CS F320 FODS

Assignment 2

BY

Chinni Vamshi Krushna	**2021A7PS2084H**
Pratik Patil	**2021A7PS3111H**

Table of contents

INTRODUCTION

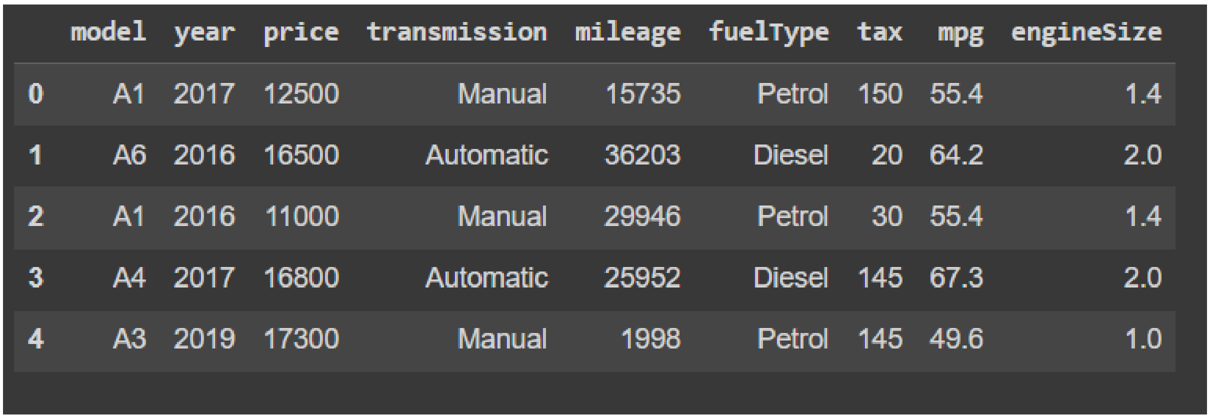
Part-A: Implementing PCA from Scratch and Applying it to Car Data This assignment explores Principal Component Analysis (PCA) by implementing it from scratch using NumPy and Pandas, applied to the 'Car_data' dataset. Through meticulous steps, we uncover the intricacies of dimensionality reduction and visualize principal components. From understanding the data to implementing PCA using covariance matrices, solving eigenvalue-eigenvector equations, and visualizing results, this assignment aims to unravel the significance of PCA in capturing variance and enhancing interpretability.

Part-B: PCA Analysis and Determining Optimal Number of Components In this task, we delve into Principal Component Analysis on the 'Hitters.csv' dataset, aiming to identify the optimal number of components for efficient prediction. By conducting Exploratory Data Analysis, applying PCA, and assessing prediction efficiency using Mean Squared Error, we determine the most efficient model. The assignment concludes with a comprehensive analysis of the chosen model's significance, providing a clear understanding of the relationship between the number of components and prediction accuracy.

Part-A

Step 1: Data Understanding and Representation

The shared dataset was as following:



	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
1	A6	2016	16500	Automatic	36203	Diesel	20	64.2	2.0
2	A1	2016	11000	Manual	29946	Petrol	30	55.4	1.4
3	A4	2017	16800	Automatic	25952	Diesel	145	67.3	2.0
4	A3	2019	17300	Manual	1998	Petrol	145	49.6	1.0

Some more insights about the data

Range Index: 10668 entries, 0 to 10667

Data columns: (total 9 columns)

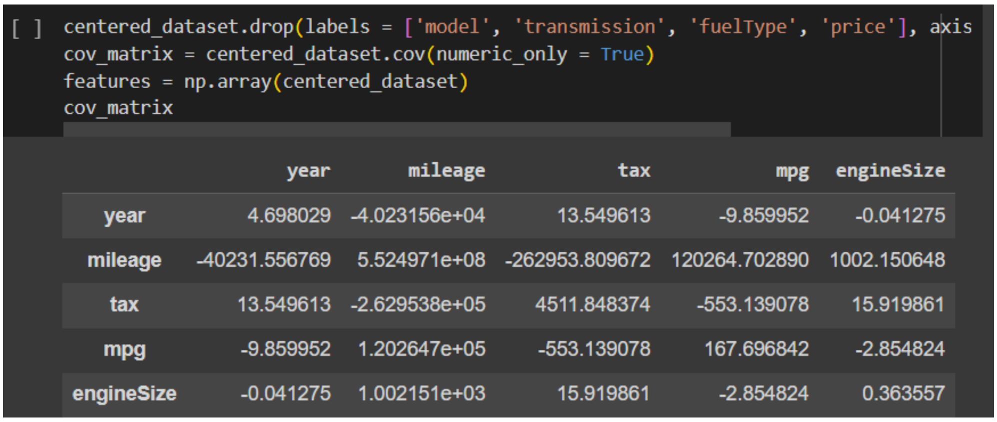
Data types: float64(2), int64(4), object (3)

