

TUGAS KELOMPOK
PROGRAM STUDI INDEPENDEN
ORBIT FUTURE ACADEMY

Identitas Kelompok

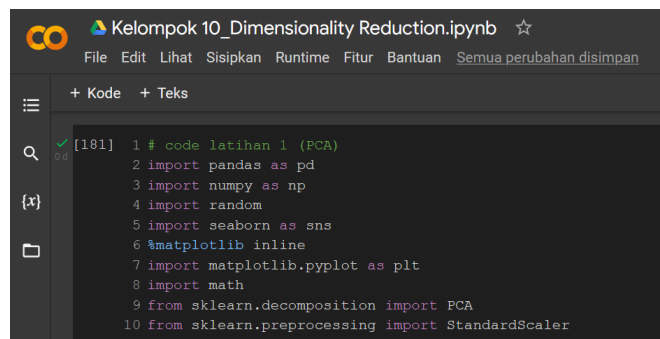
Kelompok : 10
Nama Anggota : Halomoan Filipus Simarmata (Jupyter XXI)
Diana Eka Riyani (Jupyter XXI)
Nyayu Chika Marselina (Jupyter XXI)
Sukma Imelda (Cordoba)
Athiya Shinta Wulandari (Cordoba)
Coach : Ipin Sugiyarto
Program : Foundations of AI and Life Skills for Gen-Z
Hari, Tanggal : Kamis, 31 Maret 2022

LATIHAN 1: Membuat model data reduction dengan menggunakan PCA (Principle Component Analysis), dengan ketentuan sebagai berikut:

1. Buatlah DataFrame dengan jumlah fitur 15
2. Generate DataFrame dengan bilangan float secara random, antara 0 dan 1, serta buat `n_component` berjumlah 2.
3. Membuat Label A, B, C, dan D (masing-masing jumlah 150 data), kemudian lanjutkan proses reduksi data model PCA sampai dengan visualisasi data dalam bentuk plot.

Penyelesaian:

1. Import library yang dibutuhkan terlebih dahulu

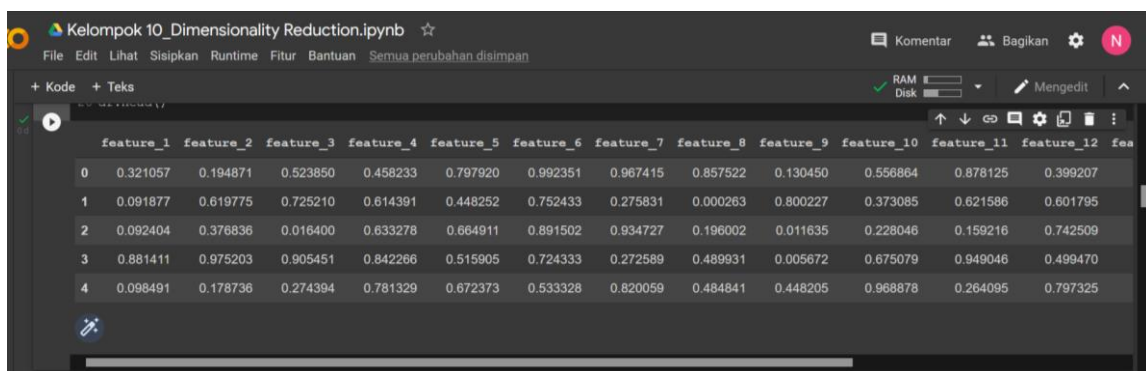


```
[181] 1 # code latihan 1 (PCA)
      2 import pandas as pd
      3 import numpy as np
      4 import random
      5 import seaborn as sns
      6 %matplotlib inline
      7 import matplotlib.pyplot as plt
      8 import math
      9 from sklearn.decomposition import PCA
     10 from sklearn.preprocessing import StandardScaler
```

