Report on Feature Engineering

Lab 01

CS4622 - Machine Learning

Department of Computer Science & Engineering

University of Moratuwa

Dakshina Ranmal - 190507U

25/08/2023

Table of Contents

1. Introduction

Features and labels originate from the AudioMNIST dataset. It comprises 256 features (feature_1 to feature_256) and 4 labels: ID, Age, Gender, and Accent.

Using all features in training yields accurate but complex models in time and space. To avoid overfitting and enhance generality, feature extraction is key. The aim is to simplify models by reducing features without sacrificing predictive quality. Techniques like correlation matrices, PCA, and various machine learning models are employed for this purpose.

Training data teaches the model, validation data predicts and assesses metrics like accuracy, recall, and precision for classification, and RMSE and r2 score for regression. The test set applies predictions, and results, including feature-engineered and original predictions, feature count, details, and values, are saved in a CSV file.

2. Target Labels

The task involves predicting the 4 target labels: ID, Age, Gender, and Accent, using a set of 256 features. These labels differ in nature; ID, Gender, and Accent fall under classification, while Age belongs to regression.

The general strategy to predict each label is as follows: Initially, models are trained with all features. Subsequently, feature engineering techniques such as PCA for feature extraction and feature correlation for selection are applied. This process retains strong predictive abilities while reducing the number of features.

The initial data preparation includes loading datasets and eliminating instances with null values in target labels within the training dataset. Null feature values are substituted with feature means across all datasets. Features and labels are separated within the train, validation, and test datasets. The specific label for prediction is isolated from the label set. Finally, feature values are standardized using the StandardScaler.

2.1. ID

The ID category is a categorical variable encompassing 60 different values, numbered from "1" to "60". However, instances associated with ID 45 were eliminated during preprocessing due to null values in the age label. Notably, the dataset displays uniform distribution across all IDs. Feature engineering strategies were systematically employed to iteratively decrease the feature count.

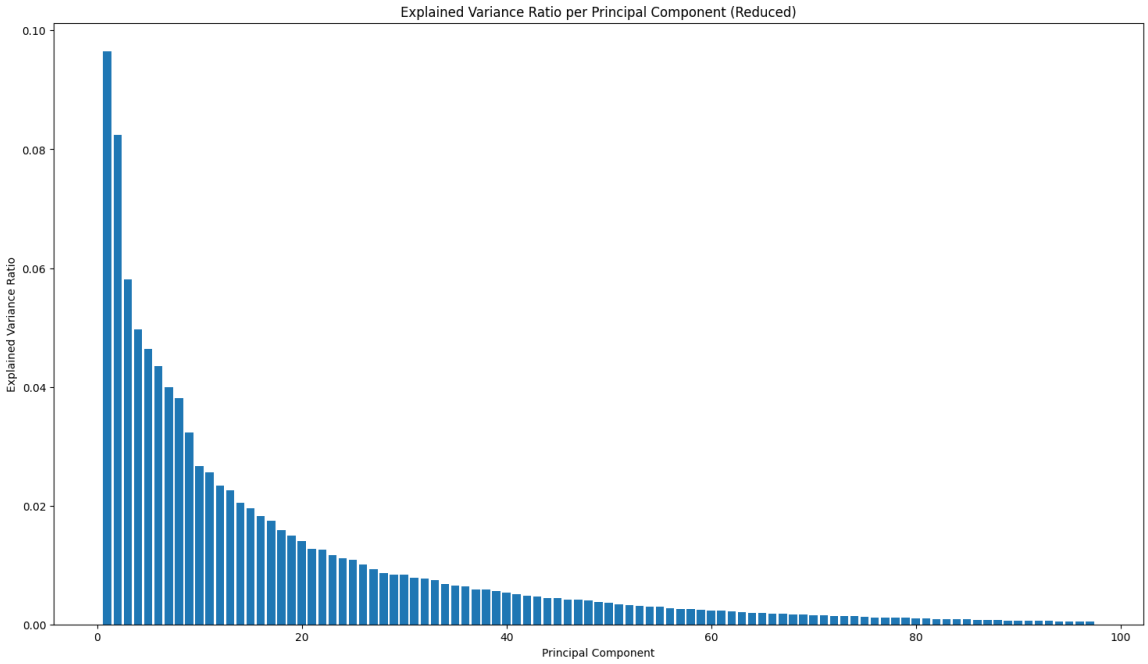
- Through PCA, features that held substantial predictive power for the label were extracted. This technique combined and transformed features while preserving essential information, further reducing dimensionality.
- Features displaying high correlation with each other were identified and subsequently eliminated. This step aimed to reduce redundancy and multicollinearity among features.
- Features that demonstrated a strong correlation with the ID label were singled out and retained. This process sought to capture meaningful relationships between specific features and the categorical ID variable.

