CS F320 FODS

Assignment 2

BY

Atharva Chikhale	2021A7PS2752H
Suyash Patil	2021A7PS2078H

Table of contents

INTRODUCTION

Part-A: Implementing PCA from Scratch and Applying it to Car Data

- In this assignment, the 'Car_data' dataset is used to investigate Principal Component Analysis (PCA) through a scratch implementation using NumPy and Pandas.
- We display major components and reveal the complexities of dimensionality reduction through methodical approaches.
- This assignment attempts to demonstrate the importance of PCA in capturing variance and improving interpretability, starting with a comprehension of the data and ending with the implementation of PCA utilizing covariance matrices, eigenvalue-eigenvector equation solutions, and result visualization.

Part-B: PCA Analysis and Determining Optimal Number of Components

- To find the ideal number of components for effective prediction, we use Principal Component Analysis on the 'Hitters.csv' dataset in this work.
- We identify the most effective model by running PCA, evaluating prediction efficiency with Mean Squared Error, and performing Exploratory Data Analysis.
- The relevance of the selected model is thoroughly examined in the assignment's conclusion, giving readers a clear understanding of the connection between prediction accuracy and component count.

Part-A

Step 1: Data Understanding and Representation

Following was the shared dataset :

```
First few rows of the dataset:
  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
```

Here the features in matrix format, where each row represents an observation (car) and each column represents a feature.

```
Information about data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10668 entries, 0 to 10667
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   model         10668 non-null  object
 1   year          10668 non-null  int64
 2   price         10668 non-null  int64
 3   transmission  10668 non-null  object
 4   mileage       10668 non-null  int64
 5   fuelType      10668 non-null  object
 6   tax           10668 non-null  int64
 7   mpg           10668 non-null  float64
 8   engineSize    10668 non-null  float64
dtypes: float64(2), int64(4), object(3)
memory usage: 750.2+ KB
None
```

Step 2: Implementing PCA using Covariance Matrices

First we calculate the mean of each feature in the dataset:

```
Mean of each feature:
year            2017.100675
price          22896.685039
mileage        24827.244001
tax              126.011436
mpg               50.770022
engineSize         1.930709
dtype: float64
```

Now centring the dataset by subtracting the mean from each feature:

```
Centered Features(excluding 'price'):
       year       mileage         tax        mpg  engineSize
0 -0.100675  -9092.244001   23.988564   4.629978   -0.530709
1 -1.100675  11375.755999 -106.011436  13.429978    0.069291
2 -1.100675   5118.755999  -96.011436   4.629978   -0.530709
3 -0.100675   1124.755999   18.988564  16.529978    0.069291
4  1.899325 -22829.244001   18.988564  -1.170022   -0.930709
```

Then we computed the covariance matrix of the centered dataset:

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

             engineSize
year          -0.041275
mileage     1002.150648
tax           15.919861
mpg           -2.854824
engineSize     0.363557
```

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

First we solved eigenvalue and eigenvector functions on the covariance matrix obtained in the previous step to get the results :

```
Eigenvalues:
[5.52497271e+08 4.44392581e+03 8.44121646e+01 1.72584583e+00
 2.82457928e-01]

