colab.research.google.com/drive/1fX--rZi5XWwW9rwuUHxtUx5d0LMWXUL9

mail YouTube Maps Chrome keeps pass...

Assignment5 -700755784.ipynb ☆

File Edit View Insert Runtime Tools Help

+ Code + Text

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report

# Load the Glass dataset (assuming you have it as a CSV file)
data = pd.read_csv('glass.csv')

# Split the dataset into features (X) and the target variable (y)
X = data.drop('Type', axis=1)
y = data['Type']

# Split the data into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize and train a Gaussian Naïve Bayes classifier
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

# Make predictions on the test data
y_pred = nb_classifier.predict(X_test)

# Evaluate the model
accuracy = nb_classifier.score(X_test, y_test)
classification_rep = classification_report(y_test, y_pred)

# Print the results
print(f'Accuracy: {accuracy:.2f}')
```

✓ Connected to Python 3 Google Compute Engine backend

Assignment5 -700755784.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# Initialize and train a Gaussian Naïve Bayes classifier
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

# Make predictions on the test data
y_pred = nb_classifier.predict(X_test)

# Evaluate the model
accuracy = nb_classifier.score(X_test, y_test)
classification_rep = classification_report(y_test, y_pred)

# Print the results
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:\n', classification_rep)
```

```
Accuracy: 0.31
Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00        19
           2       0.40      0.17      0.24        23
           3       0.08      0.75      0.15         4
           5       0.33      0.17      0.22         6
           6       0.75      1.00      0.86         3
           7       0.90      0.90      0.90        10

    accuracy                           0.31        65
   macro avg       0.41      0.50      0.40        65
weighted avg       0.35      0.31      0.29        65
```

Connected to Python 3 Google Compute Engine backend

Assignment5 -700755784.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report

# Load the Glass dataset (assuming you have it as a CSV file)
data = pd.read_csv('glass.csv')

# Split the dataset into features (X) and the target variable (y)
X = data.drop('Type', axis=1)
y = data['Type']

# Split the data into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize and train a Linear SVM classifier
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train, y_train)

# Make predictions on the test data
y_pred = svm_classifier.predict(X_test)

# Evaluate the model
accuracy = svm_classifier.score(X_test, y_test)
classification_rep = classification_report(y_test, y_pred)

# Print the results
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:\n', classification_rep)
```

Connected to Python 3 Google Compute Engine backend

Assignment5 -700755784.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

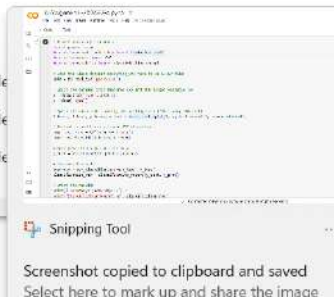
```
# Print the results
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:\n', classification_rep)
```

```
Accuracy: 0.68
Classification Report:
              precision    recall  f1-score   support

           1       0.65      0.79      0.71        19
           2       0.59      0.70      0.64        23
           3       0.00      0.00      0.00         4
           5       0.75      0.50      0.60         6
           6       0.50      0.33      0.40         3
           7       1.00      0.90      0.95        10

    accuracy                           0.68        65
   macro avg       0.58      0.54      0.55        65
weighted avg       0.65      0.68      0.65        65

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-de
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-de
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-de
  _warn_prf(average, modifier, msg_start, len(result))
```



Snipping Tool

Screenshot copied to clipboard and saved

Select here to mark up and share the image