

Reporte sobre el desempeño del modelo.

Miguel Alejandro Salas Reyna A00827219

Para el desarrollo de la solución con framework se importaron las siguientes librerías:

```
import numpy as np
import pandas as pd
import requests
import io
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import preprocessing
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns

columns = ["Class", "Alcohol", "Malic acid", "Ash", "Alcalinity of ash", "Magnesium", "Total
phenols", "Flavanoids", "Nonflavanoid phenols", "Proanthocyanins", "Color
intensity", "Hue", "OD280/OD315 of diluted wines", "Proline"]
url =
"https://raw.githubusercontent.com/miiguelsr/frame/659b263663a68bd9ebef6b142c67f6af61
be0e6a/wine.data" # Make sure the url is the raw version of the file on GitHub
download = requests.get(url).content
df = pd.read_csv(io.StringIO(download.decode('utf-8')), names=columns)
df

X = df[["Alcohol", "Malic acid", "Ash", "Alcalinity of ash", "Magnesium", "Total
phenols", "Flavanoids", "Nonflavanoid phenols", "Proanthocyanins", "Color
intensity", "Hue", "OD280/OD315 of diluted wines", "Proline"]]
y = df.Class
```

Se separó el conjunto de pruebas en test y train, es decir, un conjunto para pruebas y otro para entrenar el modelo.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state = 42)
```

El modelo elegido fue regresión logística.

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
train_accuracy = model.score(X_train, y_train)
test_accuracy = model.score(X_test, y_test)
```

```

train_acc = model.score(X_train, y_train)
print("The Accuracy for Training Set is {}".format(train_acc*100))

y_pred = model.predict(X_test)
test_acc = accuracy_score(y_test, y_pred)
print("The Accuracy for Test Set is {}".format(test_acc*100))

cm=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(12,6))
plt.title("Confusion Matrix")
sns.heatmap(cm, annot=True,fmt='d', cmap='Blues')
plt.ylabel("Actual Values")
plt.xlabel("Predicted Values")
# plt.savefig('confusion_matrix.png')
plt.show()

print(classification_report(y_test,y_pred))

```

Al correr el modelo nos da un Accuracy de 95% para entrenamiento y en el entrenamiento de 97%. Lo cual puede resultar en dos opciones: un sobreentrenamiento, es decir, el modelo se memorizó los resultados y simplemente los arroja de memoria o el modelo tiene un buen desarrollo. Si pudieramos tener un set de valoración podríamos encontrar la respuesta a esta cuestión. Por el momento, este modelo resultó ser mejor que desarrollar el algoritmo sin framework.

```

The Accuracy for Training Set is 95.48872180451127
The Accuracy for Test Set is 97.77777777777777
precision  recall  f1-score  support

```

```

1      1.00    0.93    0.97     15
2      0.95    1.00    0.97     18
3      1.00    1.00    1.00     12

```

```

accuracy                0.98     45
macro avg      0.98    0.98    0.98     45
weighted avg   0.98    0.98    0.98     45

```