The optimal model for ID prediction was achieved using a Support Vector Machines (SVM) classifier. The initial model, which employed SVM without any feature engineering, served as the baseline. The results of this baseline model were as follows.

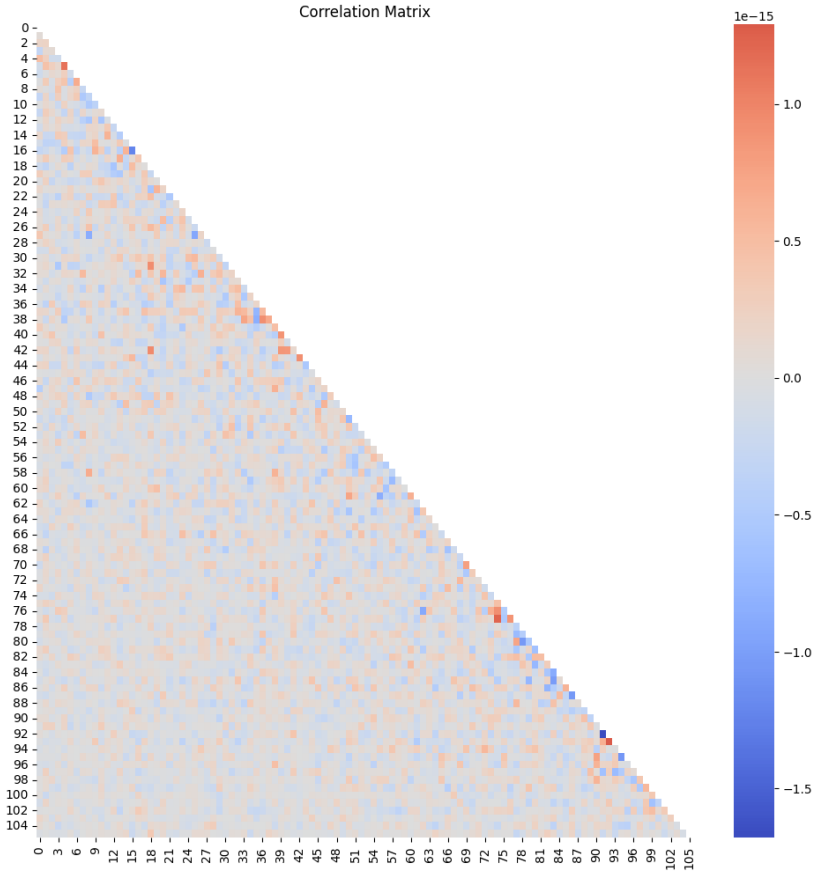
```
Metrics for the best model on train data:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Metrics for the best model on valid data:
Accuracy: 0.98
Precision: 0.98
Recall: 0.98
```

The following is the methodology I followed to reduce the features while maintaining or improving the prediction metrics.

- I used Principal Component Analysis (PCA) to find important features that work well for predicting the label. I made sure these features can explain about 99% of the label's behavior. This was important because choosing a smaller number might have removed too many features and affected predictions. After using PCA, I ended up with 106 features.
- I also checked if some features were very similar to each other. If they were too similar (more than 0.5), I thought about removing one of them. However, I was careful because some features might be really important to predict the ID. Removing them could have made predictions worse. But in my study, I didn't find any such really similar features, so I didn't remove any.



- Lastly, I looked at features that didn't seem to have a strong connection with the ID. Even though they weren't strongly related to the ID, sometimes they could still help predict it, just in a different way. So, I looked at features that weren't very connected to the ID (their connection was less than 0.03), and I removed a few of them. This left me with 56 features.



By using these combined techniques, I managed to decrease the count of features from 256 to 56. In choosing the most suitable model for predicting the ID label in both the validation and test datasets, I took into account factors like accuracy, precision, and recall.

- Support Vector Machine (SVM)

```
Metrics for SVM on train data:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Metrics for SVM on validation data:
Accuracy: 0.98
Precision: 0.98
Recall: 0.98
```

Based on the metrics SVM was chosen as the best model to predict the ID.