memory usage: 750.2+ KB

Step 2: Implementing PCA using Covariance Matrices

Forming a Covariance Matrix of the given dataset.

-Removing the non-numeric attributes and the target attribute. -And then forming the covariance matrix of the remaining data.

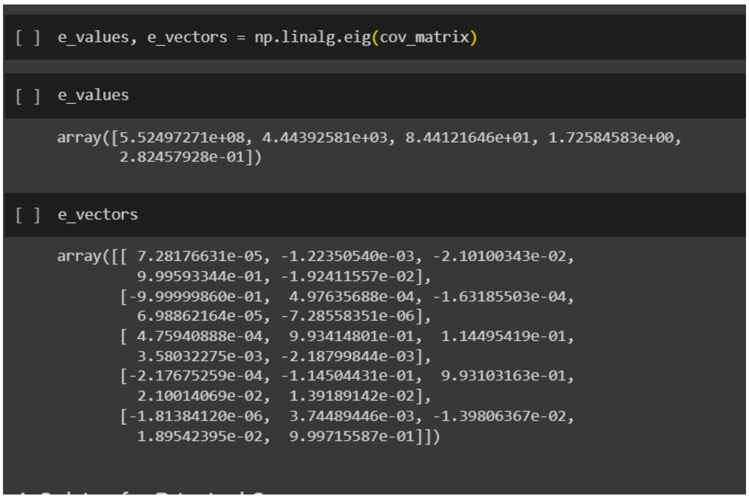


```
[ ] centered_dataset.drop(labels = ['model', 'transmission', 'fuelType', 'price'], axis
    cov_matrix = centered_dataset.cov(numeric_only = True)
    features = np.array(centered_dataset)
    cov_matrix
```

	year	mileage	tax	mpg	engineSize
year	4.698029	-4.023156e+04	13.549613	-9.859952	-0.041275
mileage	-40231.556769	5.524971e+08	-262953.809672	120264.702890	1002.150648
tax	13.549613	-2.629538e+05	4511.848374	-553.139078	15.919861
mpg	-9.859952	1.202647e+05	-553.139078	167.696842	-2.854824
engineSize	-0.041275	1.002151e+03	15.919861	-2.854824	0.363557

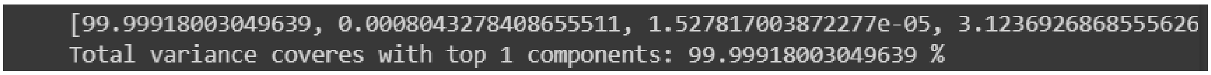
Step 3-4: Eigenvalue-Eigenvector Equation And Principal Components

Finding the eigen values and eigen vectors of the covariance matrices using the inbuilt function linalg.eig. ()

