CS 4622 Machine Learning
Lab 01 - Report

Index No: 190116U
Name: Dasanayaka D.R.S.D.

Python notebook

Google colab link: 190116U_Lab 1.ipynb
DMS link: https://dms.uom.lk/s/Qq4YWwwypcwZXbJ

Training Data set

The training data set for the lab included 256 features with 4 labels. The 4 labels are Speaker ID, Speaker age, Speaker gender, and Speaker accent respectively. Among these 4 labels, I identified labels 1,3, and 4 as classification problems and label 2 as a prediction problem. Furthermore, the training data set included some missing values for label_2.

Models used for the lab

following models are used to make the prediction.

- Label_1: Random forest
- Label_2: XGBoost
- Label_3: Random forest
- Label_4: Random forest

Since lebel_2 is a regression problem, I used XGBoost instead of random forest.

Data Preprocessing

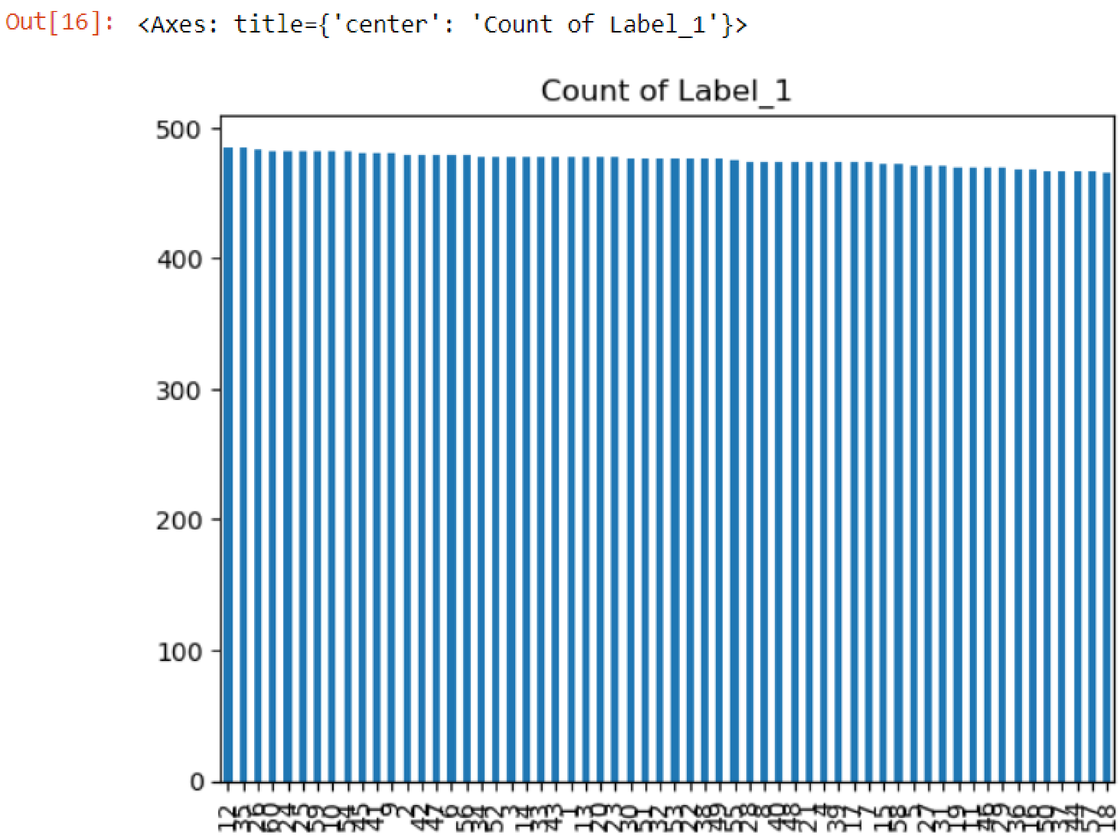
Since the random forest is a tree-based model, I didn't standardize the data. However, for the label_2, I used StandardScaler() to scale the feature values.

Furthermore, I drop the rows that have empty values since the number of rows that have missing values is relatively negligible.

