

**UNIVERSITI KUALA LUMPUR**

**ASSESSMENT BRIEF**



|  |  |
| --- | --- |
|  | **COURSE DETAILS** |
| **INSTITUTE** | UniKL BRITISH MALAYSIAN INSTITUTE |
| **COURSE NAME** | BIG DATA ANALYTICS |
| **COURSE CODE** | BEB 43403 |
| **COURSE LEADER** | MUHD KHAIRULZAMAN ABDUL KADIR |
| **LECTURER** | MUHD KHAIRULZAMAN ABDUL KADIR |
| **SEMESTER & YEAR** | OCTOBER 2023 |

|  |  |
| --- | --- |
|  | **ASSESSMENT DETAILS** |
| **TITLE/NAME** | LAB ASSIGNMENT |
| **WEIGHTING** | 25% |
| **DATE/DEADLINE** | 5 / 1 / 2024 |
| **COURSE LEARNING**  **OUTCOME(S)** | CLO1: Build meaningful predictive models using various data analytic tools  (P7, PLO4) |
| **INSTRUCTIONS** | Perform the following tasks:   1. Answer all questions. 2. All answers must be in English language only. |

|  |  |  |
| --- | --- | --- |
| **Student Name:**  MUHAMMAD IKHWAN SYAFIQ BIN NORSHAM  MUHAMMAD WAIZ BIN NOR KAMAL | **ID:**  51221221125  51221221053 | **Group:**  **L01-B02** |
| **Assessor’s Comment:** |  | **Marks:** |

|  |  |  |
| --- | --- | --- |
| **Verified by: Course Leader [MUZA] Prepared by: [MUZA]**    I hereby declare that all my team members have agreed with this assessment. All team members are certain that this assessment complies with the Course Syllabus.  Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date : \_\_\_\_\_\_18 / 12 / 2023\_\_\_\_\_\_\_\_\_\_\_\_ | **QSC format verification** | **PC/HOS content validation**      Nor Amalia Sapiee  19/12/2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TASK NO** | **CLO** | **MARKING SCHEME** |  | **MARKS** |
| 1 | 1 | **Part B** |  |  |
|  |  | Question a) |  | *5* |
|  |  | Question b) |  | 5 |
| 2 | 1 | **Part C** |  |  |
|  |  | Question a) |  | 2 |
|  |  | Question b) |  | *3* |
|  |  | Question c) |  | *8* |
|  |  | Question d) |  | *8* |
|  |  | Question e) |  | *6* |
|  |  | Question f) |  | 8 |
|  |  | Question g) |  | 8 |
|  |  | Question h) |  | 8 |
|  |  | Question i) |  | 7 |
|  |  | Question j) |  | *8* |
|  |  | Question k) |  | *8* |
|  |  | Question l) |  | *8* |
|  |  | Question m) |  | *8* |
|  |  |  | **TOTAL** | **100** |

**INFORMATION ON SK\_SP-TA FOR COURSE**

|  |  |  |
| --- | --- | --- |
| **Course Code & Name** | **:** | BEB43403 **& BIG DATA ANALYTICS** |
| **PLOs** | **:** | **4** |