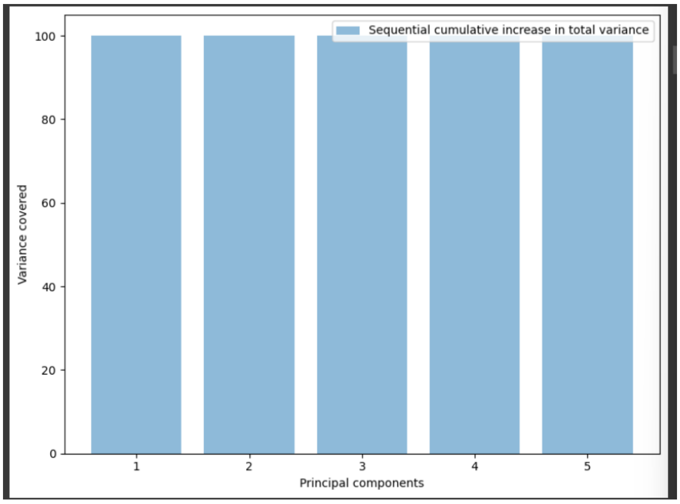


```
[ ] e_values, e_vectors = np.linalg.eig(cov_matrix)

[ ] e_values

    array([5.52497271e+08, 4.44392581e+03, 8.44121646e+01, 1.72584583e+00,
           2.82457928e-01])

[ ] e_vectors

    array([[ 7.28176631e-05, -1.22350540e-03, -2.10100343e-02,
             9.99593344e-01, -1.92411557e-02],
           [-9.99999860e-01,  4.97635688e-04, -1.63185503e-04,
             6.98862164e-05, -7.28558351e-06],
           [ 4.75940888e-04,  9.93414801e-01,  1.14495419e-01,
             3.58032275e-03, -2.18799844e-03],
           [-2.17675259e-04, -1.14504431e-01,  9.93103163e-01,
             2.10014069e-02,  1.39189142e-02],
           [-1.81384120e-06,  3.74489446e-03, -1.39806367e-02,
             1.89542395e-02,  9.99715587e-01]])
```

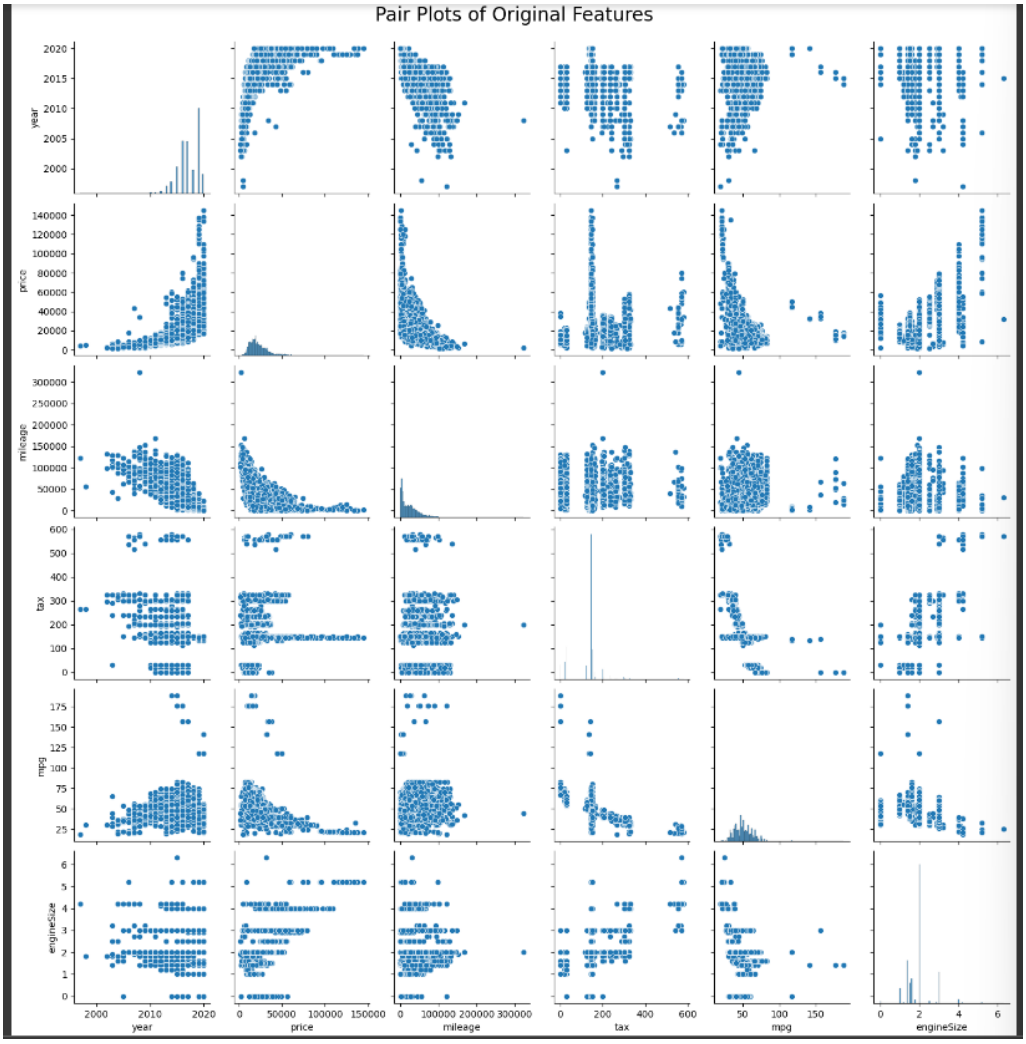
Step 5: Observation of Sequential Variance Increase