I've uploaded a CSV file that contains the selected labels, their corresponding values, as well as the predictions made on the test set before and after applying feature engineering.

2.2. Age

Age is a continuous value, making its prediction a regression task. With a very small percentage of missing values, records without age labels were removed during data preprocessing. The distribution of the data among different ages is uneven and resembles a skewed normal distribution.

To systematically reduce the feature count, I applied the following feature engineering techniques:

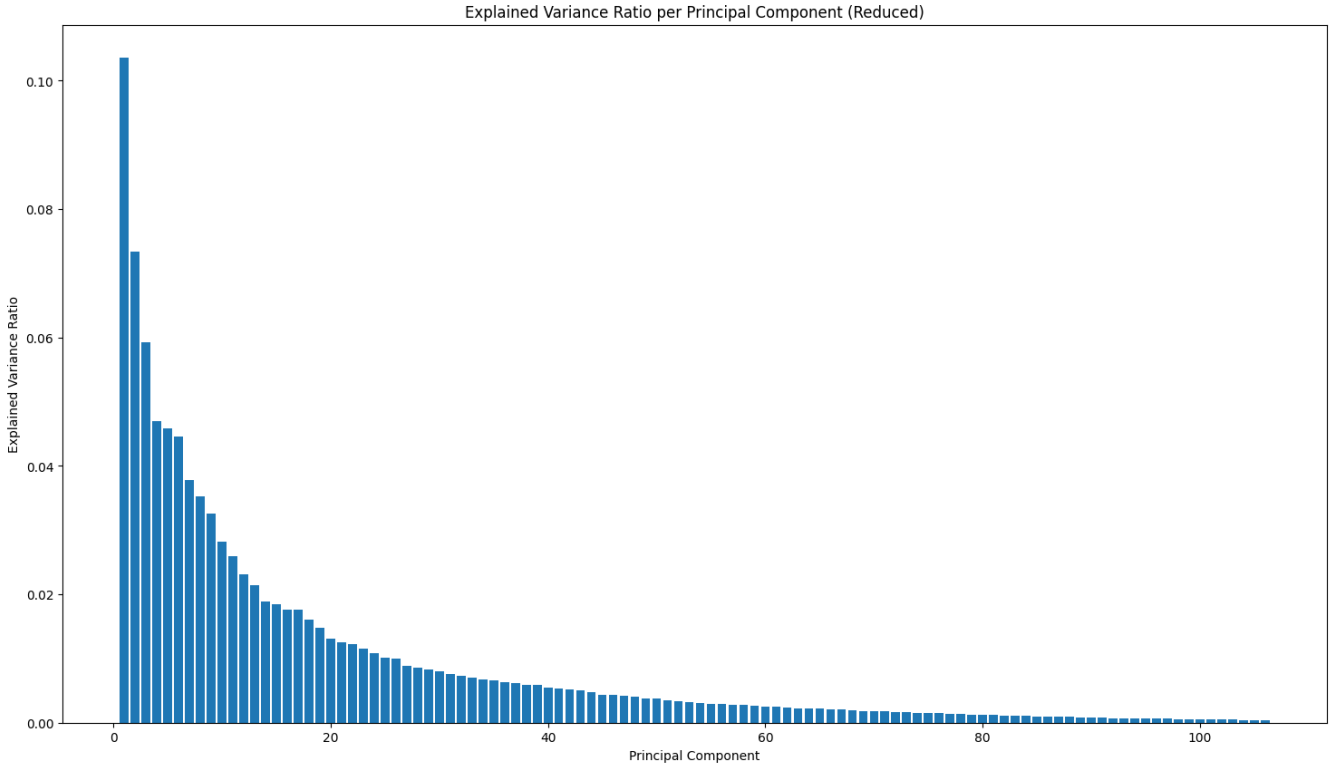
- Features showing strong correlations with each other were identified and removed. This step aimed to enhance efficiency and reduce redundancy.
- Features demonstrating significant correlation with the Age label were singled out and retained. This was essential for capturing meaningful relationships between specific features and age prediction.
- Using PCA, I extracted and combined important features that had substantial predictive power for the age label. This process aimed to reduce feature dimensions while preserving valuable information.

