Comparativo

Modelos de Clasificación

Emily Soto
Katherine García
Raul Castellanos
Andrea Reyes

Social Network Adds

Matriz de confusión

```
[[62  6]
 [ 3 29]]
accuracy decision tree
0.91
```

```
Confusion Matrix SVM
[[66  2]
 [ 8 24]]
Accuracy
0.9
```

```
[[66  2]
 [18 14]]
acurrancy Bayes:
0.8
```

```
Confusion Matrix
[[64  4]
 [ 3 29]]
Accouracy kernel
0.93
```

```
Random Forest
Confusion Matrix
[[63  5]
 [ 4 28]]
Accuracy
0.91
```

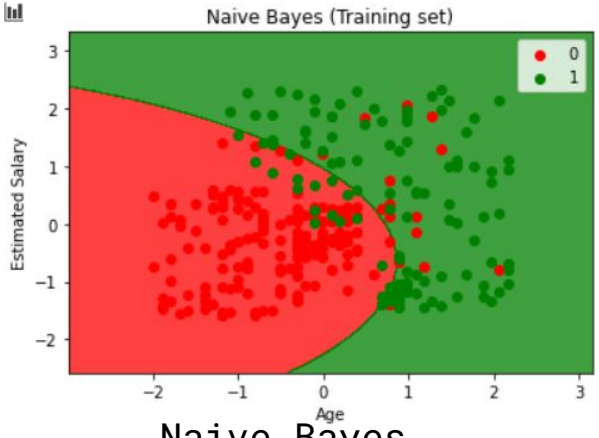
```
[[64  4]
 [ 3 29]]
accuracy KNN
0.74
```

```
[[65  3]
 [ 8 24]]
accuracy Linear regression
0.89
```

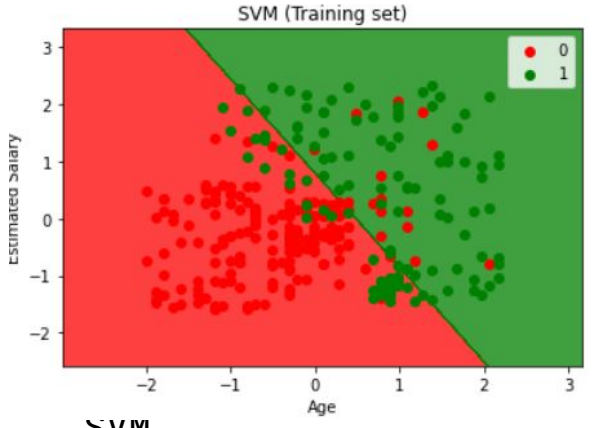
El mejor modelo para esta base de datos fue decision tree y random forest