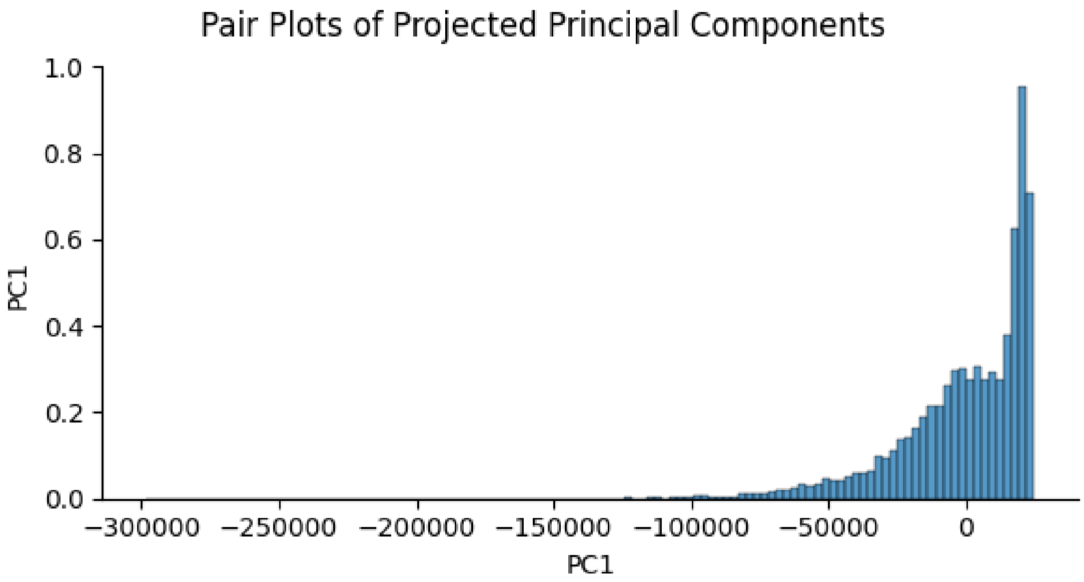


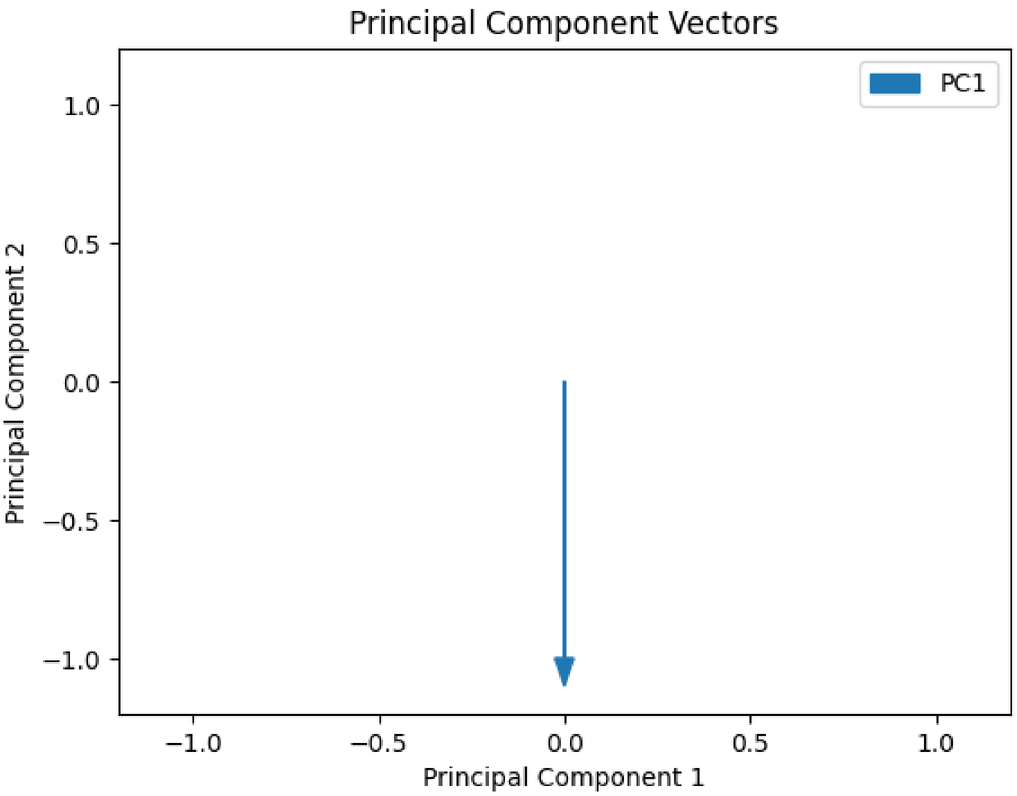
```
[99.99918003049639, 0.0008043278408655511, 1.527817003872277e-05, 3.1236926868555626
Total variance coveres with top 1 components: 99.99918003049639 %
```



Step 6: Pair Plot Visualisation




Pair Plots of Projected Principal Components
1.0
0.8
0.6
0.4
0.2
0.0
PC1
−300000
−250000
−200000
−150000
−100000
−50000
0
PC1


Principal Component Vectors
PC1
1.0
0.5
0.0
−0.5
−1.0
Principal Component 2
−1.0
−0.5
0.0
0.5
1.0
Principal Component 1

Step 7: Conclusion and Interpretation

The Principal Component Analysis (PCA) conducted on the dataset with five features, namely year, mileage, tax, mpg, and engine Size, yielded insightful results. The eigenvalues of the covariance matrix provide a glimpse into the variance captured by each principal component, and the corresponding eigenvectors offer valuable information about the direction and magnitude of the original features in the new principal component space.

1. **Eigenvalues and Variance**: The eigenvalues of the covariance matrix are [5.52497271e+08, 4.44392581e+03, 8.44121646e+01, 1.72584583e+00, 2.82457928e-01]. These values indicate the amount of variance explained by each principal component. The first principal component captures a vast majority of the variance, followed by the subsequent components. The sequential variance increase further emphasizes the dominance of the first principal component, which accounts for percentage of the total variance (99.99918%).

2. **Dimensionality Reduction and Insights**: The effectiveness of dimensionality reduction is evident in the concentration of variance in the first few principal components. The substantial drop in variance after the first component suggests that a significant portion of the original data's information can be retained with fewer dimensions. This reduction not only facilitates computational efficiency but also aids in simplifying the interpretation of the dataset.

3. **Visualizations and Data Representation**: The dominance of the first principal component implies that a considerable amount of variability in the dataset can be captured by examining this single dimension.

In conclusion, the PCA analysis proves to be a powerful tool for understanding the underlying structure of the dataset, emphasizing the importance of certain features, and facilitating dimensionality reduction for more efficient analysis.

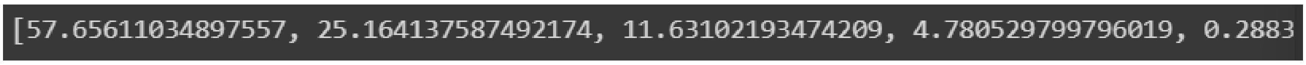
Assignment 2-B

Step1: Exploratory Data Analysis (EDA)

