Actividad Algoritmos de clasificacion

Ernesto Reynoso Lizárraga A01639915

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from numpy.random.mtrand import logistic
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
import statsmodels.api as sm
from sklearn.metrics import accuracy_score, confusion_matrix
```

```
df = pd.read_csv("/content/drive/MyDrive/Inteligencia Artificial/Iris.csv")
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
df= df.drop('Id',axis=1)
```

```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

```
x.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
y.head()
```

```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

Regresion Logistica

```
logistic = LogisticRegression()
logistic.fit(x_train, y_train)
y_predict=logistic.predict(x_test)
```

```
exactitud = accuracy_score(y_test,y_predict)
print("Exactitud:", exactitud)
confusion=confusion_matrix(y_test,y_predict)
sns.heatmap(confusion, annot = True, cmap='Blues')
class_names = ['Setosa', 'Versicolor','Virginica']
tick_marks = [0.5,1.5,2.5]
plt.xticks(tick_marks,class_names)
```

```
plt.yticks(tick_marks,class_names)
plt.ylabel("Valor verdadero")
plt.xlabel("Prediccion")
```

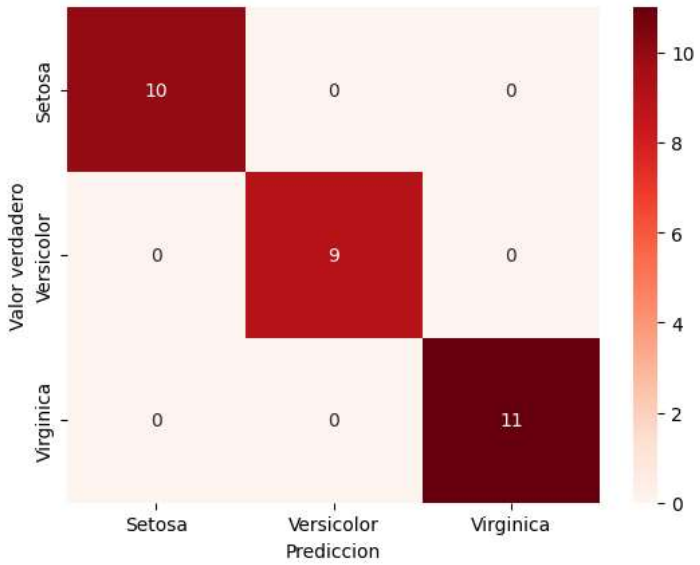
```
Exactitud: 1.0
Text(0.5, 23.52222222222222, 'Prediccion')
```

Bayes

```
nb = GaussianNB()
nb.fit(x_train, y_train)
y_predict=nb.predict(x_test)
```

```
exactitud = accuracy_score(y_test,y_predict)
print("Exactitud:", exactitud)
confusion=confusion_matrix(y_test,y_predict)
sns.heatmap(confusion, annot = True, cmap='Reds')
class_names = ['Setosa', 'Versicolor','Virginica']
tick_marks = [0.5,1.5,2.5]
plt.xticks(tick_marks,class_names)
plt.yticks(tick_marks,class_names)
plt.ylabel("Valor verdadero")
plt.xlabel("Prediccion")
```

```
Exactitud: 1.0
Text(0.5, 23.52222222222222, 'Prediccion')
```

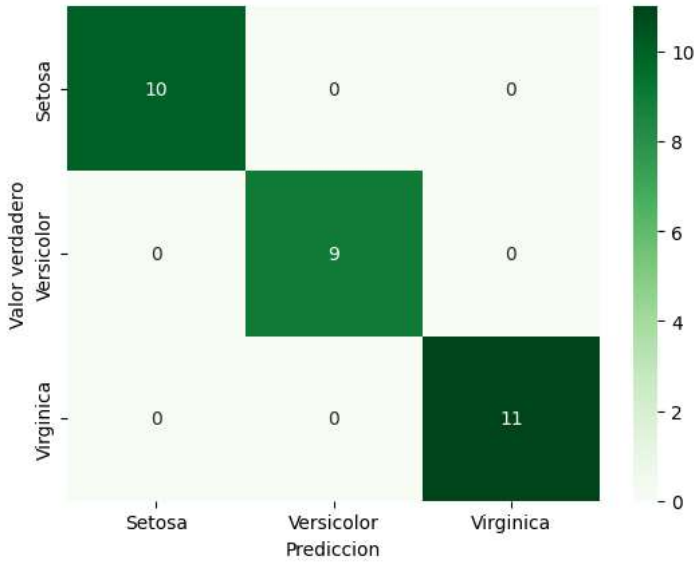


Analisis del Discriminante

```
modelo = LinearDiscriminantAnalysis()
modelo.fit(x_train,y_train)
y_predict = modelo.predict(x_test)
```

```
exactitud = accuracy_score(y_test,y_predict)
print("Exactitud:", exactitud)
confusion=confusion_matrix(y_test,y_predict)
sns.heatmap(confusion, annot = True, cmap='Greens')
class_names = ['Setosa', 'Versicolor','Virginica']
tick_marks = [0.5,1.5,2.5]
plt.xticks(tick_marks,class_names)
plt.yticks(tick_marks,class_names)
plt.ylabel("Valor verdadero")
plt.xlabel("Prediccion")
```

```
Exactitud: 1.0
Text(0.5, 23.52222222222222, 'Prediccion')
```



Al comparar los 3 algoritmos de clasificación podemos concluir que no existen diferencias entre ellos para este caso de datos en especifico, por lo que los 3 tienen la misma efectividad. Esto se puede deber a la poca cantidad de datos en el dataset.