**Please tick (**  **) in the box provided.**

|  |  |  |
| --- | --- | --- |
| ***Knowledge Profiles (SK)*** A programme that builds this type of knowledge and develops the attributes listed below is typically achieved in 4 years of study | | |
| **SK1** | A systematic, theory-based understanding of the natural sciences applicable to the sub-discipline |  |
| **SK2** | Conceptually-based mathematics, numerical analysis, statistics and aspects of computer and information science to support analysis and use of models applicable to the sub-discipline |  |
| **SK3** | A systematic , theory-based formulation of engineering fundamentals required in an accepted subdiscipline |  |
| **SK4** | Engineering specialist knowledge that provides theoretical frameworks and bodies of |  |
| knowledge for an accepted sub-discipline |
| **SK5** | Knowledge that supports engineering design using the technologies of a practice area |  |
| **SK6** | Knowledge of engineering technologies applicable in the sub-discipline |  |
| **SK7** | Comprehension of the role of technology in society and identified issues in applying |  |
| engineering technology: ethics and impacts: economic, social, environmental and |
| sustainability |
| **SK8** | Engagement with the technological literature of the discipline |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Definition of Broadly-Defined Problem Solving (SP)*** | | | |
| **No.** | **Attribute** | **Broadly-defined Engineering Problems have characteristic SP1 and some or all of SP2 to SP7:** | |
| **SP1** | **Depth of Knowledge Required** | Cannot be resolved without engineering knowledge at the level of one or more of SK 4, SK5, and SK6 supported by SK3 with a strong emphasis on the application of developed technology |  |
| **SP2** | **Range of conflicting requirements** | Involve a variety of factors which may impose conflicting constraints. |  |
| **SP3** | **Depth of analysis required** | Can be solved by application of well-proven analysis techniques |  |
| **SP4** | **Familiarity of issues** | Belong to families of familiar problems which are solved in well-accepted ways |  |
| **SP5** | **Extent of applicable codes** | May be partially outside those encompassed by standards or codes of practice |  |
| **SP6** | **Extent of stakeholder involvement and level of conflicting requirements** | Involve several groups of stakeholders with differing and occasionally conflicting needs |  |
| **SP7** | **Interdependence** | Are parts of, or systems within complex engineering problems |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Range of Engineering Activities (TA)*** | | | |
| **No.** | **Attribute** | **Broadly-defined activities** | |
| **TA1** | **Range of resources** | Involve a variety of resources (and for this purposes resources includes people, money, equipment, materials, information and technologies) |  |
| **TA2** | **Level of interactions** | Require resolution of occasional interactions between technical, engineering and other issues, of which few are conflicting |  |
| **TA3** | **Innovation** | Involve the use of new materials, techniques or processes in non-standard ways |  |
| **TA4** | **Consequences to society and the environment** | Have reasonably predictable consequences that are most important locally, but may extend more widely |  |
| **TA5** | **Familiarity** | Require a knowledge of normal operating procedures and processes |  |

# PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA) is a powerful method for reducing the complexity of high-dimensional data while preserving its key information. It works by identifying the most important underlying patterns in the data, allowing us to focus on a smaller set of variables that have the greatest impact on the overall analysis. In this lab, we will explore how PCA can be applied to a certain dataset and how it can help us extract the most relevant information from the data for deeper insights.

A – Tools needed for Python.

1. Please check if your Python have installed NumPy, Matplotlib, SciPy, Pandas and Scikit-learn.
2. If not yet install, please ensure to install all these tools or library before starting the lab assignment.

## B – Understanding PCA

1. Briefly explain the concept of PCA and its applications.

Firstly, Transforming the Data: Data neatly sorting and organizing those variables based on their main directions. Secondly, Capturing Information: Each data point can be projected onto these principal components, resulting in a set of scores on each component.

Application PCA for data visualization is to helps visualize high-dimensional data by reducing it to a smaller number of dimensions (typically two or three). This aids in understanding the underlying structure and patterns in the data. Next, for machine learning is to reduces dimensions, focusing on the most relevant bits, making algorithms learn faster and predict with greater accuracy.

* 1. marks)

1. Discuss the importance of eigenvalues and eigenvectors in PCA.

Eigenvalues and eigenvectors play a crucial role in PCA, acting as the engine behind its magic of dimensionality reduction and pattern identification. Eigenvalues they help to quantify the amount of variance captured by each principal component. Eigenvectors they define the direction of each principal component. These two methods very important together. They provide a roadmap for navigating the high-dimensional data, highlighting the most important paths, and discarding irrelevant detours. Both are crucial for understanding and utilizing PCA effectively, from compressing data to revealing hidden patterns and empowering machine learning algorithms.