Training set

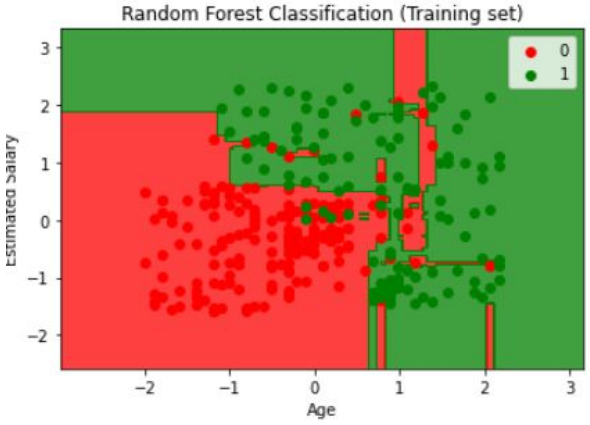
Decision tree



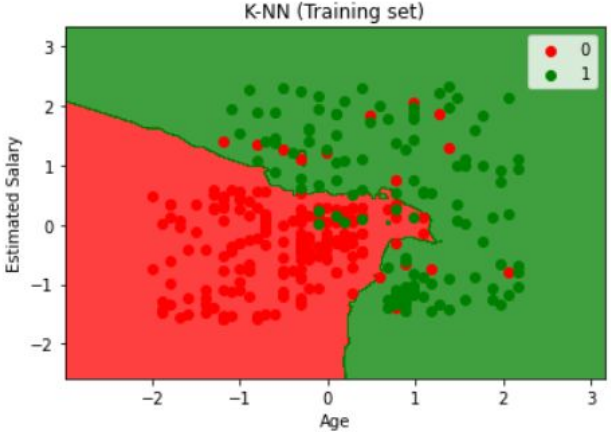
Naive Bayes



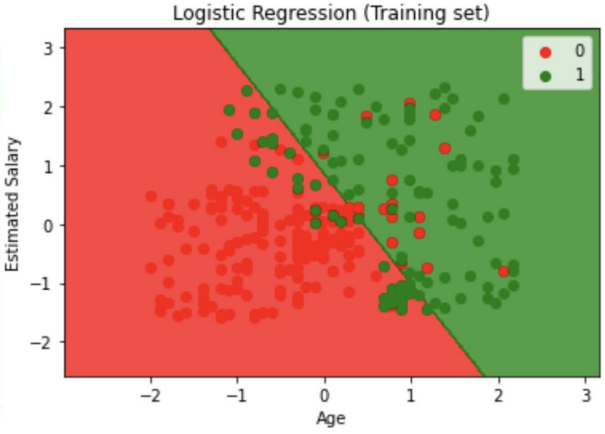
SVM



Random Forest



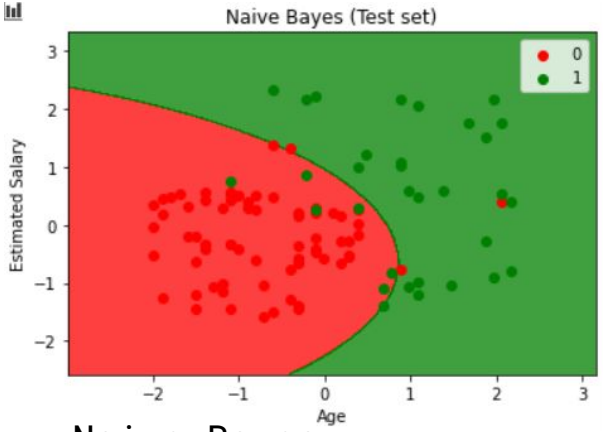
KNN



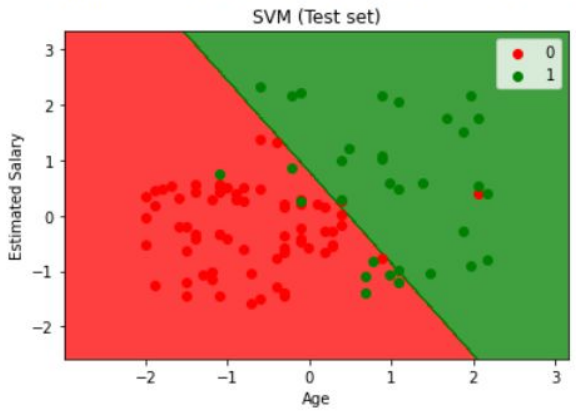
Logistic Regression

Test set

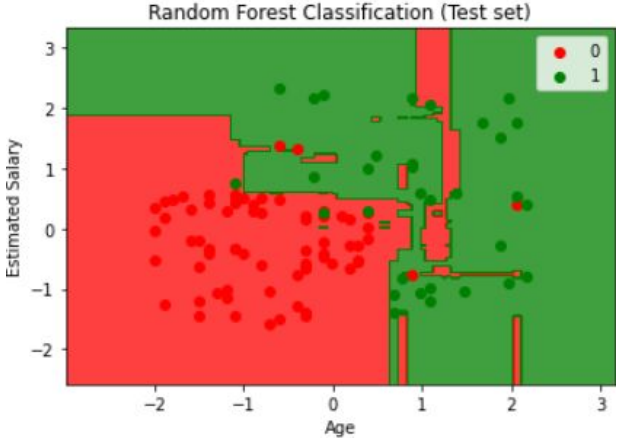
Decision tree



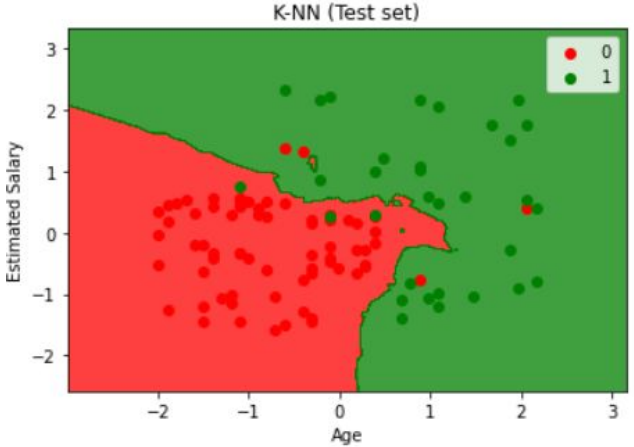
Naive Bayes



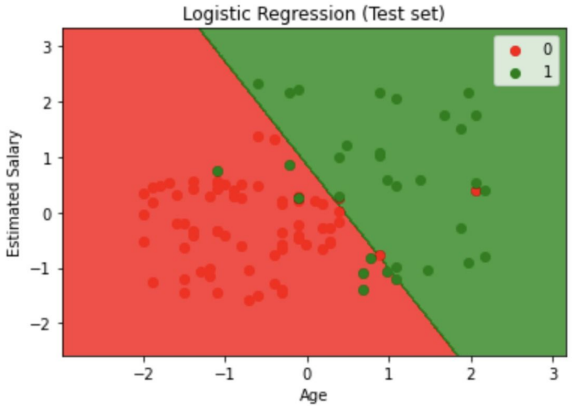
