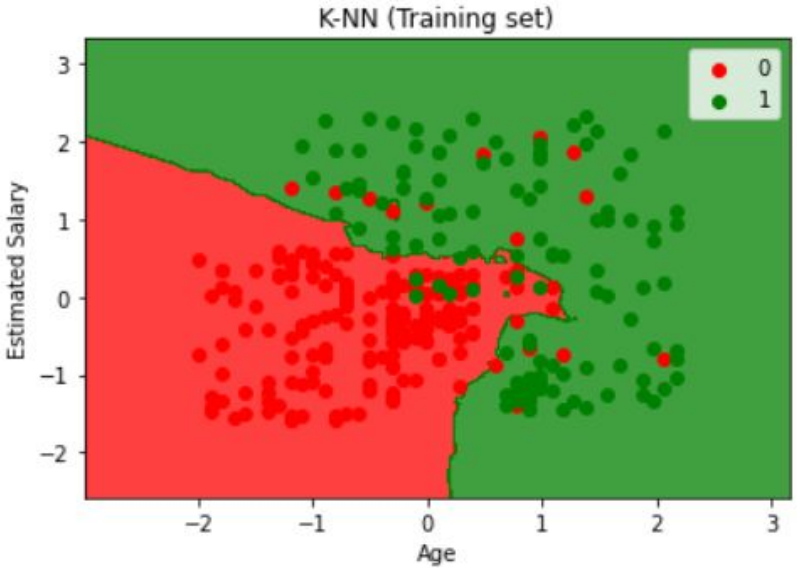


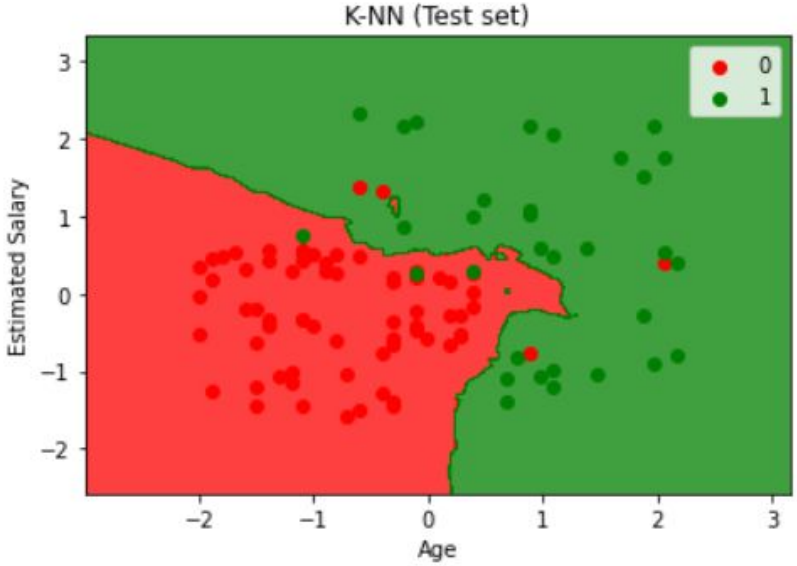


KNN SNA

```
In [11]: from sklearn.metrics import accuracy_score
         print(accuracy_score(y_test, y_pred))
```

```
0.93
```

