Mahati Akella 22070521027

```
#PRINCIPAL COMPONENT ANALYSIS AND DATA Visualisation
```

```
#TO PERFORM DIMENSIONALITY REDUCTION USING PCA AND CREATE APPROPRIATE PLOTS TO ANALYZE COMPONETS  ON IRIS DATASET
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

```
# Load dataset
df = pd.read_csv('Iris.csv')

# Display first few rows
df.head()
```

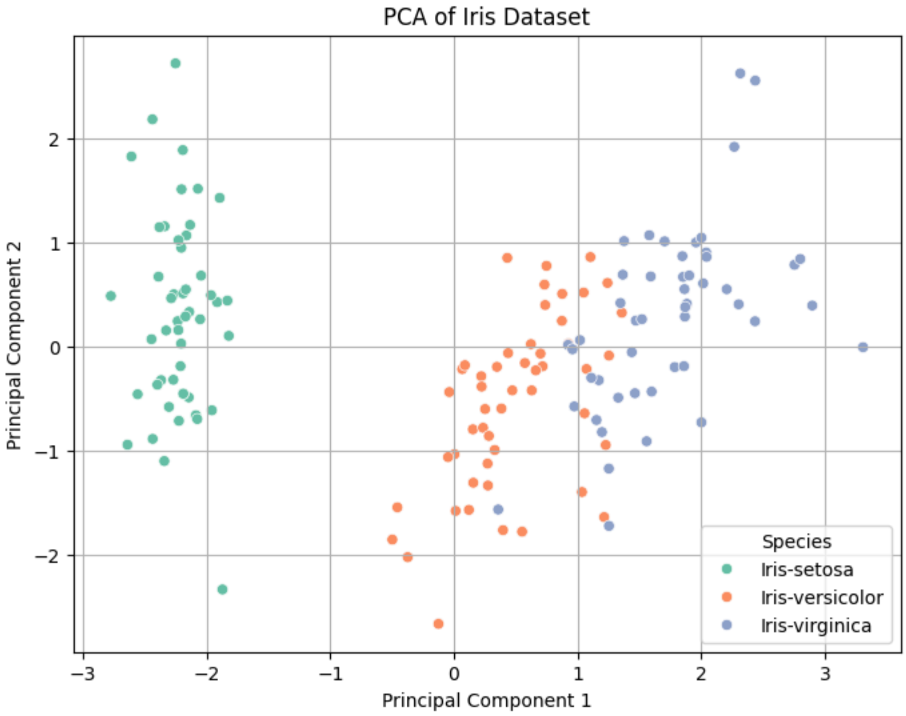
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Drop unnecessary columns like Id (if present)
if 'Id' in df.columns:
    df = df.drop(['Id'], axis=1)

# Separate features and target
features = df.columns[:-1]  # All columns except 'Species'
X = df[features]
y = df['Species']

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

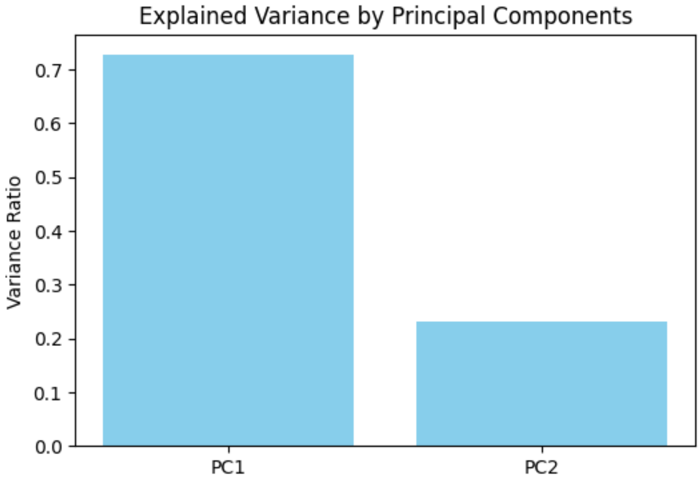
```
# Initialize PCA
pca = PCA(n_components=2)  # Reduce to 2 components
X_pca = pca.fit_transform(X_scaled)

# Create DataFrame for PCA result
pca_df = pd.DataFrame(data=X_pca, columns=['PC1', 'PC2'])
pca_df['Species'] = y
```

```
plt.figure(figsize=(8,6))
sns.scatterplot(data=pca_df, x='PC1', y='PC2', hue='Species', palette='Set2')
plt.title('PCA of Iris Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.grid(True)
plt.show()
```

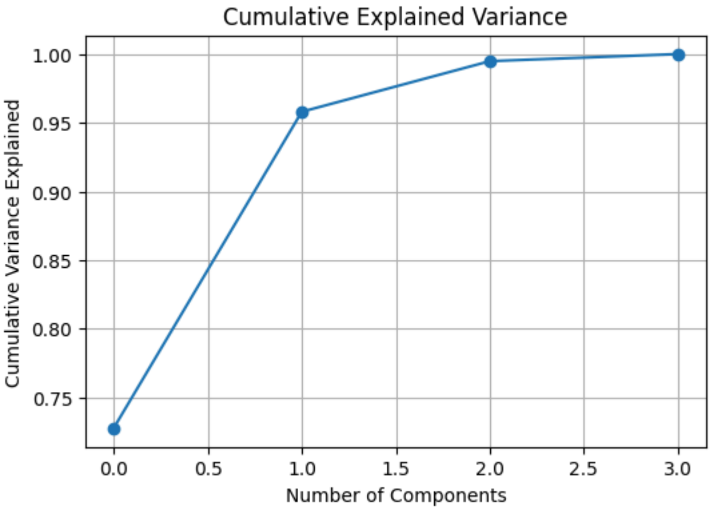


```
print("Explained variance by each component:")
print(pca.explained_variance_ratio_)

# Bar plot of explained variance
plt.figure(figsize=(6,4))
plt.bar(['PC1', 'PC2'], pca.explained_variance_ratio_, color='skyblue')
plt.title('Explained Variance by Principal Components')
plt.ylabel('Variance Ratio')
plt.show()
```

```
Explained variance by each component:
[0.72770452 0.23030523]
```



```
# Try PCA with all components to see cumulative explained variance
pca_full = PCA()
pca_full.fit(X_scaled)

# Plot cumulative explained variance
plt.figure(figsize=(6,4))
plt.plot(np.cumsum(pca_full.explained_variance_ratio_), marker='o')
plt.title('Cumulative Explained Variance')
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Variance Explained')
plt.grid(True)
plt.show()
```


Cumulative Explained Variance
Cumulative Variance Explained
1.00
0.95
0.90
0.85
0.80
0.75
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Number of Components