2. Membuat DataFrame di mana berisikan 15 feature yang masing-masingnya memiliki jumlah data sebanyak 150 data serta terdiri dari label A, B, C, dan D.

```
1 #Membuat DataFrame
2 data = {'feature_1': [random.uniform(0, 1) for i in range(150)],
3         'feature_2': [random.uniform(0, 1) for i in range(150)],
4         'feature_3': [random.uniform(0, 1) for i in range(150)],
5         'feature_4': [random.uniform(0, 1) for i in range(150)],
6         'feature_5': [random.uniform(0, 1) for i in range(150)],
7         'feature_6': [random.uniform(0, 1) for i in range(150)],
8         'feature_7': [random.uniform(0, 1) for i in range(150)],
9         'feature_8': [random.uniform(0, 1) for i in range(150)],
10        'feature_9': [random.uniform(0, 1) for i in range(150)],
11        'feature_10': [random.uniform(0, 1) for i in range(150)],
12        'feature_11': [random.uniform(0, 1) for i in range(150)],
13        'feature_12': [random.uniform(0, 1) for i in range(150)],
14        'feature_13': [random.uniform(0, 1) for i in range(150)],
15        'feature_14': [random.uniform(0, 1) for i in range(150)],
16        'feature_15': [random.uniform(0, 1) for i in range(150)],
17        'label': [random.choice(['A','B','C','D']) for i in range(150)]}
18
19 df = pd.DataFrame(data)
20 df.head()
```

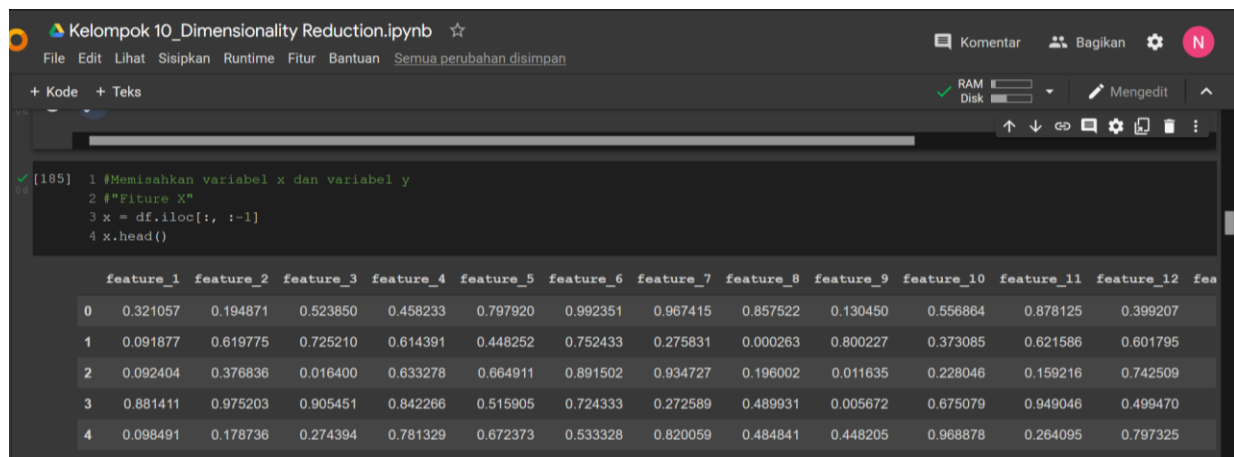
Hasil running coding di atas:



	feature_1	feature_2	feature_3	feature_4	feature_5	feature_6	feature_7	feature_8	feature_9	feature_10	feature_11	feature_12	feature_13	feature_14	feature_15	label
0	0.321057	0.194871	0.523850	0.458233	0.797920	0.992351	0.967415	0.857522	0.130450	0.556864	0.878125	0.399207	0.621586	0.601795	0.742509	A
1	0.091877	0.619775	0.725210	0.614391	0.448252	0.752433	0.275831	0.000263	0.800227	0.373085	0.621586	0.601795	0.621586	0.601795	0.742509	B
2	0.092404	0.376836	0.016400	0.633278	0.664911	0.891502	0.934727	0.196002	0.011635	0.228046	0.159216	0.742509	0.621586	0.601795	0.742509	C
3	0.881411	0.975203	0.905451	0.842266	0.515905	0.724333	0.272589	0.489931	0.005672	0.675079	0.949046	0.499470	0.621586	0.601795	0.742509	D
4	0.098491	0.178736	0.274394	0.781329	0.672373	0.533328	0.820059	0.484841	0.448205	0.968878	0.264095	0.797325	0.621586	0.601795	0.742509	A

Karena menggunakan df.head(), maka hanya menampilkan 5 baris pertama dari DataFrame.

