ASSIGNMENT 3

NAME : PRANAV ANAND LEELARAM
REG : 21BCE0464
EMAIL : pranavanand.leelaram2021@vitstudent.ac.in
CAMPUS : VIT VELORE

INSTALLATION :

```
pip install pandas seaborn scikit-learn matplotlib
```

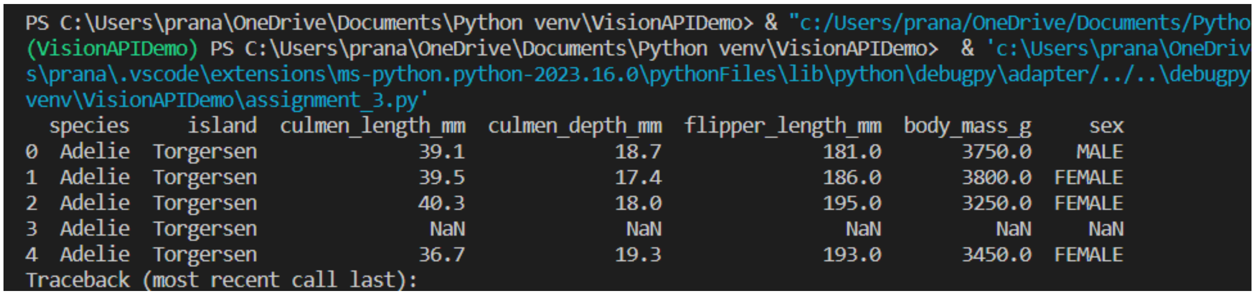
CODE :

```
# Import necessary libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

# Load the dataset
penguins = pd.read_csv("C:/Users/prana/Downloads/penguins_size.csv")

# Check the first few rows of the dataset
print(penguins.head())

# Perform data preprocessing
# Handle missing values if any
```

```
penguins.dropna(inplace=True)

# Encode categorical variables (Species and Sex) using Label Encoding
label_encoder = LabelEncoder()
penguins['Species'] = label_encoder.fit_transform(penguins['Species'])
penguins['Sex'] = label_encoder.fit_transform(penguins['Sex'])

# Visualizations
# Pairplot to visualize relationships between features
sns.pairplot(penguins, hue='Species')
plt.show()

# Correlation heatmap
correlation_matrix = penguins.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# Clustering
# Select relevant features for clustering
X = penguins[['culmen_length_mm', 'culmen_depth_mm', 'flipper_length_mm', 'body_mass_g']]

# Perform K-Means clustering to group penguins based on their physical characteristics
kmeans = KMeans(n_clusters=3, random_state=42)
penguins['Cluster'] = kmeans.fit_predict(X)

# Visualize the clusters
sns.scatterplot(x='culmen_length_mm', y='culmen_depth_mm', hue='Cluster', data=penguins,
palette='Set1')
plt.title('K-Means Clustering')
plt.show()

# Classification
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, penguins['Species'], test_size=0.2, random_state=42)

# Train a Random Forest classifier
rf_classifier = RandomForestClassifier(random_state=42)
rf_classifier.fit(X_train, y_train)
rf_predictions = rf_classifier.predict(X_test)

# Evaluate the classifier
print("Random Forest Classification Report:")
print(classification_report(y_test, rf_predictions))

# Train a Support Vector Machine (SVM) classifier
svm_classifier = SVC(random_state=42)
svm_classifier.fit(X_train, y_train)
```

```
svm_predictions = svm_classifier.predict(X_test)

# Evaluate the classifier
print("SVM Classification Report:")
print(classification_report(y_test, svm_predictions))
```

OUTPUT :



```
PS C:\Users\prana\OneDrive\Documents\Python venv\VisionAPIDemo> & "c:/Users/prana/OneDrive/Documents/Pytho
(VisionAPIDemo) PS C:\Users\prana\OneDrive\Documents\Python venv\VisionAPIDemo>  & 'c:\Users\prana\OneDriv
s\prana\.vscode\extensions\ms-python.python-2023.16.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy
venv\VisionAPIDemo\assignment_3.py'
  species     island  culmen_length_mm  culmen_depth_mm  flipper_length_mm  body_mass_g     sex
0  Adelie  Torgersen              39.1             18.7              181.0       3750.0    MALE
1  Adelie  Torgersen              39.5             17.4              186.0       3800.0  FEMALE
2  Adelie  Torgersen              40.3             18.0              195.0       3250.0  FEMALE
3  Adelie  Torgersen               NaN              NaN                NaN          NaN     NaN
4  Adelie  Torgersen              36.7             19.3              193.0       3450.0  FEMALE
Traceback (most recent call last):
```