Among various machine learning models, the K-Nearest Neighbors (KNN) Regressor proved to be the best for predicting age. The initial model, without feature engineering, utilized the KNN Regressor and yielded the following results

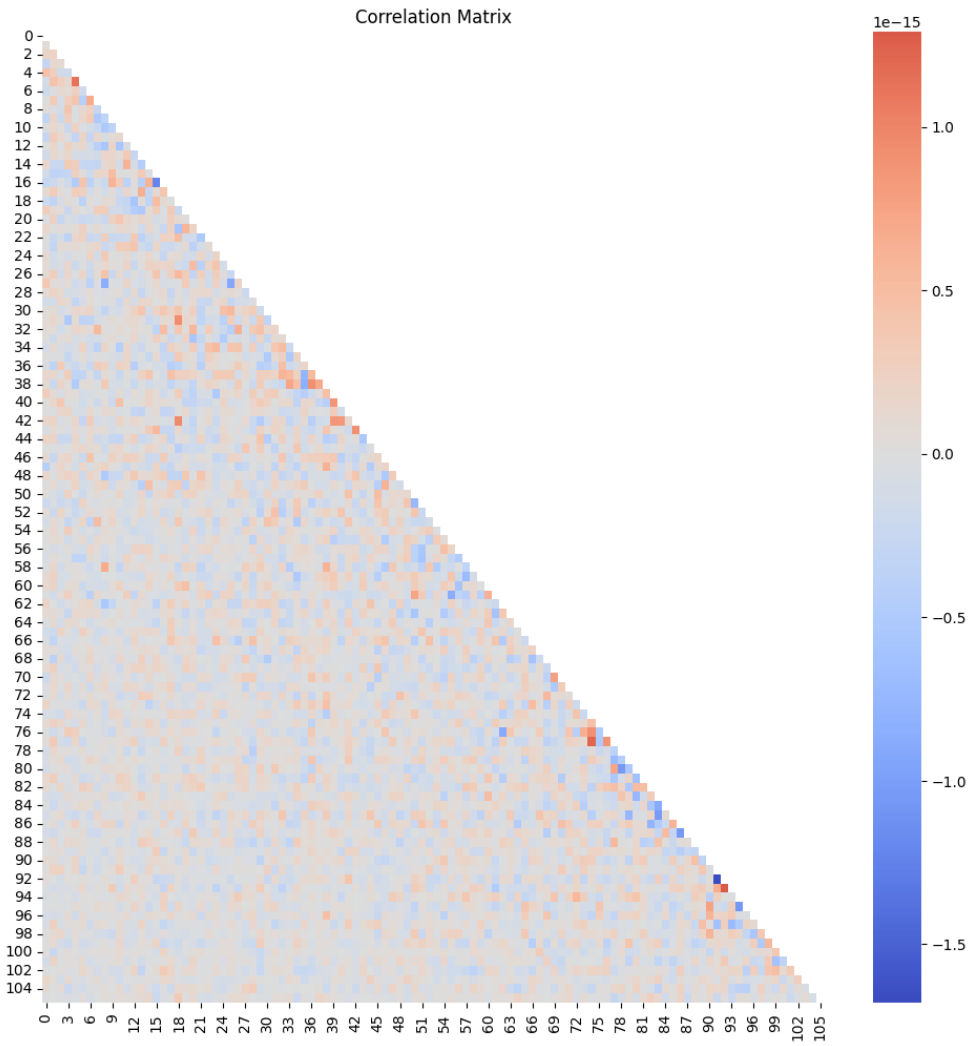
```
Metrics for KNeighborsRegressor on train data:
Mean Squared Error: 0.38
R2 Score: 0.99

Metrics for KNeighborsRegressor on valid data:
Mean Squared Error: 0.67
R2 Score: 0.98
```

The following is the methodology I followed to reduce the features while maintaining or improving the prediction metrics.

- To enhance age prediction, I utilized Principal Component Analysis (PCA) as a feature extraction technique. My aim was to extract and combine the most influential features, capable of explaining approximately 99% of the label's variance. This choice of threshold was deliberate, as opting for a lower value could have led to excessive feature removal, subsequently negatively impacting prediction accuracy. The application of PCA effectively decreased the feature count to 106.