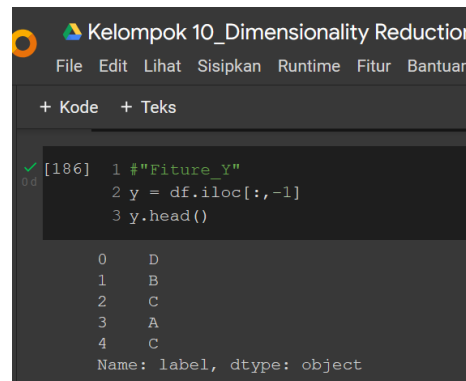
3. Memisahkan variabel x dari DataFrame yang dibentuk. Pada bagian ini, menampilkan 5 baris data pertama dari DataFrame yang dibuat dengan variabel x yang telah dipisahkan.



```
[185] 1 #Memisahkan variabel x dan variabel y
      2 #Feature X
      3 x = df.iloc[:, :-1]
      4 x.head()
```

	feature_1	feature_2	feature_3	feature_4	feature_5	feature_6	feature_7	feature_8	feature_9	feature_10	feature_11	feature_12	feature_13	feature_14	feature_15
0	0.321057	0.194871	0.523850	0.458233	0.797920	0.992351	0.967415	0.857522	0.130450	0.556864	0.878125	0.399207	0.621586	0.601795	0.742509
1	0.091877	0.619775	0.725210	0.614391	0.448252	0.752433	0.275831	0.000263	0.800227	0.373085	0.621586	0.601795	0.621586	0.601795	0.742509
2	0.092404	0.376836	0.016400	0.633278	0.664911	0.891502	0.934727	0.196002	0.011635	0.228046	0.159216	0.742509	0.621586	0.601795	0.742509
3	0.881411	0.975203	0.905451	0.842266	0.515905	0.724333	0.272589	0.489931	0.005672	0.675079	0.949046	0.499470	0.621586	0.601795	0.742509
4	0.098491	0.178736	0.274394	0.781329	0.672373	0.533328	0.820059	0.484841	0.448205	0.968878	0.264095	0.797325	0.621586	0.601795	0.742509

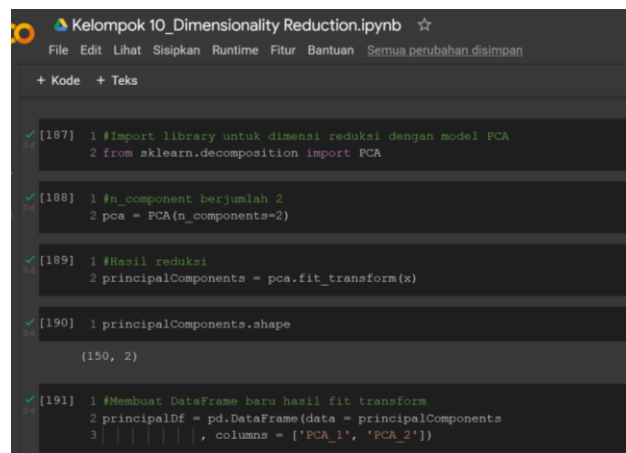
- Menampilkan 5 baris data pertama dari DataFrame dengan variabel y yang telah dipisahkan.



```
[186] 1 #Future_Y
      2 y = df.iloc[:, -1]
      3 y.head()

0    D
1    B
2    C
3    A
4    C
Name: label, dtype: object
```

- Mengimport library yang akan digunakan untuk mereduksi dimensi dengan model PCA. Kemudian membentuk n_components berjumlah 2 yang disimpan di variabel pca. Lalu, melakukan reduksi yang mana hasilnya disimpan di dalam variabel principalComponents. Lalu mencetak ukuran dari hasil reduksi menggunakan principalComponents.shape. Setelah itu, membuat DataFrame yang baru (hasil dari fit transform) yang disimpan di dalam variabel principalDf. Serta menampilkan 5 baris pertama dari DataFrame baru hasil fit transform.



```
[187] 1 #Import library untuk dimensi reduksi dengan model PCA
      2 from sklearn.decomposition import PCA

[188] 1 #n_component berjumlah 2
      2 pca = PCA(n_components=2)

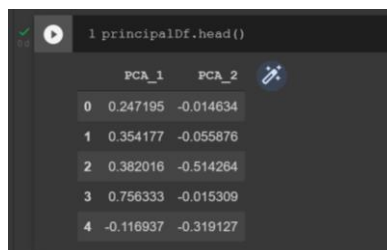
[189] 1 #Hasil reduksi
      2 principalComponents = pca.fit_transform(x)

[190] 1 principalComponents.shape

(150, 2)

[191] 1 #Membuat DataFrame baru hasil fit transform
      2 principalDf = pd.DataFrame(data = principalComponents
      3 | | | | | , columns = ['PCA_1', 'PCA_2'])
```