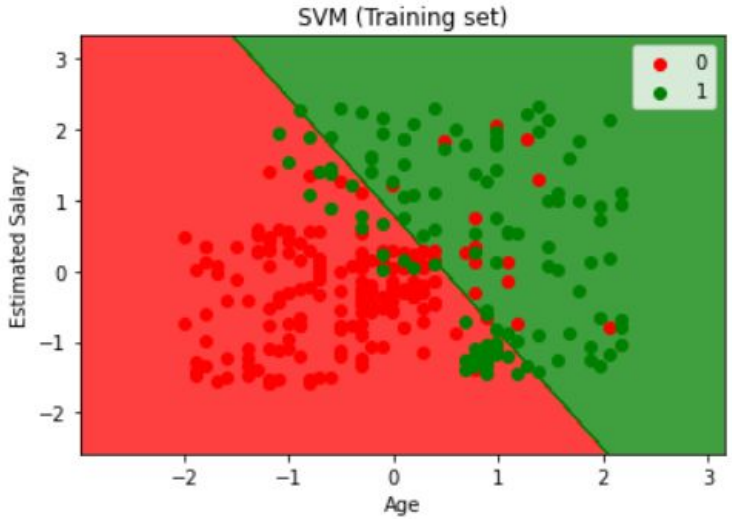


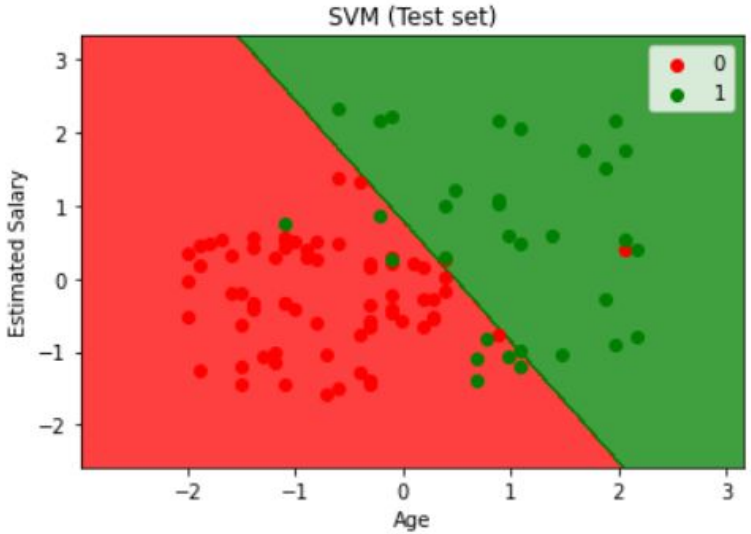


SUPPORT VECTOR MACHINE SNA

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.9
```

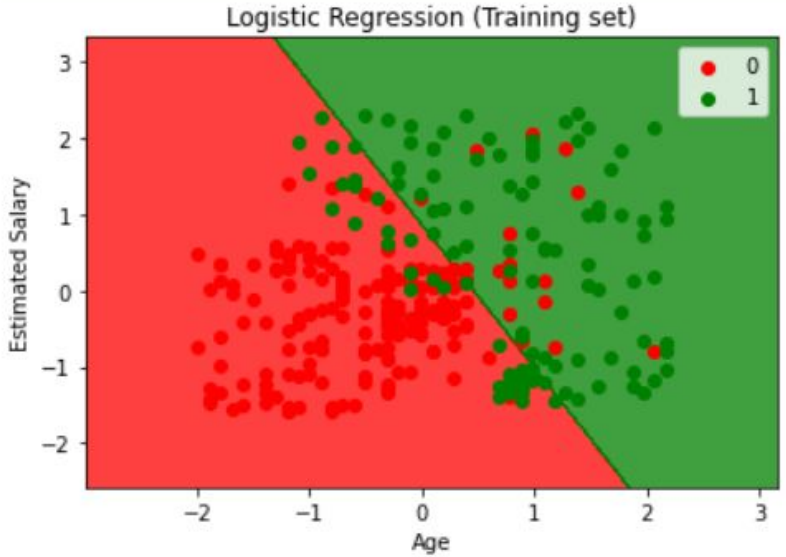


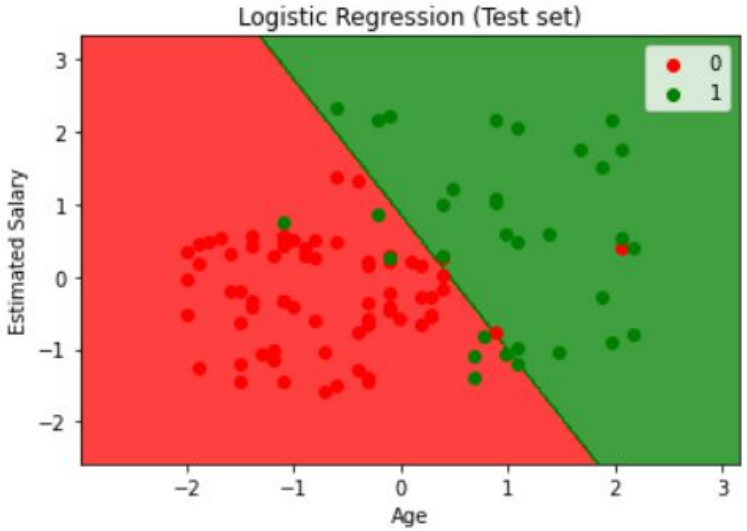


LOGISTIC REGRESSION SNA

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.89
```