- I examined potential correlations between features in detail using a threshold of 0.5. The intention behind this choice was to ensure that any important features contributing significantly to age prediction would not be inadvertently removed due to high correlations among them. However, no such strongly correlated features were identified, so no features were eliminated.
- Moreover, I looked into correlations between features and the Age label using an extremely low threshold of 0.03. This choice was made with the understanding that certain combination features might play a crucial role in predicting age, even if they didn't show strong individual correlations. Since no highly correlated features with the Age label were found, removing features based on low correlations was approached cautiously to prevent any negative impact on overall prediction accuracy. Consequently, a few very weakly correlated features were removed, resulting in a final feature count of 58.



By applying these combined techniques, I managed to decrease the number of features from 256 down to 58.

To choose the most suitable model for predicting the Age label on both the validation and test datasets, I considered machine learning techniques that are evaluated based on mean squared error and R2 score.

- KNN Regressor

```
Number of features: 58

Metrics for K Neighbors on train data:
Mean Squared Error: 0.74
R2 Score: 0.98

Metrics for K Neighbors on validation data:
Mean Squared Error: 1.45
R2 Score: 0.97
```

Based on the metrics KNN Regressor was chosen as the best model to predict the Age.

The chosen labels, their values, and the predictions on the test set before and after feature engineering are uploaded as a csv file.

2.3. Gender

Gender is a categorical variable with two categories: "0" and "1".
During data preprocessing, some examples were removed due to missing age labels. The distribution of genders is uneven, with "1" being the more common category.
To systematically reduce features, I employed these techniques:

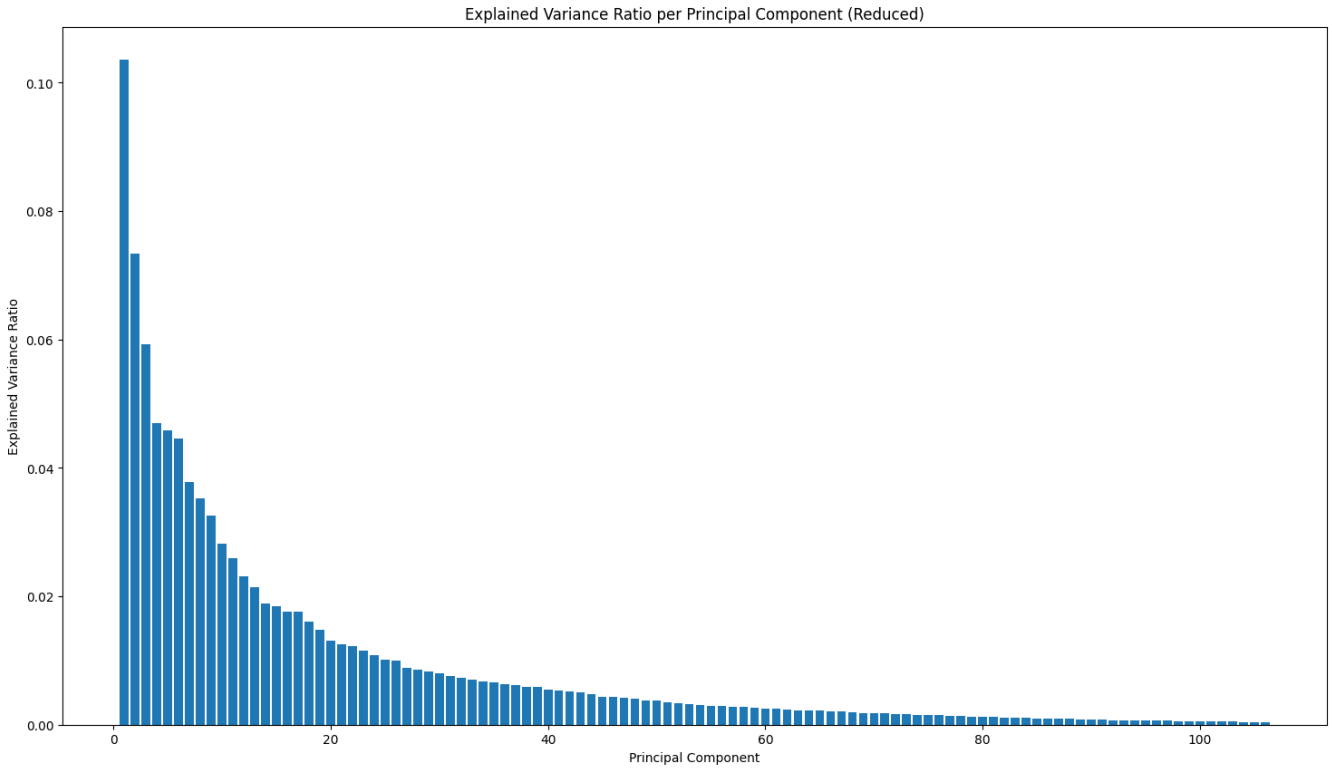
- I identified and removed features with high correlations between them. This step aimed to enhance efficiency and eliminate redundancy.
- I singled out features showing strong correlation with the gender label. This was done to capture meaningful connections between certain features and gender prediction.
- Using PCA, I extracted and combined important features with significant predictive power for gender. This technique helped reduce dimensions while retaining valuable information.