Feature Engineering

Label_1

1. As the first step, I used PCA to transform the feature set. By choosing 0.98 as the variance I could reduce the number of features to 88 without losing the accuracy.
2. Then I checked the data set to identify if there was any bias in the data set. But there wasn't any significant bias for label_1.



```
Out[16]: <Axes: title={'center': 'Count of Label_1'}>
```

3. Then I trained the model using a reduced data set. After that, I remove less important features for the prediction utilizing RandomForestClassifier’s feature_importances_ attribute. Using this I could reduce the number of features to 67 features.

```
importance = rf.feature_importances_
columns_to_delete = []
for i,v in enumerate(importance):
    if v < 0.008:
        columns_to_delete.append(i)
train_reduced = np.delete(X1_train_pca, columns_to_delete,
axis=1)
test_reduced = np.delete(X1_test_pca, columns_to_delete,
axis=1)
test_reduced_ = np.delete(X1_test_pca_, columns_to_delete,
axis=1)
```

4. Finally, I test the new feature set with the validated data set. I got 0.96 accuracy with 67 features.

Label_2

1. First I scale all the features using StandardScaler()

```
scaler = StandardScaler()
scaler.fit(X2_train)
```

```
X2_train_sca = scaler.transform(X2_train)
```

2. Then, I used PCA to transform the feature set. By choosing 0.95 as the variance I could reduce the number of features to 88 without losing the accuracy.

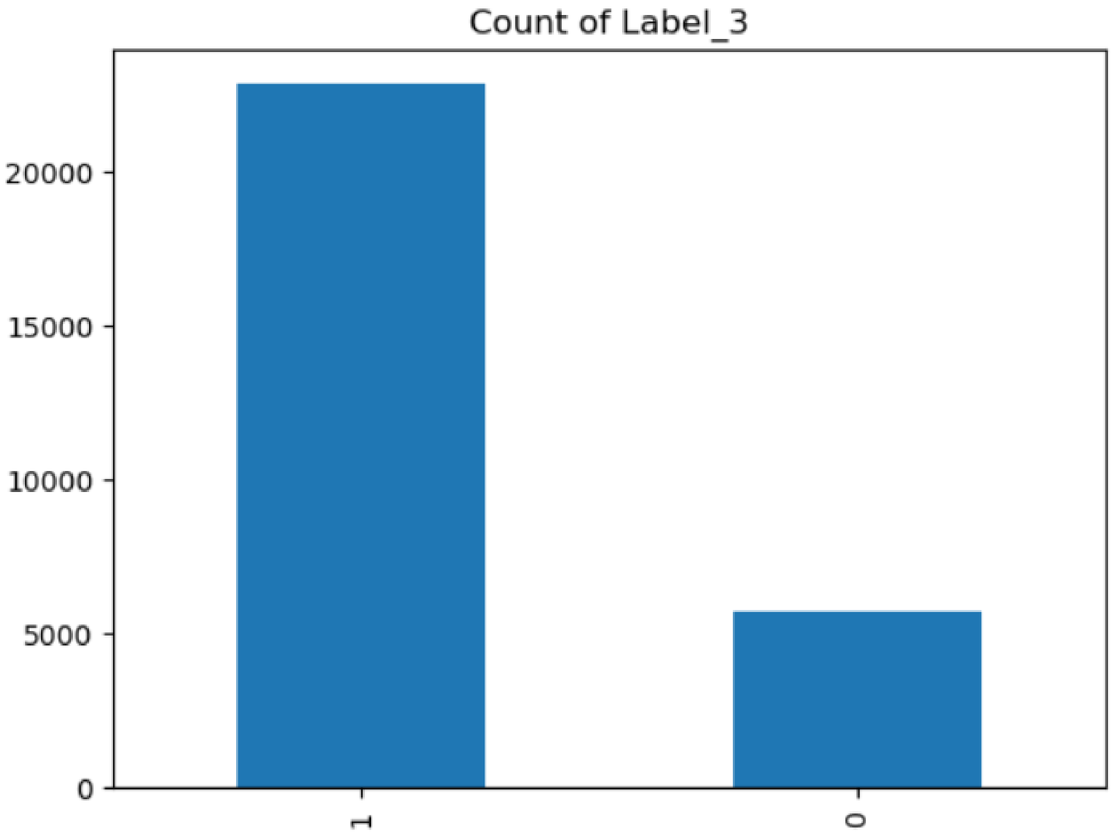
```
scaler = StandardScaler()
scaler.fit(X2_train_pca)

X2_train_pca_sca = scaler.transform(X2_train_pca)
```