* 1. marks)

## C – PCA Lab Tutorial

Question 1

1. Download iris.csv dataset in Microsoft TEAMS and give the dataset size in column and row in below box. At the same time, state how many parameters used, and details parameter used. (2 marks)

Data size: 149

Parameter / feature size: 5

Parameter / feature details: ['sepal.length', 'sepal.width', 'petal.length', 'petal.width', 'variety']

1. After that, load all the dataset iris.csv in Python by using Pandas ‘pd.read\_csv’ or any code suitable and write the coding in the box below: - (3 marks)

Code:   
# Load the dataset

iris\_data = pd.read\_csv('/content/iris.csv')  
  
# Understanding the data set

numSamples, numFeatures = iris\_data.shape #used to extract the number of samples (rows) and the number of features (columns)

print("Number of samples:", numSamples)

print("Number of features:", numFeatures)

#Target column is present in the dataset, you can print the unique values in the target column which is my target

print ("Target names:", iris\_data['variety'].unique())  
  
# Display the size of the dataset (rows, columns)

data\_size = iris\_data.shape

# Display the number of parameters/features and their details

parameter\_size = iris\_data.shape[1] # Number of columns

parameter\_details = iris\_data.columns.tolist() # List of column names

# Print the results

print("Data size:", data\_size)

print("Parameter / feature size:", parameter\_size)

print("Parameter / feature details:", parameter\_details)  
  
# Display the entire data

print(iris\_data)

1. Get the value of x and y for the iris.csv dataset by using Pandas ‘dataframes’ or any code suitable. Write the code and the results in below box.

Code:   
  
x = iris\_data.iloc[:, :-1] # Features (all columns except the last one)

y = iris\_data.iloc[:, -1] # Labels (the last column)

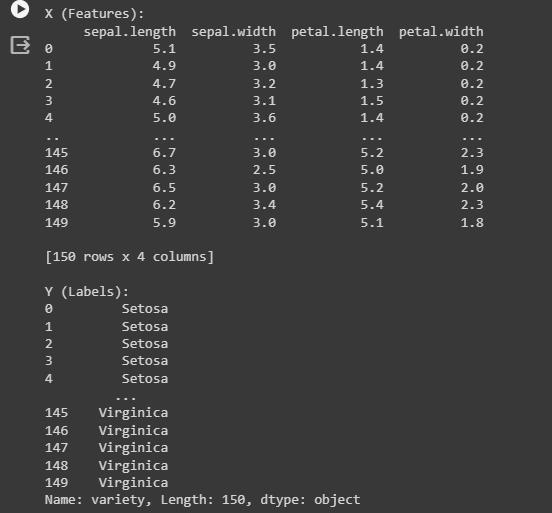
# Displaying x and y

print("X (Features):")

print(x)

print("\nY (Labels):")

print(y)

Result of x and y: 

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

* 1. marks)

1. Use ‘StandardScaler’ in Scikit-learn tools to scale the data in proper scale to standardize all the data available in iris.csv or any code suitable. Write the code and the ‘dataframes’ in variable explorer below.

Code:

scaler = StandardScaler() #preprocessing technique commonly used in machine learning to standardize the features of a dataset.

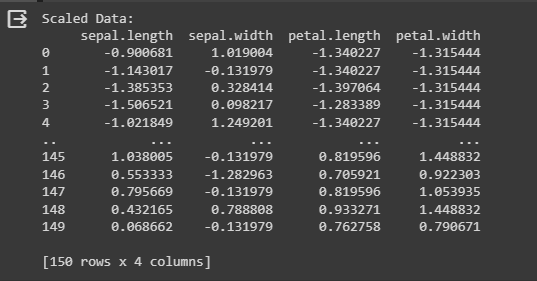
scaled\_data = scaler.fit\_transform(x) #The fit method is used to compute the mean and standard deviation of each feature in the dataset

# Displaying the scaled data

scaled\_data\_df = pd.DataFrame(scaled\_data, columns=x.columns)

print("Scaled Data:")

print(scaled\_data\_df)

Dataframes: 

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

* 1. marks)

1. Next, use the Numpy mean function to find the mean for the whole data available in iris.csv. Write the code and the mean results in below box.

Code:

# Calculate the mean from the scaled data

scaled\_data\_mean = np.mean(scaled\_data, axis=0)

# Displaying the mean results based on the StandardScaler

print("Mean Result based on StandardScaler:")

print(scaled\_data\_mean)

Mean result: 

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

(6 marks)

1. Subtract the mean with all the data in the iris.csv dataset and give the new table for the data. Write the code in below box.