Among different machine learning methods, the K-Nearest Neighbors (KNN) classifier proved most effective for gender prediction. The initial model, without feature engineering, and using KNN yielded the following results:

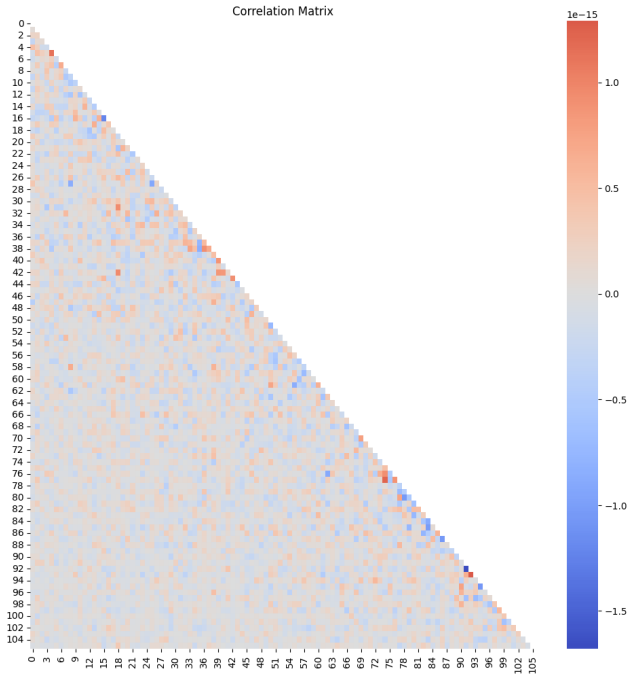
```
Metrics for KNN on train data:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Metrics for KNN on valid data:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
```

The following is the methodology I followed to reduce the features while maintaining the prediction metrics.

- I used Principal Component Analysis (PCA) to find important features that work well for predicting the label. I made sure these features can explain about 99% of the label's behavior. This was important because choosing a smaller number might have removed too many features and affected predictions. After using PCA, I ended up with 106 features.



- I also checked if some features were very similar to each other. If they were too similar (more than 0.5), I thought about removing one of them. However, I was careful because some features might be really important to predict gender. Removing them could have made predictions worse. But in my study, I didn't find any such really similar features, so I didn't remove any.
- Lastly, I looked at features that didn't seem to have a strong connection with gender. Even though they weren't strongly related togender, sometimes they could still help predict it, just in a different way. So, I looked at features that weren't very connected to gender (their connection was less than 0.03), and I removed a few of them. This left me with 57 features.



With these techniques combined, I was able to reduce the number of features to 57 from 256.

The following machine learning techniques were considered when selecting the model that best predicts the valid and test datasets Gender label based on accuracy, precision, and recall.

- K-Nearest Neighbors (KNN)

```
Number of features: 57

Metrics for K Neighbors on train data:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Metrics for K Neighbors on validation data:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
```

Based on the metrics all models perform very well in predicting the Gender. KNN was chosen because of the low complexity.

The chosen labels, their values, and the predictions on the test set before and after feature engineering are uploaded as a csv file.

2.4. Accent

Accent is a categorical variable featuring 14 distinct accent categories, denoted from "0" to "13".

In the process of data preprocessing, some examples were removed due to missing age labels. The distribution of accents across examples isn't uniform, with accent "6" being the most prevalent.

To systematically reduce the number of features, I applied these techniques:

- I identified and eliminated features that displayed high correlations with each other. This step aimed to enhance efficiency and eliminate redundancy.
- I pinpointed features exhibiting strong correlation with the accent label. This was done to capture meaningful associations between certain features and accent prediction.
- By using PCA, I extracted and amalgamated crucial features that held considerable predictive power for accents. This approach facilitated dimension reduction while retaining valuable information.