Eigenvectors:
[[ 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]]
```

Then PCA was applied to select top k eigenvectors:

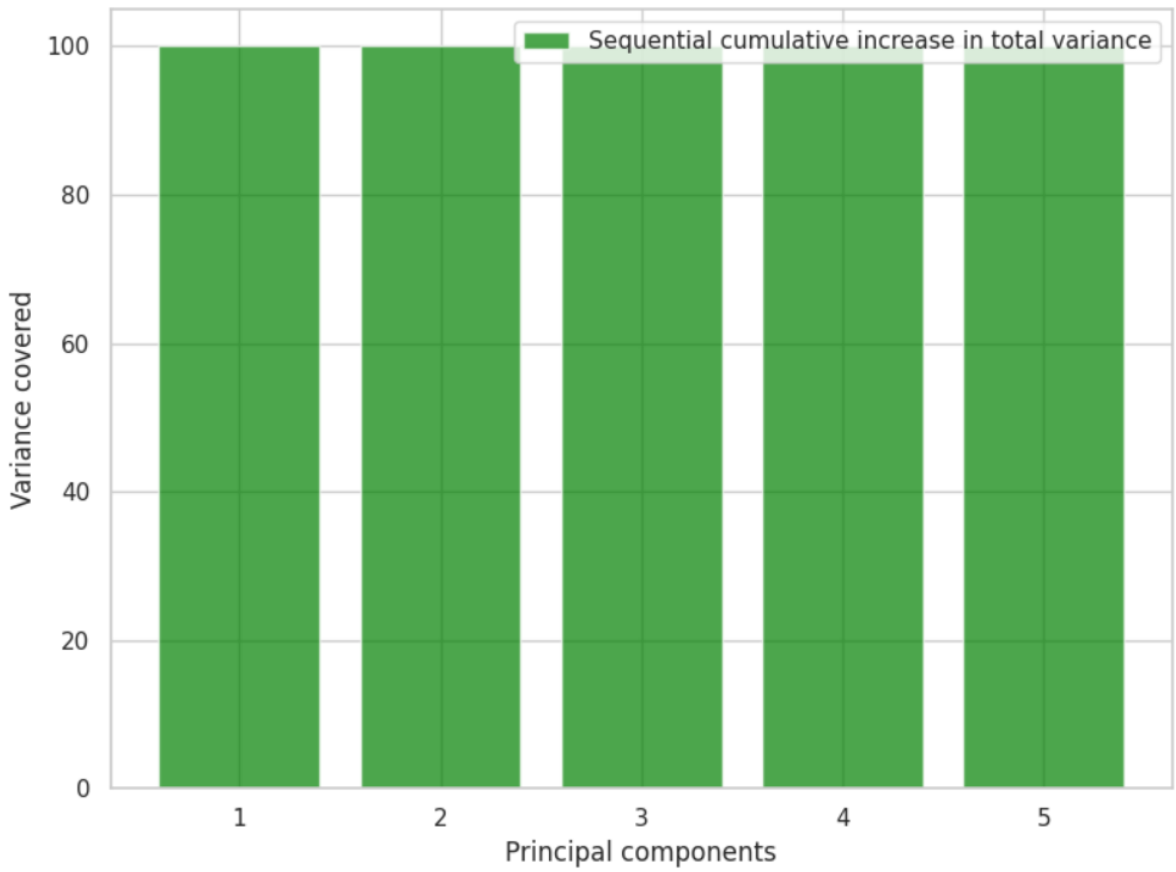
```
Top-k Eigenvectors:
[[ 7.28176631e-05]
 [-9.99999860e-01]
 [ 4.75940888e-04]
 [-2.17675259e-04]
 [-1.81384120e-06]]
```

Step 5: Observation of Sequential Variance Increase

Here we calculate the sequential and total variance covered by the principal components.