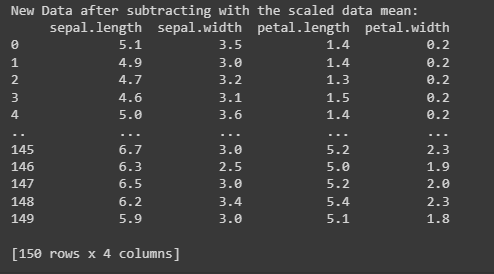
Code:   
  
 # Subtract the scaled data mean from each row in the original data

new\_data = x - scaled\_data\_mean

# Displaying the new\_data

print("New Data after subtracting with the scaled data mean:")

print(new\_data)

NewData after subtracting with the means: 

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

* 1. marks)

1. Calculate the covariance matrix for the iris.csv dataset by using Numpy cov.

Write the code in below box and the covariance matrix data in the box below.

Code:

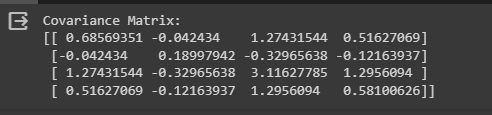
# new\_data is the modified data obtained by subtracting scaled\_data\_mean

covariance\_matrix = np.cov(new\_data, rowvar=False)

# Displaying the covariance matrix

print("Covariance Matrix:")

print(covariance\_matrix)

Covariance matrix: 

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

* 1. marks)

1. Calculate the eigenvector and eigenvalues of the covariance matrix in g) by using Numpy (linalg.eig) or any other suitable code. Write the code and its results in below box.

Code:

eigenvalues, eigenvectors = np.linalg.eig(covariance\_matrix)

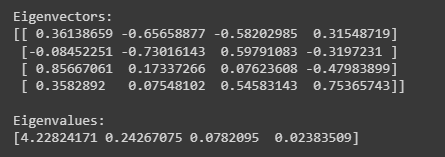
# Displaying the eigenvectors and eigenvalues

print("Eigenvectors:")

print(eigenvectors)

print("\nEigenvalues:")

print(eigenvalues)

Eigenvector & Eigenvalues: 

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

* 1. marks)

1. Use PCA code to get the best components for the dataset given. Write the code and the results in below box.

Code:   
  
# Initialize PCA with the number of components you want to retain

# For example, if you want to retain 2 components, set n\_components=2

pca = PCA(n\_components=2)

# Fit and transform the data using PCA

pca\_result = pca.fit\_transform(new\_data)

# Display the explained variance ratio

print("Explained Variance Ratio:")

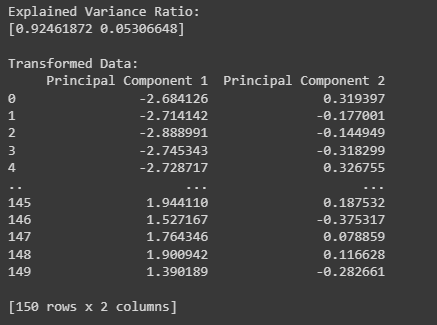
print(pca.explained\_variance\_ratio\_)

# Display the transformed data

pca\_df = pd.DataFrame(data=pca\_result, columns=['Principal Component 1', 'Principal Component 2'])

print("\nTransformed Data:")

print(pca\_df)

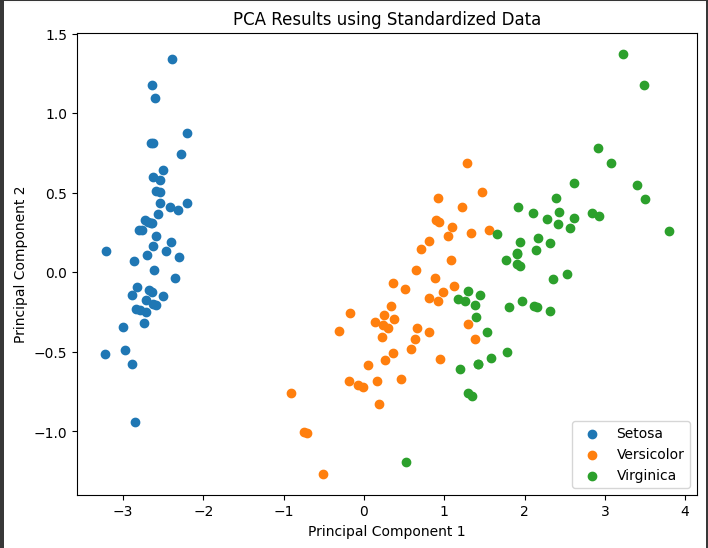


Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

(7 marks)