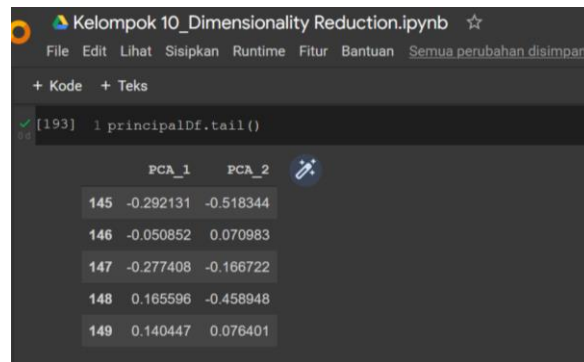
- Menampilkan 5 baris pertama dan 5 baris terakhir DataFrame baru hasil fit transform. Berikut tampilan 5 baris pertama DataFrame baru:



```
1 principalDf.head()

   PCA_1  PCA_2
0  0.247195 -0.014634
1  0.354177 -0.055876
2  0.382016 -0.514264
3  0.756333 -0.015309
4 -0.116937 -0.319127
```

Berikut tampilan 5 baris terakhir DataFrame baru:



Kelompok 10_Dimensionality Reduction.ipynb

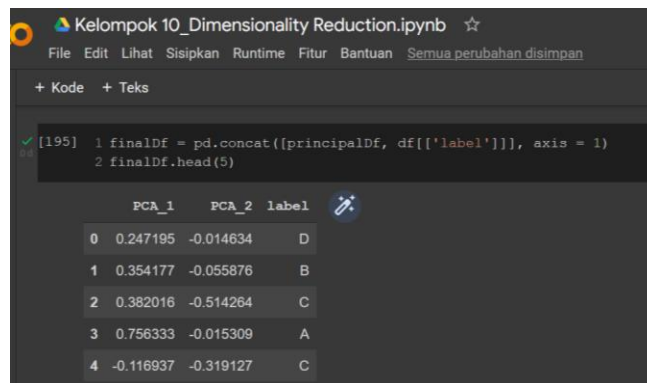
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

```
[193] 1 principalDf.tail()
```

	PCA_1	PCA_2
145	-0.292131	-0.518344
146	-0.050852	0.070983
147	-0.277408	-0.166722
148	0.165596	-0.458948
149	0.140447	0.076401

- Menampilkan 5 baris pertama final DataFrame yang isinya meliputi nilai PCA1, PCA2, dan label.



Kelompok 10_Dimensionality Reduction.ipynb

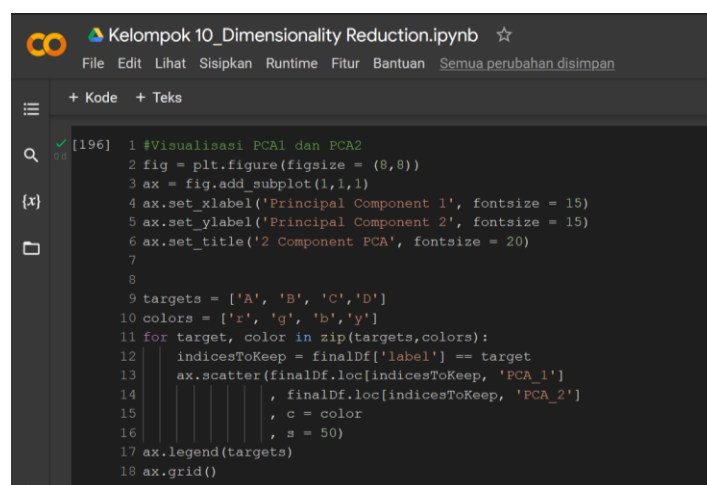
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

```
[195] 1 finalDf = pd.concat([principalDf, df[['label']], axis = 1)
      2 finalDf.head(5)
```

	PCA_1	PCA_2	label
0	0.247195	-0.014634	D
1	0.354177	-0.055876	B
2	0.382016	-0.514264	C
3	0.756333	-0.015309	A
4	-0.116937	-0.319127	C