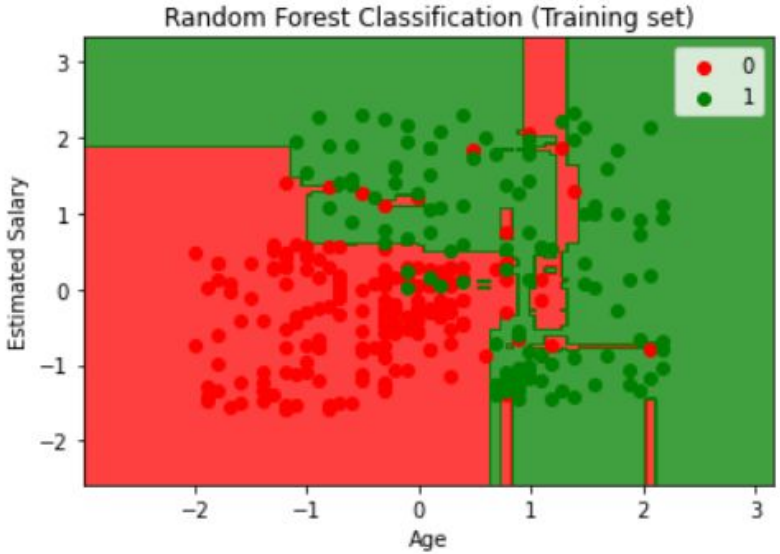


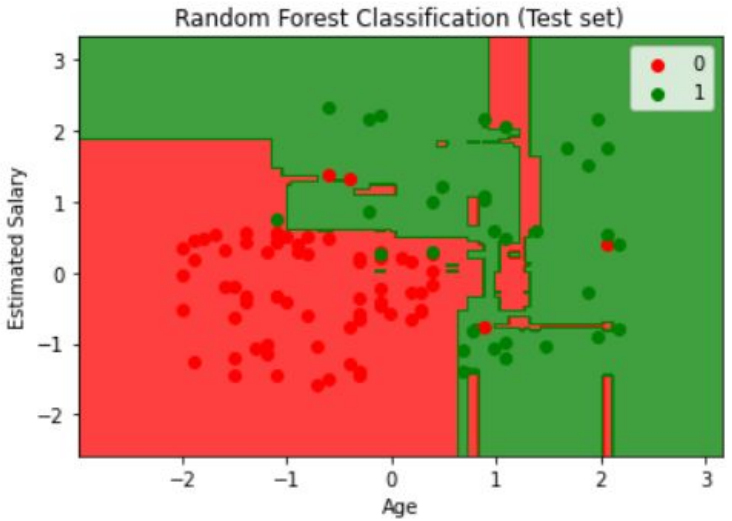


RANDOM FOREST CLASSIFIER SNA

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.91
```





01 CONCLUSION

KNN obtuvo un 0.93 en el accuracy score en SNA. Poniéndolo ligeramente más alto que los demás. Para el Iris.csv, todos los modelos parecen haber tenido la misma exactitud.