1. Plot all the PCA results and put in below box to see the features comparison.

Plot result: 

Code:  
# Concatenate the transformed data with the target variable

pca\_df\_with\_target = pd.concat([pca\_df, iris\_data['variety']], axis=1)

# Create scatter plot

plt.figure(figsize=(8, 6))

for variety in iris\_data['variety'].unique():

subset = pca\_df\_with\_target[pca\_df\_with\_target['variety'] == variety]

plt.scatter(subset['Principal Component 1'], subset['Principal Component 2'], label=variety)

# Add labels and title

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('PCA Results using Standardized Data')

plt.legend()

plt.show()

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

* 1. marks)

1. Analysis from the plot result obtain.

The task given is to analyse the PCA plot using iris data to visualize standardized data, here are the observations and views, found the plot reveals a clear separation between the three Iris species which is Setosa forms a distinct cluster on the left side of the plot, well-separated from the other two species. Versicolor and Virginica exhibit some overlap, but they still form distinguishable clusters. The first two principal components form a large proportion of the variance in the original data. This suggests that PCA has effectively captured the most important information, enabling visualization and analysis in a lower-dimensional space.

For the interpretation of principal components, first for PCA 1, the axis captures variations in petal length and width, as these features differentiate the species well. Then for PCA 2, the axis represents variations in sepal length and width, contributing to the separation of Setosa from the other species. The distinct clusters observed suggest that clustering algorithms could effectively group the Iris species based on their features. Petal length and width play a more significant role in distinguishing the species, as they contribute more to PC1, which exhibits the most separation.

Overall, the PCA plot effectively visualizes the relationships within the Iris dataset, revealing clear separation between species and suggesting potential for clustering and feature importance analysis.

Verify with your lecturer.

|  |  |
| --- | --- |
| Verification: |  |

* 1. marks)

1. Give and explain another 2 techniques that can be used for big data analytics.

Firstly, among other techniques that can be used are K-Means Clustering which is can grouped the similar data points into distinct clusters like putting many subjects or variables in separate boxes. This unsupervised technique helps discover hidden patterns and segment data based on inherent similarities. Benefits for K-means Clustering are they efficiently handles large datasets and enables exploratory data analysis and anomaly detection. Next, Apache Spark MLlib offers a robust collection of algorithms pre-optimized for distributed processing on big data platforms like Apache Spark. This allows you to train and deploy powerful models for diverse tasks. Benefits for Apache Spark are scalable and efficient for large datasets and useful for predictive analytics, recommendation systems, and anomaly detection.

* 1. marks)

1. Conclusion on the lab done.

In conclusion, what can be learned from the PCA assignment by applying PCA to the Iris dataset are PCA can visualize the relationships between the four Iris features (petal and sepal length and width) in a lower-dimensional space. This makes it easier to see how the three Iris species cluster which is setosa versicolor and virginica and identify the most important features for differentiating them. Next, can learn how much variance each principal component explains. Typically, the first few components capture the majority of the variance, indicating which features contribute most to the overall data structure. In species relationships, PCA can help visualize if the clusters are well-defined or overlapping, revealing potential challenges for classifying individual data points. In Preprocessing for Machine Learning, PCA can be used as a pre-processing step for machine learning algorithms, particularly when dealing with high-dimensional data. Reducing the dimensionality can improve training efficiency and potentially alleviate issues like the curse of dimensionality. Lastly, applying PCA to the Iris dataset serves as a valuable learning experience for understanding fundamental concepts in data analysis and dimensionality reduction.