3. Finally, I test the new feature set with the validated data set. I got a 12.416 mean squared error with 66 features.

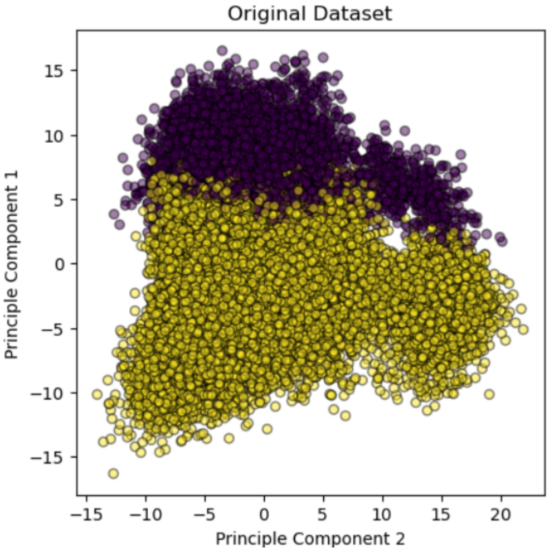
Label_3

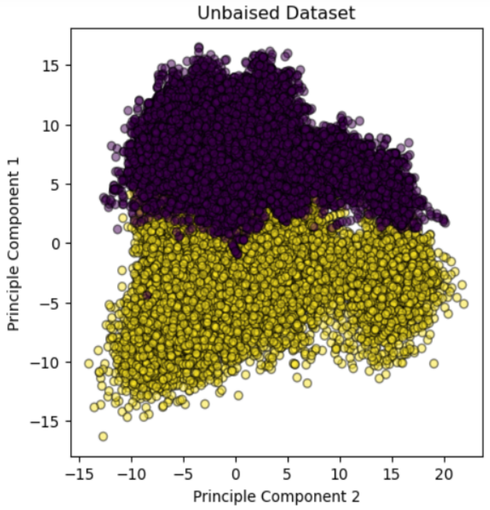
1. As the first step, I used PCA to transform the feature set. By choosing 0.98 as the variance I could reduce the number of features to 88 without losing the accuracy.

2. Then I checked the data set to identify if there was any bias in the data set. There was a significant number of 1s in the training data set compared with 0s.



3. I used ‘imblearn’ library to perform oversampling and undersampling to generate an unbiased dataset.





4. Then I trained the model using a reduced data set. After that, I remove less important features for the prediction utilizing RandomForestClassifier's feature_importances_ attribute. Using this I could reduce the number of features to 12 features.

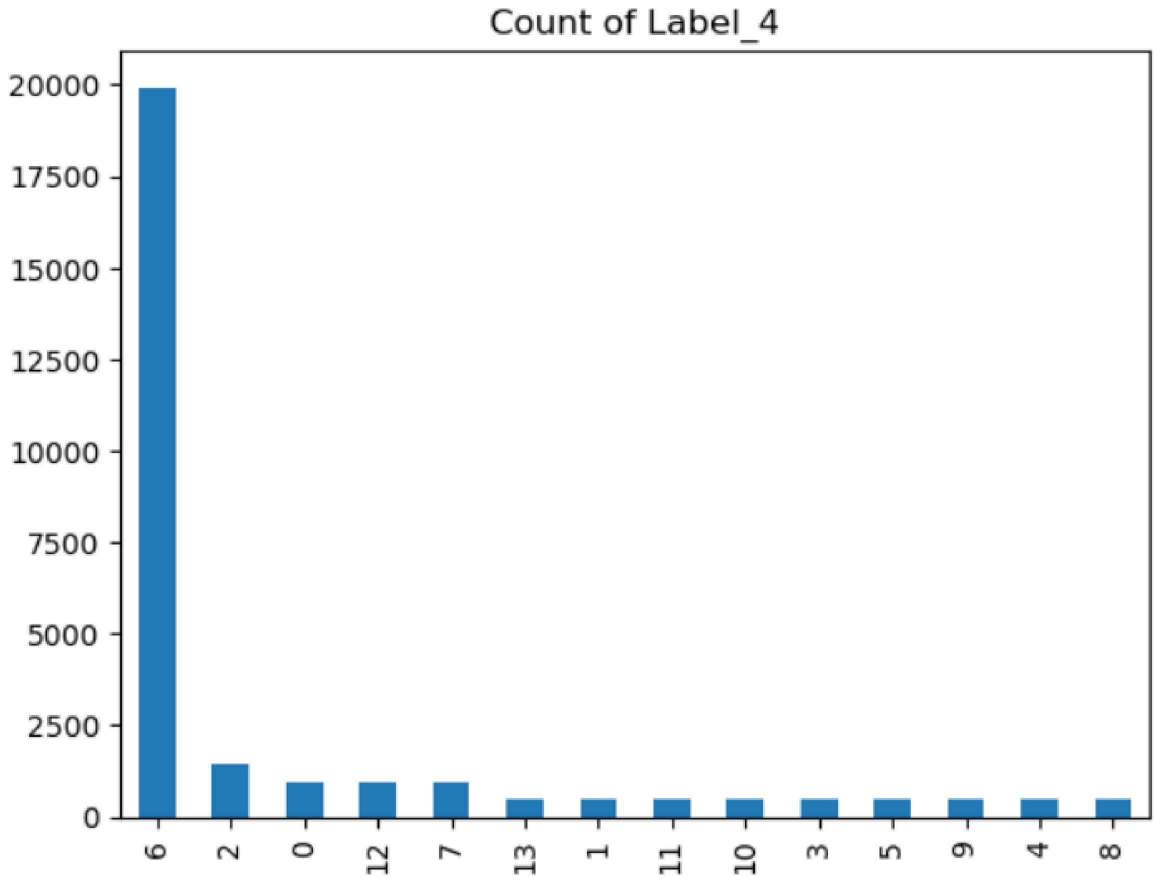
```
importance = rf.feature_importances_
columns_to_delete = []
for i,v in enumerate(importance):
    if v < 0.008:
        columns_to_delete.append(i)
train_reduced = np.delete(X3_train_pca, columns_to_delete,
axis=1)
test_reduced = np.delete(X3_test_pca, columns_to_delete,
axis=1)
test_reduced_ = np.delete(X3_test_pca_, columns_to_delete,
axis=1)
Train_reduced.shape
```