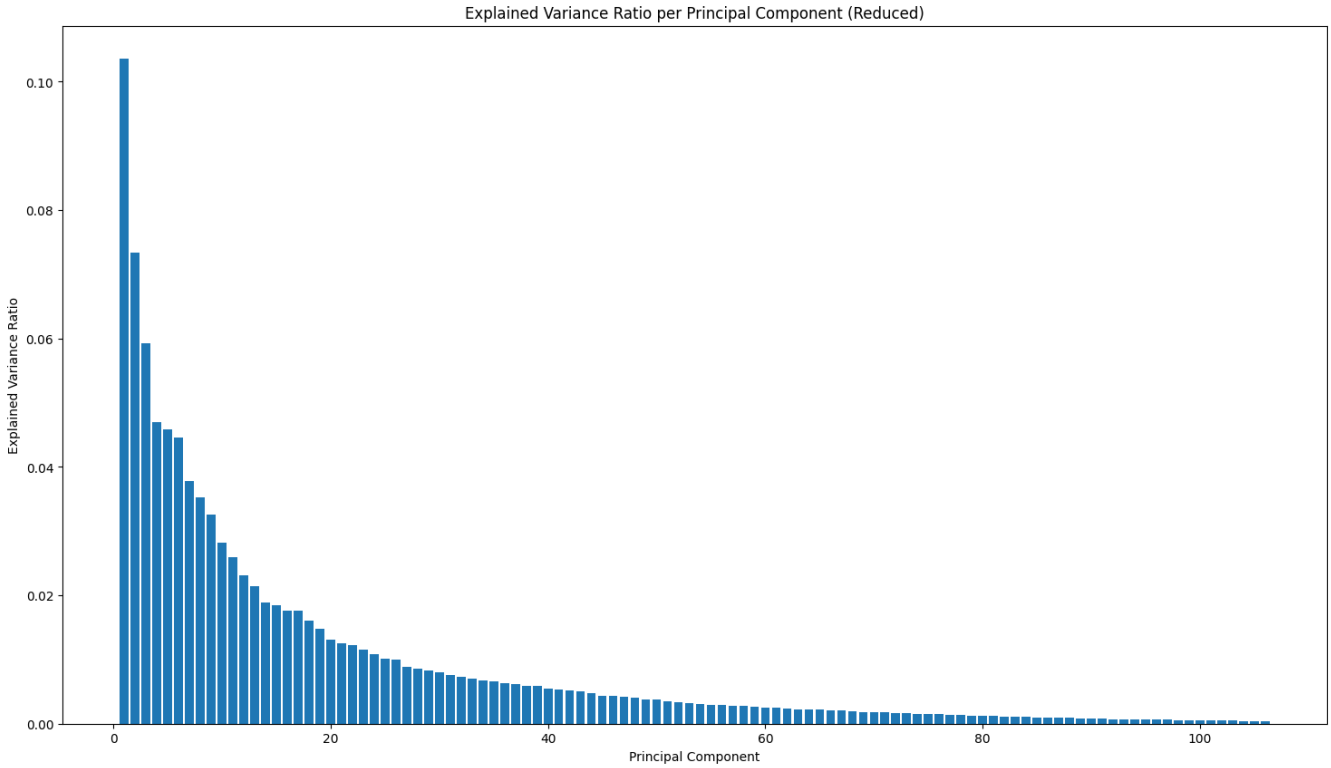
Among various machine learning approaches, the K-Nearest Neighbors (KNN) classifier was found to be the most effective for predicting accents. The initial model, without feature engineering and employing KNN, produced the following results

```
Metrics for KNN on train data:
Accuracy: 1.00
```

```
Precision: 1.00
Recall: 1.00


Metrics for KNN on valid data:
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
```