* 1. marks)

**Reference**

Full coding:

#calling for python libraries

import pandas as pd #provides data structures like DataFrame

from sklearn.preprocessing import StandardScaler #provides simple and efficient tools for data analysis and modeling

import numpy as np #provides multi-dimensional arrays and matrices

from sklearn.decomposition import PCA # to transform the original features of a dataset into a new set of uncorrelated features, called principal components

import matplotlib.pyplot as plt #provides a variety of functions to create different types of plots, charts, and visualizations

from sklearn.preprocessing import LabelEncoder #provides several functions for data preprocessing and feature engineering

# Load the dataset

iris\_data = pd.read\_csv('/content/iris.csv')

# Understanding the data set

numSamples, numFeatures = iris\_data.shape #used to extract the number of samples (rows) and the number of features (columns)

print("Number of samples:", numSamples)

print("Number of features:", numFeatures)

#Target column is present in the dataset, you can print the unique values in the target column which is my tar

print("Target names:", iris\_data['variety'].unique())

# Display the size of the dataset (rows, columns)

data\_size = iris\_data.shape

# Display the number of parameters/features and their details

parameter\_size = iris\_data.shape[1] # Number of columns

parameter\_details = iris\_data.columns.tolist() # List of column names

# Print the results

print("Data size:", data\_size)

print("Parameter / feature size:", parameter\_size)

print("Parameter / feature details:", parameter\_details)

# Display the entire data

print(iris\_data)

x = iris\_data.iloc[:, :-1] # Features (all columns except the last one)

y = iris\_data.iloc[:, -1] # Labels (the last column)

# Displaying x and y

print("X (Features):")

print(x)

print("\nY (Labels):")

print(y)

scaler = StandardScaler() #preprocessing technique commonly used in machine learning to standardize the features of a dataset.

scaled\_data = scaler.fit\_transform(x) #The fit method is used to compute the mean and standard deviation of each feature in the dataset

# Displaying the scaled data

scaled\_data\_df = pd.DataFrame(scaled\_data, columns=x.columns)

print("Scaled Data:")

print(scaled\_data\_df)

# Calculate the mean from the scaled data

scaled\_data\_mean = np.mean(scaled\_data, axis=0)

# Displaying the mean results based on the StandardScaler

print("Mean Result based on StandardScaler:")

print(scaled\_data\_mean)

# Subtract the scaled data mean from each row in the original data

new\_data = x - scaled\_data\_mean

# Displaying the new\_data

print("New Data after subtracting with the scaled data mean:")

print(new\_data)

# new\_data is the modified data obtained by subtracting scaled\_data\_mean

covariance\_matrix = np.cov(new\_data, rowvar=False)

# Displaying the covariance matrix

print("Covariance Matrix:")

print(covariance\_matrix)

eigenvalues, eigenvectors = np.linalg.eig(covariance\_matrix)

# Displaying the eigenvectors and eigenvalues

print("Eigenvectors:")

print(eigenvectors)

print("\nEigenvalues:")

print(eigenvalues)

# Initialize PCA with the number of components you want to retain

# For example, if you want to retain 2 components, set n\_components=2

pca = PCA(n\_components=2)

# Fit and transform the data using PCA

pca\_result = pca.fit\_transform(new\_data)

# Display the explained variance ratio

print("Explained Variance Ratio:")

print(pca.explained\_variance\_ratio\_)

# Display the transformed data

pca\_df = pd.DataFrame(data=pca\_result, columns=['Principal Component 1', 'Principal Component 2'])

print("\nTransformed Data:")

print(pca\_df)

# Concatenate the transformed data with the target variable

pca\_df\_with\_target = pd.concat([pca\_df, iris\_data['variety']], axis=1)

# Create scatter plot

plt.figure(figsize=(8, 6))

for variety in iris\_data['variety'].unique():

subset = pca\_df\_with\_target[pca\_df\_with\_target['variety'] == variety]

plt.scatter(subset['Principal Component 1'], subset['Principal Component 2'], label=variety)

# Add labels and title

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('PCA Results using Standardized Data')

plt.legend()

plt.show()