- Membuat visualisasi PCA1 dan PCA2



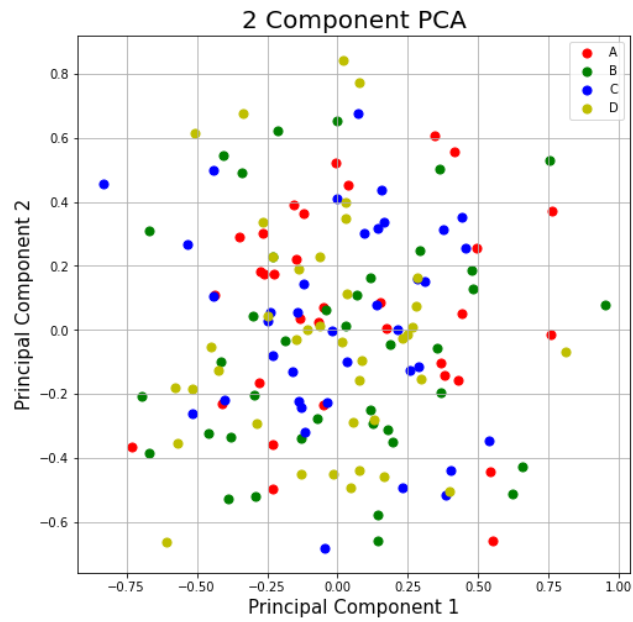
Kelompok 10_Dimensionality Reduction.ipynb

File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

```
[196] 1 #Visualisasi PCA1 dan PCA2
      2 fig = plt.figure(figsize = (8,8))
      3 ax = fig.add_subplot(1,1,1)
      4 ax.set_xlabel('Principal Component 1', fontsize = 15)
      5 ax.set_ylabel('Principal Component 2', fontsize = 15)
      6 ax.set_title('2 Component PCA', fontsize = 20)
      7
      8
      9 targets = ['A', 'B', 'C','D']
     10 colors = ['r', 'g', 'b','y']
     11 for target, color in zip(targets,colors):
     12     indicesToKeep = finalDf['label'] == target
     13     ax.scatter(finalDf.loc[indicesToKeep, 'PCA_1']
     14               , finalDf.loc[indicesToKeep, 'PCA_2']
     15               , c = color
     16               , s = 50)
     17 ax.legend(targets)
     18 ax.grid()
```

Berikut running code untuk visualisasi PCA1 dan PCA2:



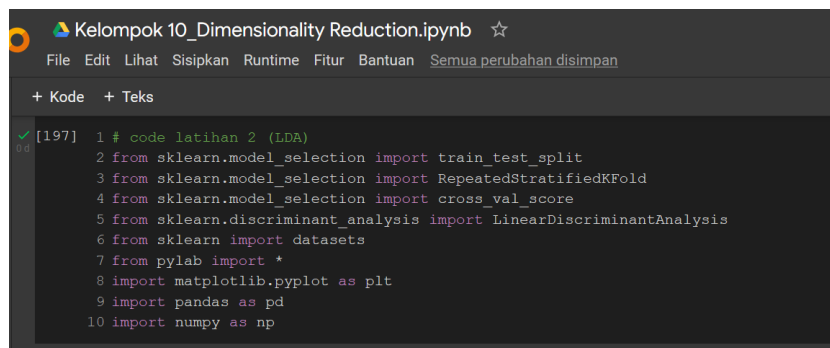
Dari visualisasi di atas, plot berwarna merah untuk label A, plot berwarna hijau untuk label B, plot berwarna biru untuk label C, plot berwarna cream untuk label berwarna D.

LATIHAN 2: Membuat model data reduction dengan menggunakan LDA (Linear Discriminant Analysis) dengan ketentuan sebagai berikut:

1. Menggunakan dataset yang berbeda dari latihan yang kemarin.
2. Tampilkan DataFrame untuk variabel x dan y (data tabular). Kemudian, lanjutkan proses reduksi data model LDA sampai dengan visualisasi data dalam bentuk plot.