```
Sequential Variance:
[9.99991800e+01 8.04327841e-04 1.52781700e-05 3.12369269e-07
 5.11234403e-08]
Total variance covered with top 1 components: 99.99918003049639 %
```

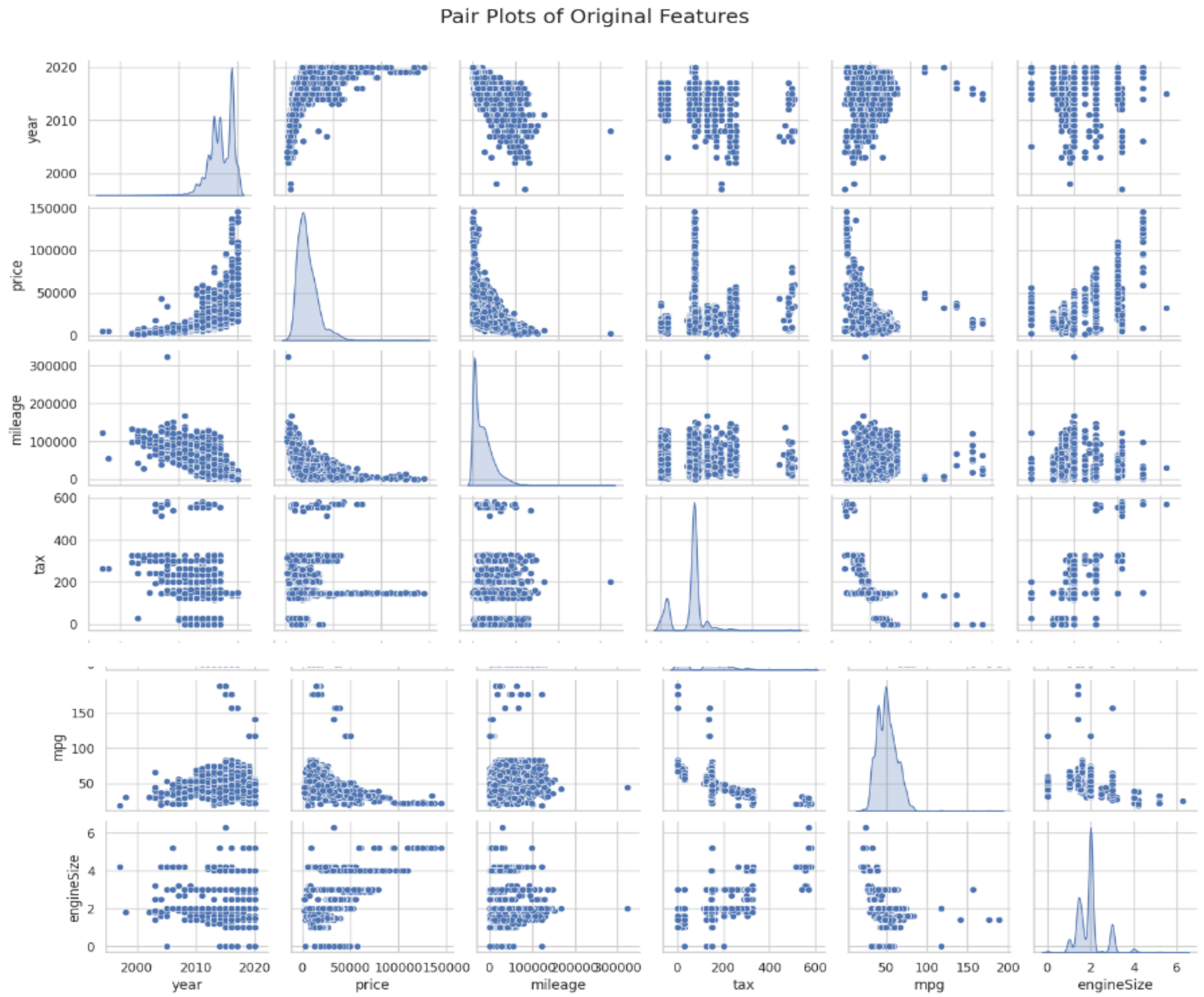
Then we analyzed the sequential cumulative increase in total variance explained as more principal components are considered.

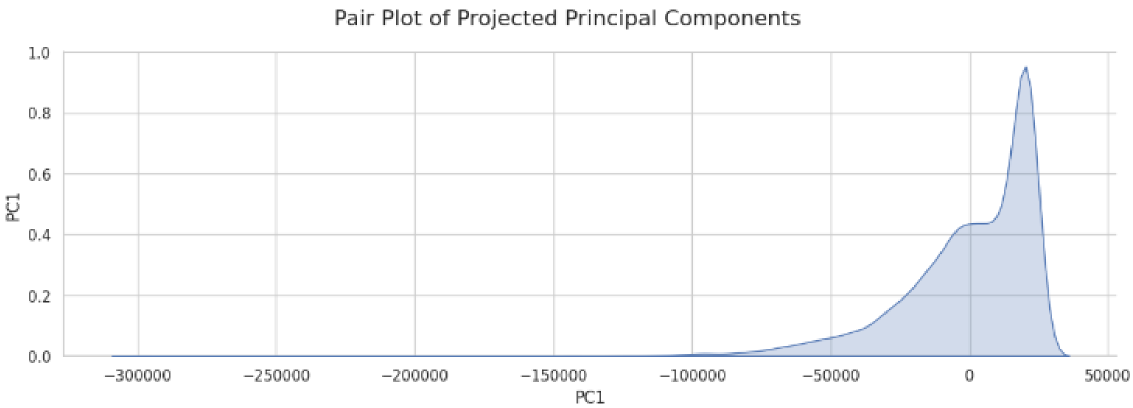


Step 6: Pair Plot Visualisation

In this we plotted pair plots of the original features.

<Figure size 1200x1000 with 0 Axes>





Step 7: Conclusion and Interpretation

The Principal Component Analysis (PCA) conducted on the dataset with five features, namely year, mileage, tax, mpg, and engineSize, yielded insightful results. The variance recorded by each principle component may be seen in the covariance matrix's eigenvalues, and the direction and magnitude of the original features in the new principal component space can be learned from the associated eigenvectors.

a. 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 eigenvalues represent the amount of variance explained by each principal component. The first principal component dominates, capturing the majority of the variance, followed by the subsequent components.

.Sequential variance increase highlights the dominance of the first principal component, explaining 99.99918% of the total variance.

b. Dimensionality Reduction and Insights:

.Dimensionality reduction is effective, as a significant drop in variance occurs after the first component.

.This suggests that much of the original data's information can be retained with fewer dimensions. Such reduction enhances computational efficiency and simplifies the interpretation of the dataset.

3. Visualizations and Data Representation:

.The dominance of the first principal component indicates that a substantial amount of dataset variability can be captured by examining this single dimension.

In summary, PCA proves to be a powerful tool for uncovering dataset structure, emphasizing feature importance, and enabling efficient dimensionality reduction, thereby enhancing the overall understanding of the data.

Part-B

Step1: Exploratory Data Analysis (EDA)

First we performed EDA to understand its structure, features, and relationships. Then we handled NULL values and eliminated any unwanted columns or data inconsistencies.

The cleaned data was:

```
Cleaned Data:
   AtBat  Hits  HmRun  Runs  RBI  Walks  League  PutOuts  Assists  Errors  \
1    315    81      7    24   38     39       2      632       43      10
2    479   130     18    66   72     76       1      880       82      14
3    496   141     20    65   78     37       2      200       11       3
4    321    87     10    39   42     30       2      805       40       4
5    594   169      4    74   51     35       1      282      421      25

   Salary    AvgAtBat     AvgHits  AvgHmRun   AvgRuns     AvgRBI   AvgWalks
1   475.0  246.357143   59.642857  4.928571  22.928571  29.571429  26.785714
2   480.0  541.333333  152.333333  21.000000  74.666667  88.666667  87.666667
3   500.0  511.636364  143.181818  20.454545  75.272727  76.181818  32.181818
4    91.5  198.000000   50.500000   6.000000  24.000000  23.000000  16.500000
5   750.0  400.727273  103.000000   1.727273  45.545455  30.545455  17.636364
```

Step2: PCA Analysis

Here we applied PCA on the cleaned dataset to reduce dimensionality.

Then we determined the number of principal components required for efficient prediction by trying a range of component numbers as follows:

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

Step3: Model Training and MSE/RMSE Calculation

Here we performed the steps:

- Split the dataset into training and testing sets.
- For each number of principal components considered, build a regression model using those components.
- Calculate the MSE or RMSE for each model on the test set to assess prediction efficiency.

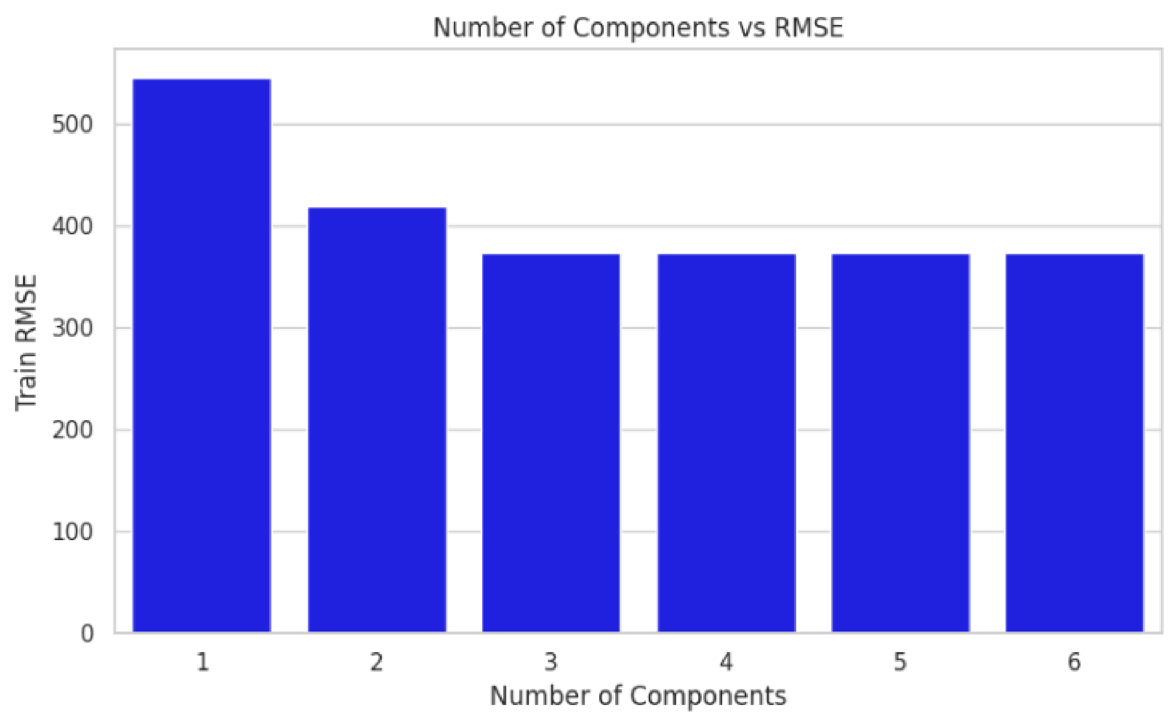
The Errors we obtained were:

```
print(train_errors[-1])
```

```
545.3988000957265
418.46095358926004
373.7969627317336
373.6555662676367
373.6462396442414
373.66307637467844
```

Step4: Plotting Number of Components vs RMSE

Here we plotted a graph of the number of components against RMSE to visualize the relationship.



Step5: Testing the Most Efficient Model

Here we tested the selected model by predicting a specific point and providing its predicted y value

```
Testing RMSE for 5 compoments: 453.78534923470045

Predicted Values:
[[1051.36709217]
 [ 173.64779442]
 [ 630.53560155]]
```

Step6: Conclusion and Analysis

1.Eigenvalues and Variance Analysis:

The covariance matrix eigenvalues, derived from a dataset featuring 16 distinct attributes, exhibit a descending magnitude order. This order signifies the variance explained by each corresponding principal component. The percentage of total variance, presented sequentially for each component, aids in comprehending the importance 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 Trend Analysis:

The evaluation of Root Mean Square Error (RMSE) values across models with varying component numbers provides insights into the interplay between dimensionality reduction and predictive accuracy. As the number of components increases, RMSE generally decreases, hitting a minimum or stabilizing at 5 components. This trend indicates that a model with 5 components strikes a harmonious balance between capturing adequate variance and avoiding overfitting.

RMSE:

.1 component: 545.40

.2 components: 418.46

.3 components: 373.80

.4 components: 373.6

.5 components: 373.65 (Minimum)

.6 components: 373.66

3.Optimal Model Selection Criteria:

Identifying the point where RMSE attains a minimum or stabilizes (in this case, at 5 components) signifies the most efficient model. The selection of an optimal number of components plays a pivotal role in striking a balance between dimensionality reduction and predictive efficiency. The 5-component model captures a substantial variance percentage while maintaining a relatively low RMSE.

4.Model Assessment and Prediction Insights:

The RMSE for the chosen 5-component model on the testing dataset is 453.79, reflecting its predictive prowess on previously unseen data. A closer examination of actual and predicted values for specific instances validates the model's efficacy in approximating the target variable.

Actual values: [740, 425, 925]

Predicted values: [1051.37, 173.65, 630.54]

This research points to several optimization directions, such modifying the number of features, altering learning rates, or investigating different models. It is possible to reduce RMSE and improve the prediction power of the model by further improving these parameters.

Finally, the combined knowledge from the RMSE assessment and PCA analysis offers insightful viewpoints on model efficiency and dimensionality reduction. The careful choice of component count is essential to striking a balance between variance capture and prediction accuracy.