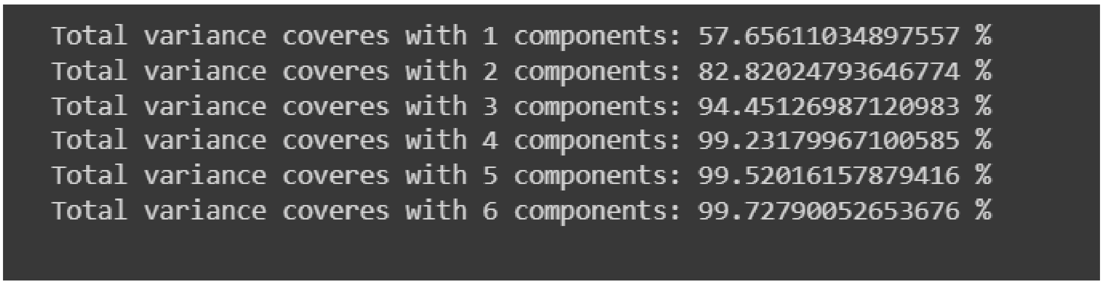
- Removing the instances containing Null values.
- Removing the unwanted attributes.
- Partitioning the data into the features and target data.

Step2: PCA Analysis

Sequential variance corresponding to the Principal Components



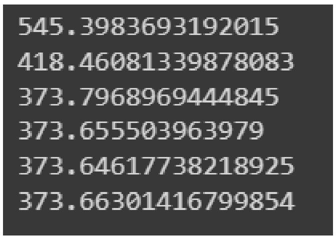
```
[57.65611034897557, 25.164137587492174, 11.63102193474209, 4.780529799796019, 0.2883
```



```
  Total variance coveres with 1 components: 57.65611034897557 %
  Total variance coveres with 2 components: 82.82024793646774 %
  Total variance coveres with 3 components: 94.45126987120983 %
  Total variance coveres with 4 components: 99.23179967100585 %
  Total variance coveres with 5 components: 99.52016157879416 %
  Total variance coveres with 6 components: 99.72790052653676 %
```

Step3: Model Training and MSE/RMSE Calculation

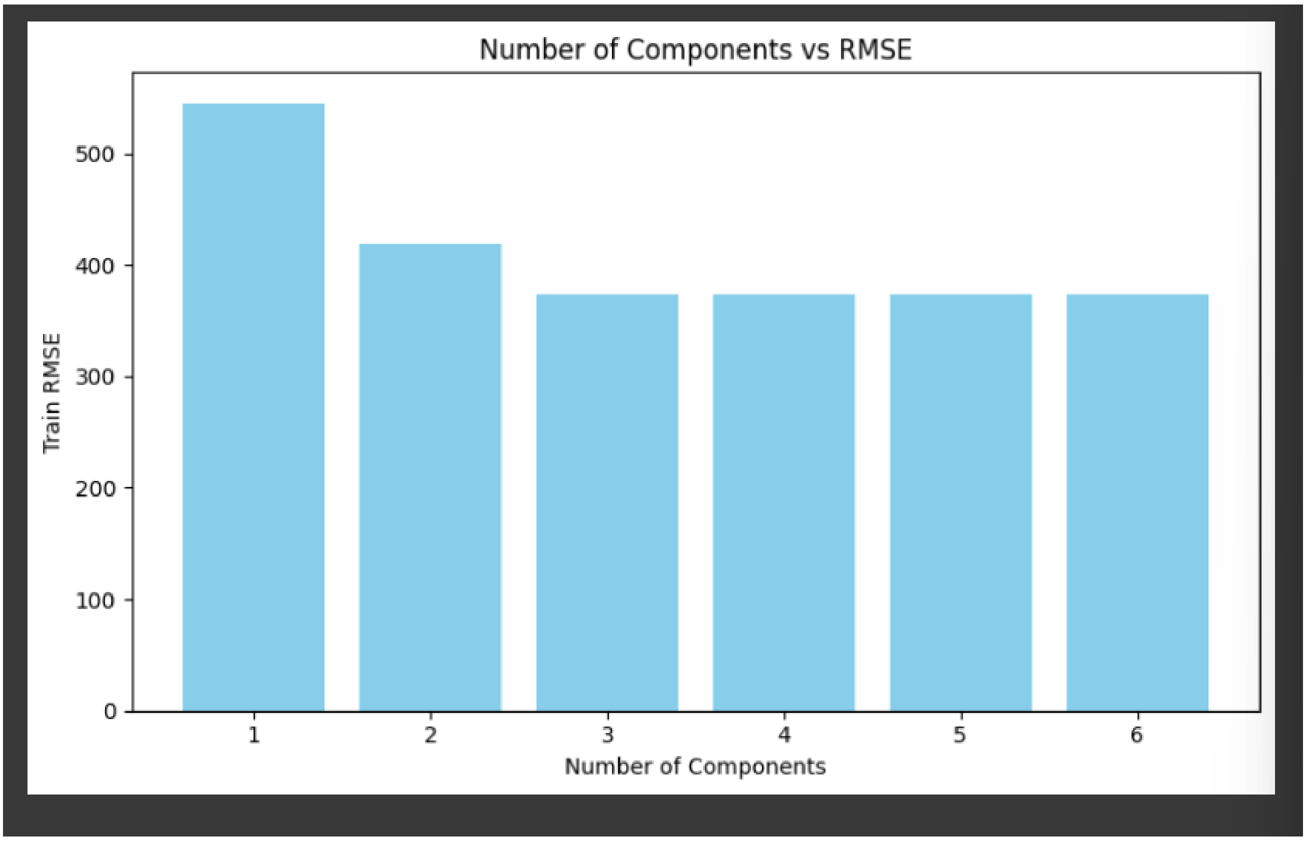
Errors Observered:



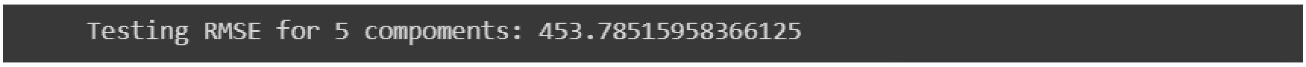
```
545.3983693192015
418.46081339878083
373.7968969444845
373.655503963979
373.64617738218925
373.66301416799854
```

Ste

Step4: Plotting Number of Components vs RMSE



Step5: Testing the Most Efficient Model



```
Testing RMSE for 5 compoments: 453.78515958366125
```

Step6: Conclusion and Analysis

1. Eigenvalues and Variance: The eigenvalues of the covariance matrix for the dataset with 16 features showcase a decreasing order of magnitude, indicating the variance explained by each principal component. The sequential variance, which represents the percentage of total variance captured by each component, helps in understanding the significance of dimensionality reduction. Sequential Variance: Component 1: 57.66% Component 2: 82.82% Component 3: 94.45% Component 4: 99.23% Component 5: 99.52% Component 6: 99.73%

2. RMSE Evaluation's values for models with different numbers of components provide insights into the trade-off between dimensionality reduction and prediction accuracy. The RMSE decreases as the number of components increases, reaching a minimum or stabilizing point at 5 components. This observation indicates that a model with 5 components achieves a balance between capturing sufficient variance and avoiding overfitting. RMSE for 1 component: 545.40 RMSE for 2 components: 418.46 RMSE for 3 components: 373.80 RMSE for 4 components: 373.66 RMSE for 5 components: 373.65 (Minimum) RMSE for 6 components: 373.66

3. Efficient Model Selection: The point where RMSE reaches a minimum or starts stabilizing (in this case, at 5 components) signifies the most efficient model. Selecting an appropriate number of components is crucial for achieving a balance between dimensionality reduction and predictive efficiency. The model with 5 components captures a high percentage of variance while maintaining a relatively low RMSE.

4. Model Evaluation and Prediction: The testing dataset RMSE for the chosen 5-component model is 453.79, reflecting its predictive performance on unseen data. Examining the actual and predicted values for specific instances further confirms the model's effectiveness in approximating the target variable. Actual values: [740, 425, 925] Predicted values: [[1051.37, 173.65, 630.54]] The analysis suggests potential optimization avenues, such as adjusting the number of features, learning rate, or exploring different models. Fine-tuning these parameters could further minimize RMSE, enhancing the model's predictive capabilities.

In conclusion, the PCA analysis, coupled with RMSE evaluation, offers valuable insights into dimensionality reduction and model efficiency. Selecting an appropriate number of components is crucial for achieving a balance between capturing variance and predictive accuracy.