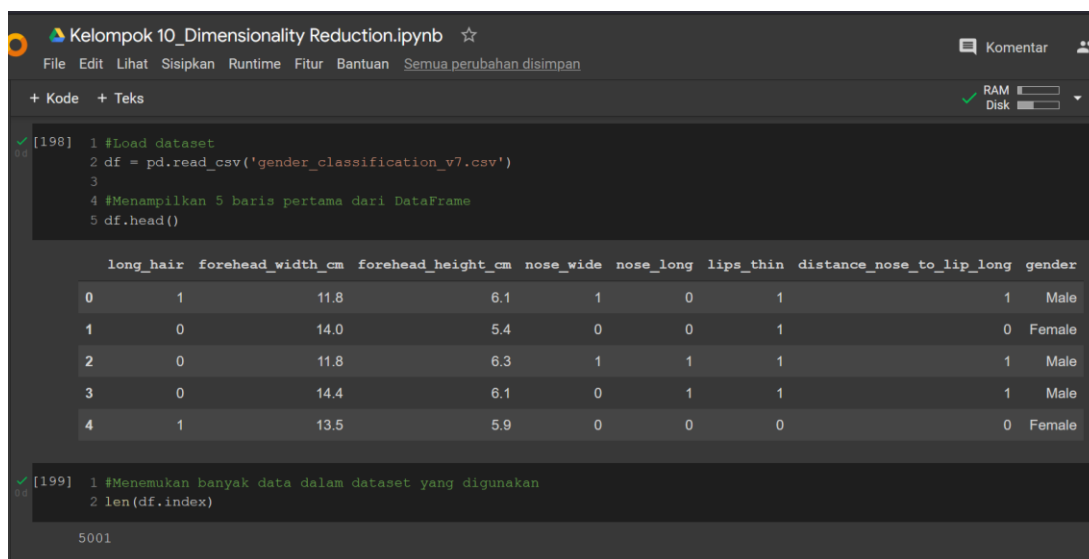
Penyelesaian:

1. Mengimport library yang akan digunakan



```
[197] 1 # code latihan 2 (LDA)
      2 from sklearn.model_selection import train_test_split
      3 from sklearn.model_selection import RepeatedStratifiedKFold
      4 from sklearn.model_selection import cross_val_score
      5 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
      6 from sklearn import datasets
      7 from pylab import *
      8 import matplotlib.pyplot as plt
      9 import pandas as pd
     10 import numpy as np
```

2. Membaca dataset yang akan digunakan ke dalam bentuk DataFrame, kemudian menampilkan 5 baris pertama dari DataFrame. Kelompok 10 menggunakan dataset pengklasifikasian gender. Selain itu, menemukan banyak data dalam dataset yang digunakan.



```
[198] 1 #Load dataset
      2 df = pd.read_csv('gender_classification_v7.csv')
      3
      4 #Menampilkan 5 baris pertama dari DataFrame
      5 df.head()
```

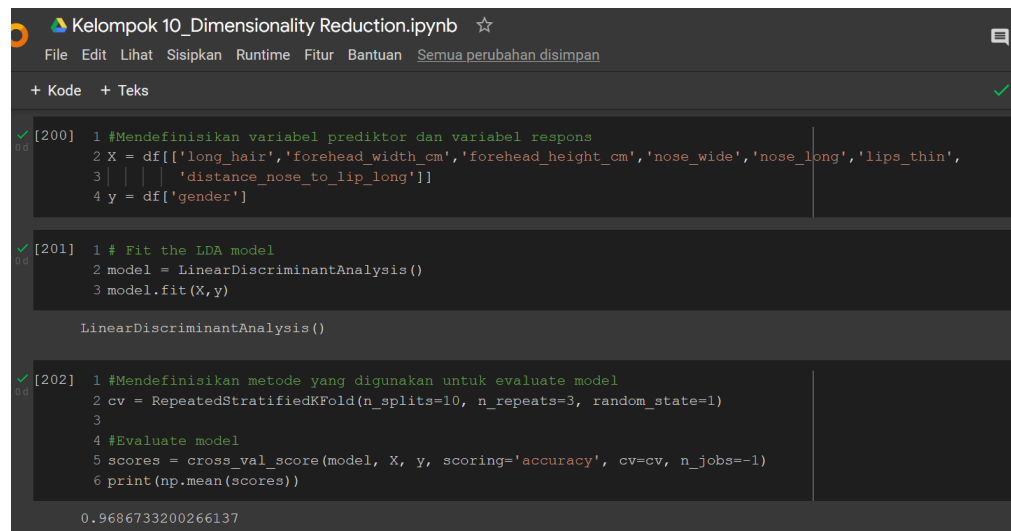
	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long	gender
0	1	11.8	6.1	1	0	1	1	Male
1	0	14.0	5.4	0	0	1	0	Female
2	0	11.8	6.3	1	1	1	1	Male
3	0	14.4	6.1	0	1	1	1	Male
4	1	13.5	5.9	0	0	0	0	Female

```
[199] 1 #Menemukan banyak data dalam dataset yang digunakan
      2 len(df.index)
```

5001

Dari hasil running code di atas, terdapat 5001 data pada dataset yang digunakan.