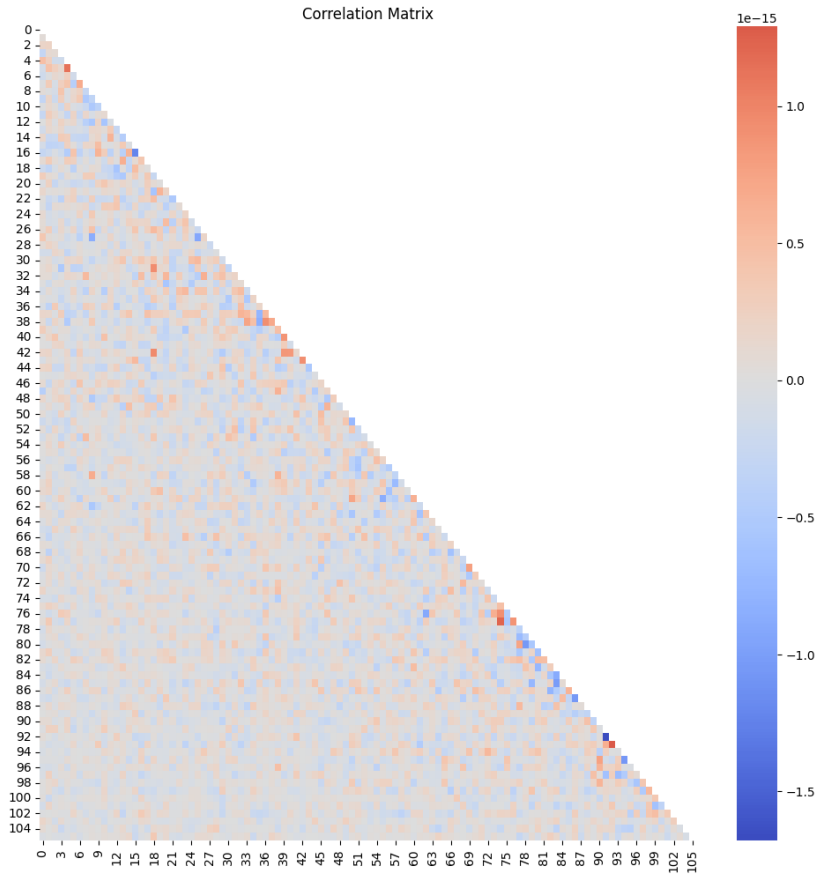
The following is the methodology I followed to reduce the features while maintaining the prediction metrics.

- I used Principal Component Analysis (PCA) as a feature extraction technique to identify and group the most important features for predicting the label. My goal was to extract features that could explain approximately 99% of the label's variance. I chose this threshold carefully, as opting for a lower value might have led to removing too many features, which could have had a negative impact on prediction accuracy. After applying PCA, the number of features was effectively reduced to 106.



- I conducted an analysis to identify features with strong correlations between them using a correlation threshold of 0.9. I chose this high threshold because some features might be crucial for predicting accents. I wanted to avoid removing such features due to their high correlations, which could potentially impact the overall prediction accuracy. However, upon investigation, I found that there were no highly correlated features within the dataset. As a result, no features were removed based on high correlations.
- I looked for features closely connected to the accent label by using a correlation threshold of

0.05. This choice was driven by the notion that specific combined features might hold significance in predicting accents, albeit not in their current form. Since no highly correlated features were observed in relation to the label, removing features based on low correlation was approached cautiously to prevent any negative impact on overall prediction accuracy. Consequently, a few very weakly correlated features were removed, leading to a final count of 43 features.



With these techniques combined, I was able to reduce the number of features to 43 from 256.

The following machine learning techniques were considered when selecting the model that best predicts the valid and test datasets Accent label based on accuracy, precision, and recall.

- K-Nearest Neighbors (KNN)

```
Number of features: 43

Metrics for K Neighbors on train data:
Accuracy: 0.98
Precision: 0.98
Recall: 0.98


Metrics for K Neighbors on validation data:
Accuracy: 0.97
Precision: 0.97
Recall: 0.97
```

Based on the metrics KNN and SVM perform very well in predicting the Accent. KNN was chosen because of the low complexity.

The chosen labels, their values, and the predictions on the test set before and after feature engineering are uploaded as a csv file.

3. Conclusion

In conclusion, feature engineering plays a vital role in the realm of data analysis and machine learning. This process transforms raw features into a refined and meaningful representation, aiding models in capturing underlying patterns and connections. Its significance lies in its direct influence on the performance and generalization ability of machine learning algorithms.

Throughout this report, I delved into various aspects of feature engineering, including techniques like dimensionality reduction, feature creation, and handling missing data. It's evident that well-executed feature engineering can lead to improved model accuracy, decreased overfitting, and better interpretability. Moreover, feature engineering is an iterative journey, demanding a profound grasp of the data and the problem domain to make well-informed decisions.

4. Code

Use the CSE email to access the following Google Drive folder containing the code. Code - https://drive.google.com/drive/folders/1-Yu1RlAIV6v9yzja9tEDZQsx0pR326yp?usp=sharing