SVM



Random Forest



KNN



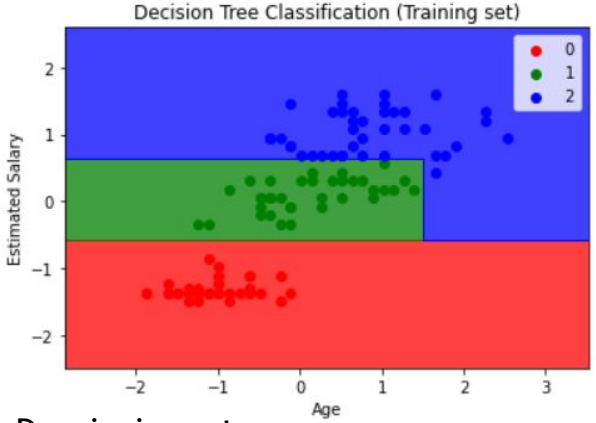
Logistic Regression

Matriz de confusión

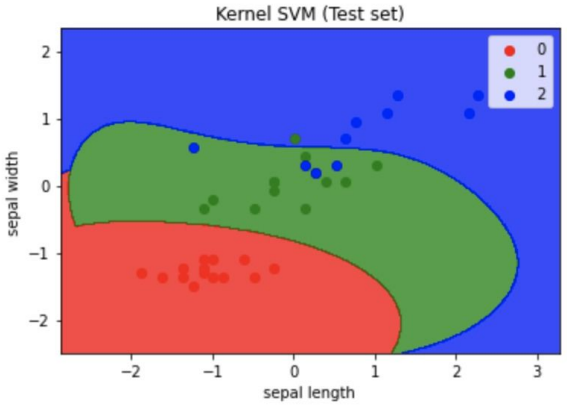
```
[[14  0  0]
 [ 0 13  1]
 [ 0  4  6]]
acurrancy:
0.868421052631579
```

Ya que se utilizó la librería scikit learn para todos los modelos, la matriz de confusión fue la misma en este dataframe, y de igual forma el cálculo de la exactitud. La exactitud fue relativamente buena con un 86.8%

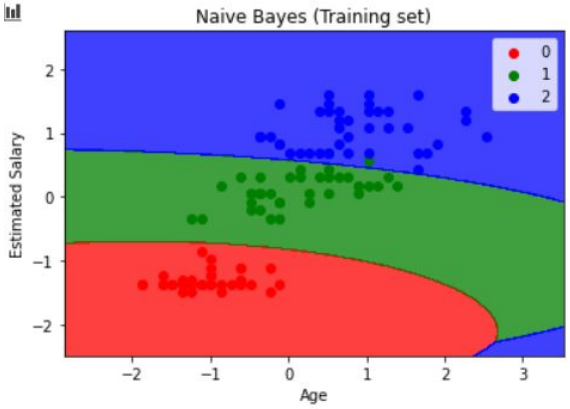
Training set



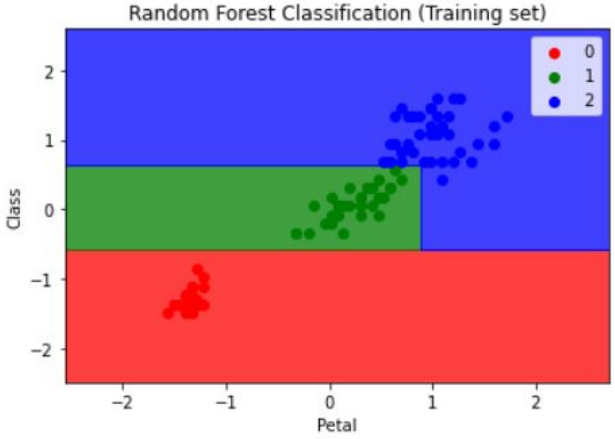
Decision tree



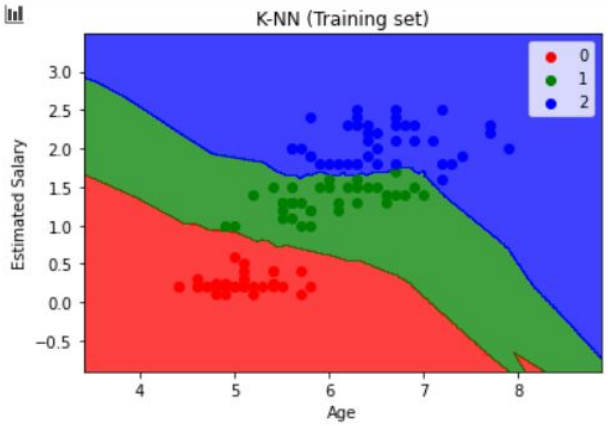
Kernel SVM



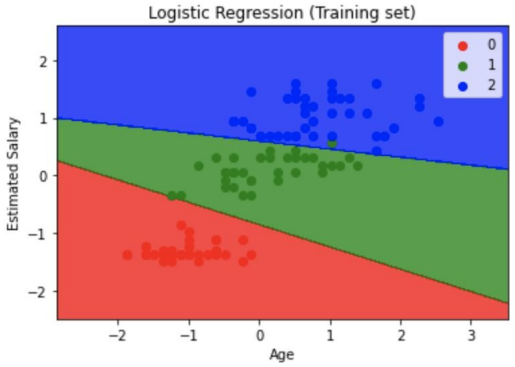
Naive Bayes



Random Forest

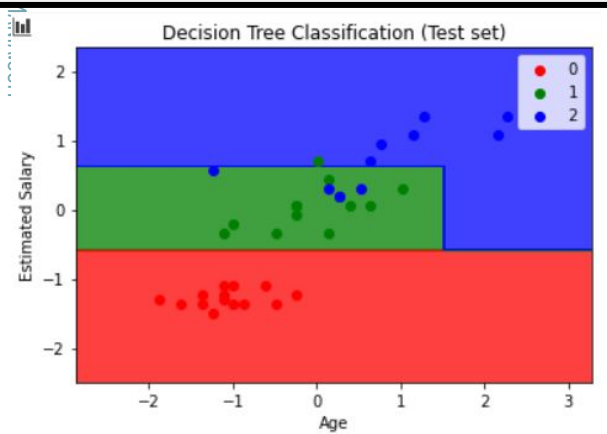


KNN

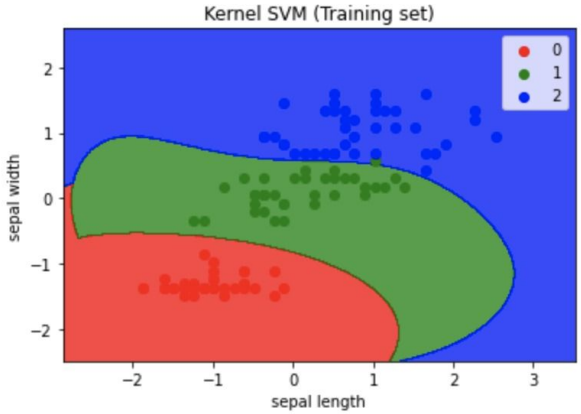


Logistic Regression

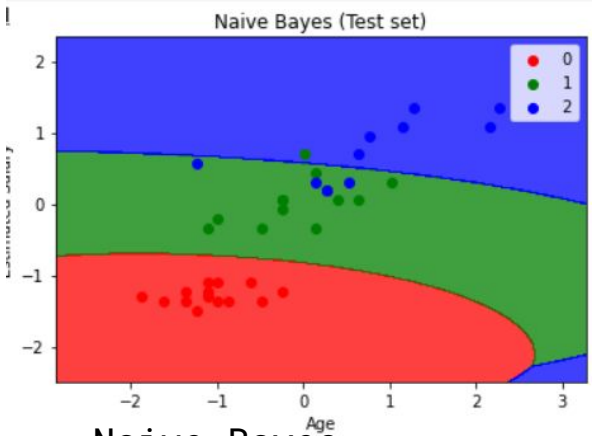
Test set



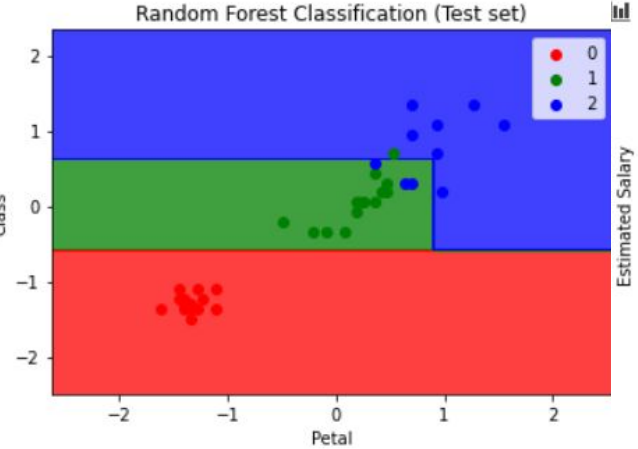
Decision tree



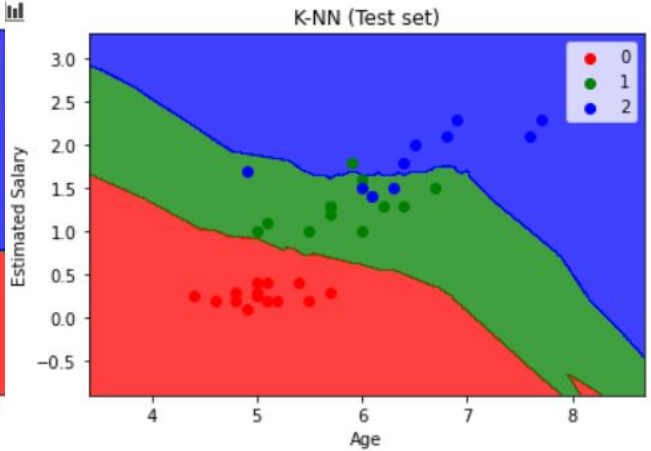
Kernel SVM



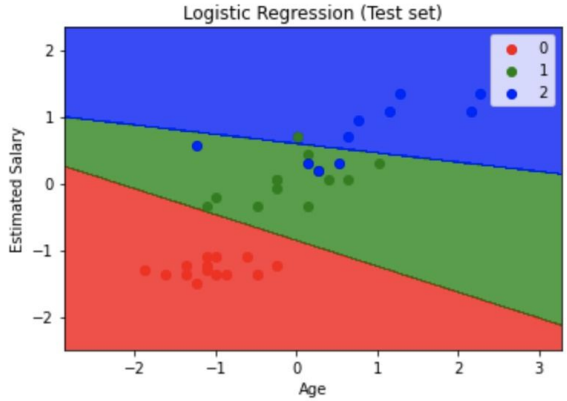
Naive Bayes



Random Forest



KNN



Logistic Regression