5. Finally, I test the new feature set with the validated data set. I got 0.99 accuracy with 12 features.

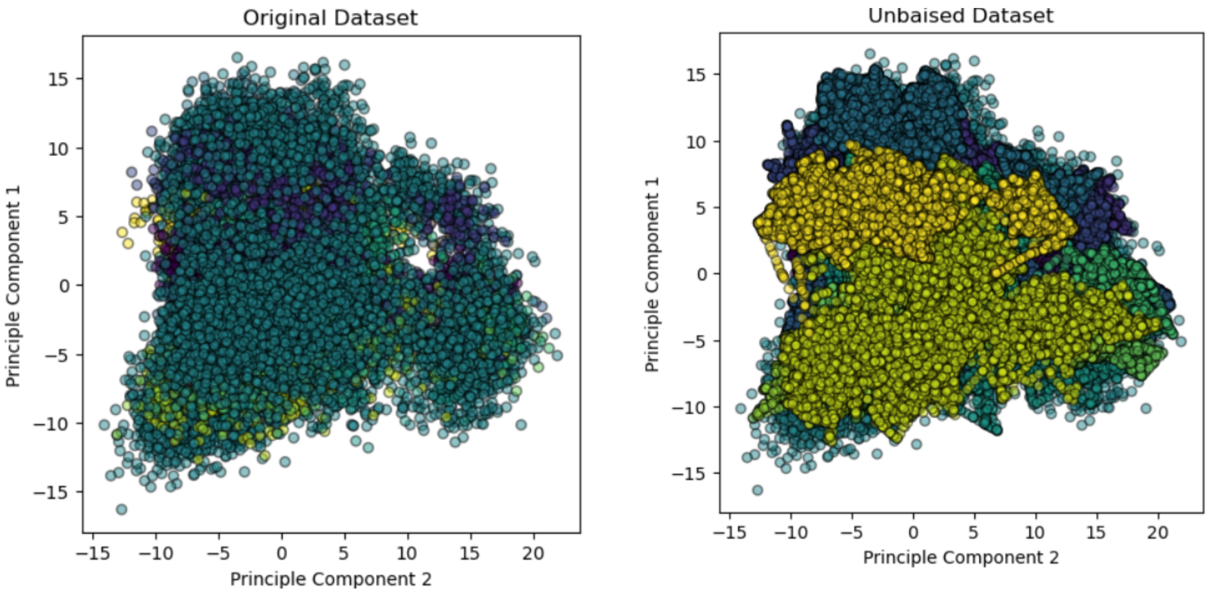
Label_4

1. As the first step, I used PCA to transform the feature set. By choosing 0.98 as the variance I could reduce the number of features to 88 without losing the accuracy.

2. Then I checked the data set to identify if there was any bias in the data set. There was a significant number of 6s in the training data set compared to other values.



3. I used ‘imblearn’ library to perform oversampling and undersampling to generate an unbiased dataset.



4. Then I trained the model using a reduced data set. After that, I remove less important features for the prediction utilizing RandomForestClassifier’s feature_importances_ attribute. Using this I could reduce the number of features to 40 features.

```
importance = rf.feature_importances_
columns_to_delete = []
for i,v in enumerate(importance):
    if v < 0.008:
        columns_to_delete.append(i)
train_reduced = np.delete(X4_train_pca, columns_to_delete,
axis=1)
test_reduced = np.delete(X4_test_pca, columns_to_delete,
axis=1)
test_reduced_ = np.delete(X4_test_pca_, columns_to_delete,
axis=1)
train_reduced.shape
```

5. Finally, I test the new feature set with the validated data set. I got 0.96 accuracy with 40 features.