3. Mendefinisikan variabel predictor dan variabel response. Fit model LDA dari variabel X dan y. Kemudian mendefinisikan metode yang digunakan untuk meng-evaluate model lalu dilakukan evaluate model sehingga diperoleh nilai akurasi sebesar 97%.



```
Kelompok 10_Dimensionality Reduction.ipynb ☆
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

[200] 1 #Mendefinisikan variabel prediktor dan variabel respons
      2 X = df[['long_hair','forehead_width_cm','forehead_height_cm','nose_wide','nose_long','lips_thin',
      3 | | | 'distance_nose_to_lip_long']]
      4 y = df['gender']

[201] 1 # Fit the LDA model
      2 model = LinearDiscriminantAnalysis()
      3 model.fit(X,y)

      LinearDiscriminantAnalysis()

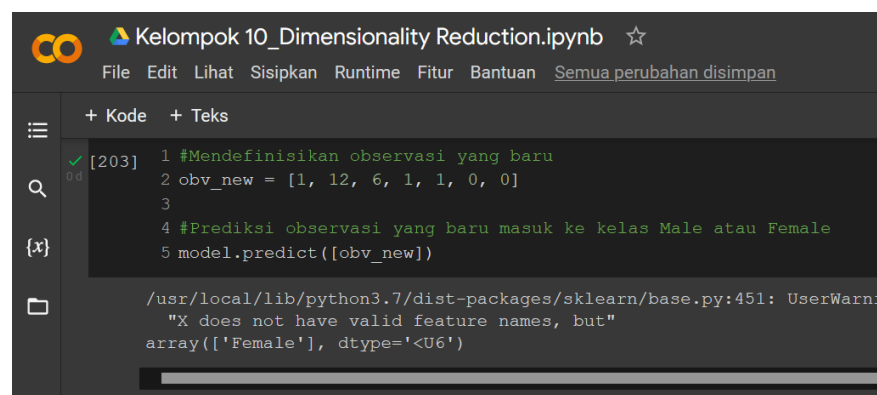
[202] 1 #Mendefinisikan metode yang digunakan untuk evaluate model
      2 cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
      3
      4 #Evaluate model
      5 scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
      6 print(np.mean(scores))

0.9686733200266137
```

4. Mendefinisikan observasi yang baru lalu melakukan prediksi dari observasi yang baru. Jika dimisalkan:

- long_hair = 1
- forehead_width_cm = 12
- forehead_height_cm = 6
- nose_wide = 1
- nose_long = 1
- lips_thin = 0
- distance_nose_to_lip_long = 0

Diperoleh hasil prediksinya, yakni Female



```
Kelompok 10_Dimensionality Reduction.ipynb ☆
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

[203] 1 #Mendefinisikan observasi yang baru
      2 obv_new = [1, 12, 6, 1, 1, 0, 0]
      3
      4 #Prediksi observasi yang baru masuk ke kelas Male atau Female
      5 model.predict([obv_new])

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning:
  "X does not have valid feature names, but"
array(['Female'], dtype='<U6')
```

5. Mendefinisikan data ke dalam plot, lalu membuat visualisasi LDA plot. Lalu, menambahkan legend pada plot dan menampilkan LDA plot yang sudah dibentuk.

```
Kelompok 10_Dimensionality Reduction.ipynb ☆
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

1 #Mendefinisikan data ke dalam plot
2 model = LinearDiscriminantAnalysis()
3 data_plot = model.fit(X, y).transform(X)

[205] 1 #Membuat LDA plot
      2 plt.figure()
      3 colors = ['red', 'green', 'blue']
      4 lw = 2
      5 target_names = ['Male', 'Female']
      6
      7 for color, i, gender in zip(colors, [0, 1, 2], target_names):
      8 | plt.scatter(data_plot[y == gender, 0], data_plot[y == gender, 1], alpha=.8, color=color,
      9 | | | | | label=gender)
      10
      11 #Menambahkan legend pada plot
      12 plt.legend(loc='best', shadow=False, scatterpoints=1)
      13
      14 #Display LDA plot
      15 plt.show()
```

Berikut hasil visualisasi LDA plot:

