p1

August 7, 2018

1 DMG2 Assignment

Fisher Discriminant Analysis

1.1 Problem 1

```
In [1]: import pandas as pd
        import numpy as np
        import os
        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set_style('ticks')

        from sklearn.preprocessing import StandardScaler
        from sklearn.decomposition import PCA
        from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
```

```
In [4]: DATA_DIR='/home/jishnu/Documents/ISB/Term3/dmg2/assignments/hw_assignment1/dmg2/dataset
        iris_train = pd.read_csv(os.path.join(DATA_DIR,'iris/train.csv'))
        iris_test = pd.read_csv(os.path.join(DATA_DIR,'iris/test.csv'))
```

```
In [5]: iris_train.drop(labels='Unnamed: 0',axis=1,inplace=True)
        iris_train.head(5)
        iris_test.drop(labels='Unnamed: 0',axis=1,inplace=True)
        iris_test.head(5)
```

Out[5]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	4.7	3.2	1.3	0.2	setosa
1	4.6	3.1	1.5	0.2	setosa
2	5.4	3.9	1.7	0.4	setosa
3	4.6	3.4	1.4	0.3	setosa
4	5.0	3.4	1.5	0.2	setosa

```
In [6]: x = iris_train.iloc[:,:4]
        y = iris_train.iloc[:,4]
```

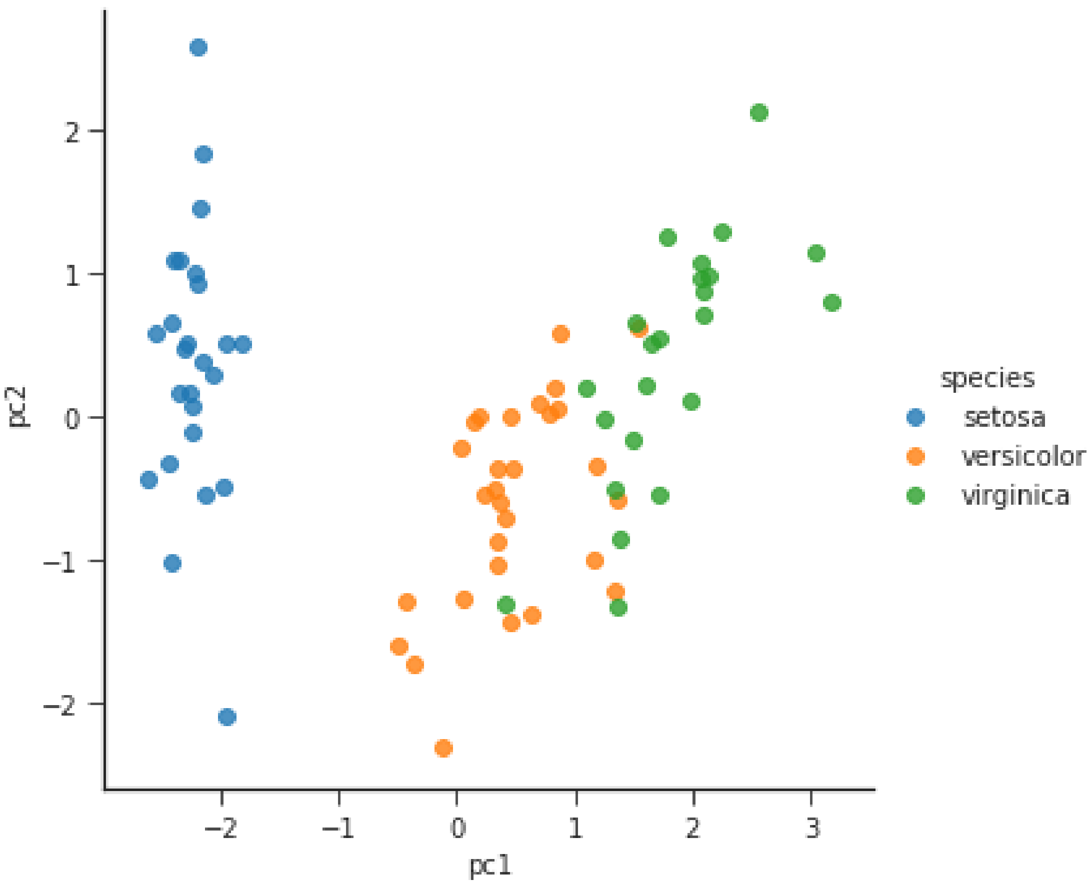
```
In [7]: # Standardizing feature values
        x = StandardScaler().fit_transform(x)
```

```
In [8]: # Applying PCA
        pc = PCA(n_components=2).fit_transform(x)
        pc_df = pd.DataFrame(data=pc,columns=['pc1','pc2'])
        pc_df['species'] = y
        pc_df.head(5)
```

Out[8]:

	pc1	pc2	species
0	-2.281237	0.509187	setosa
1	-2.125167	-0.554033	setosa
2	-2.410055	0.646865	setosa
3	-2.414650	-1.012994	setosa
4	-2.362836	0.157713	setosa

```
In [9]: sns.lmplot(x='pc1',y='pc2',hue='species',data=pc_df,fit_reg=False)
        plt.show()
```



It is seen that **versicolor** and **virginica** are the two “more” similar species, by plotting the 2-D principal components.

Creating meta-class

```
In [10]: def return_class(row):
             if row[4] == 'setosa':
                 return 'class_3'
             else:
                 return 'class_4'
         y_3_4 = iris_train.apply(lambda row : return_class(row),axis=1)
```

Fitting Fisher projection by discriminating classes 3 and 4

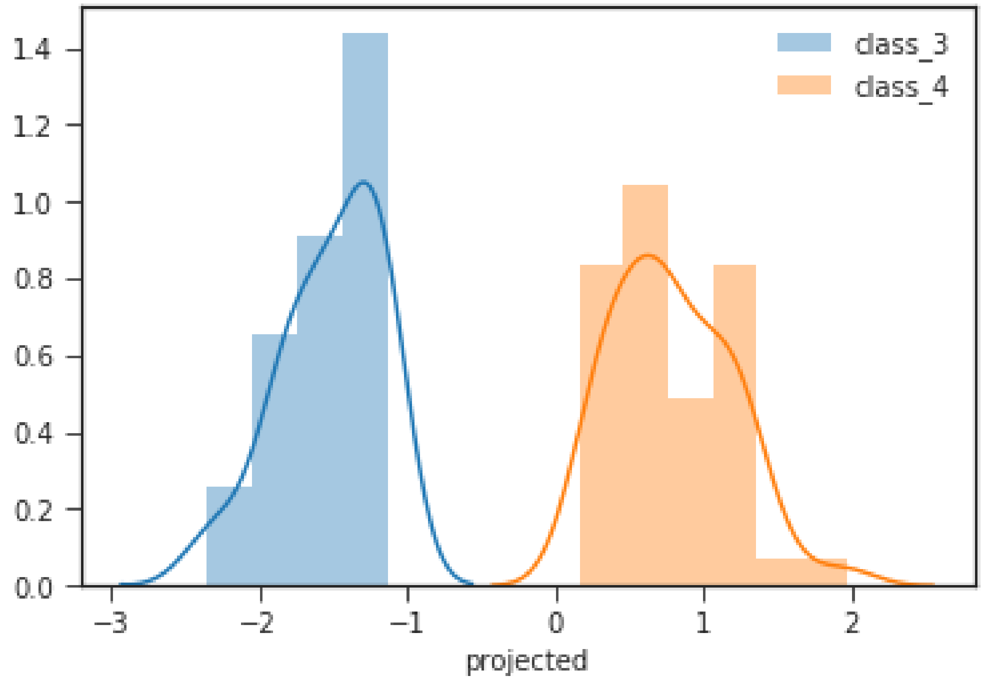
```
In [11]: fisher_c34 = LinearDiscriminantAnalysis(solver='eigen',n_components=2).fit(x,y_3_4)
         fisher_c34.coef_

Out[11]: array([[-0.35777222, -0.79744203,  2.17379404,  0.45444277]])
```

Fitting Fisher projection by discriminating classes 1 and 2

```
In [13]: iris_train_1_2 = iris_train.loc[iris_train['Species'].isin(['versicolor','virginica'])
         x_1_2 = iris_train_1_2.iloc[:,:4]
         y_1_2 = iris_train_1_2.iloc[:,4]
         x_1_2 = StandardScaler().fit_transform(x_1_2)

In [14]: fisher_c12 = LinearDiscriminantAnalysis(solver='eigen',n_components=2).fit(x_1_2,y_1_
         fisher_c12.coef_

Out[14]: array([[-0.40330778, -0.40343313,  1.12831703,  0.94488192]])
```

Projecting test data to above two projections

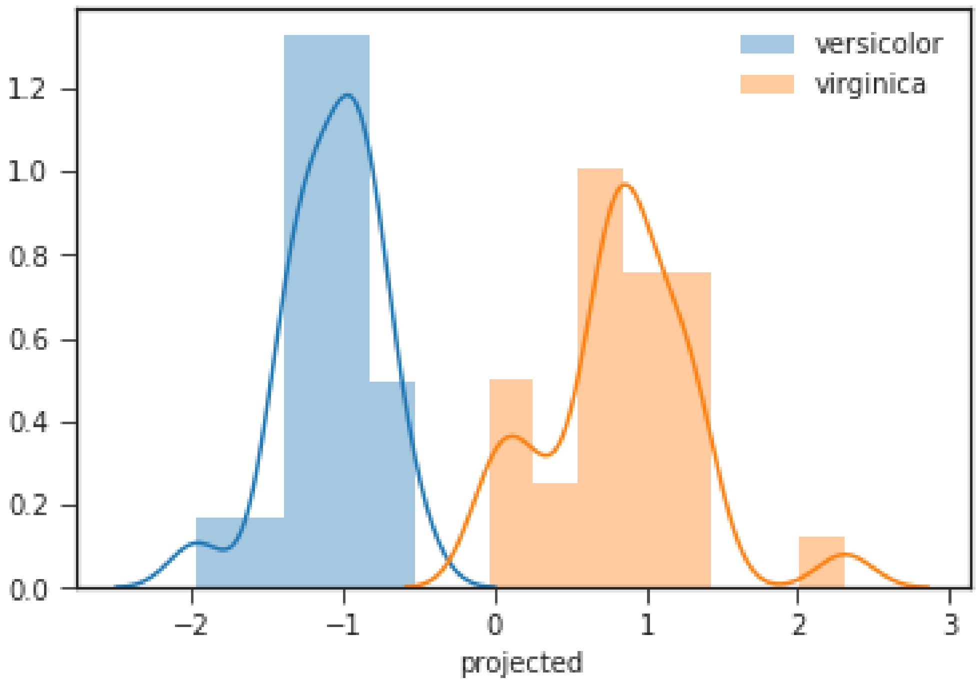
```
In [15]: x_test = StandardScaler().fit_transform(iris_test.iloc[:,:4])
         y_test_3_4 = iris_test.apply(lambda row : return_class(row),axis=1)

In [16]: fisher_proj_3_4 = pd.DataFrame(fisher_c34.transform(x_test),columns=['projected'])
         fisher_proj_3_4['class'] = y_test_3_4

In [17]: sns.distplot(fisher_proj_3_4.loc[fisher_proj_3_4['class'] == 'class_3']['projected'],l
         sns.distplot(fisher_proj_3_4.loc[fisher_proj_3_4['class'] == 'class_4']['projected'],l
         plt.legend()
         plt.show()

/home/jishnu/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning:
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
/home/jishnu/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning:
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



1.1.1 Finding fisher discriminant value

```
In [23]: mean_1 = np.mean(fisher_proj_3_4.loc[fisher_proj_3_4['class'] == 'class_3'])
         mean_2 = np.mean(fisher_proj_3_4.loc[fisher_proj_3_4['class'] == 'class_4'])

         sd_1 = np.std(fisher_proj_3_4.loc[fisher_proj_3_4['class'] == 'class_3'])
         sd_2 = np.std(fisher_proj_3_4.loc[fisher_proj_3_4['class'] == 'class_4'])

         fd_3_4 = (mean_1 - mean_2)**2 / (sd_1**2 + sd_2**2)
         np.round(fd_3_4,4)

Out[23]: projected    19.6663
         dtype: float64

In [25]: iris_test_1_2 = iris_test.loc[iris_test['Species'].isin(['versicolor','virginica'])]
         x_test_1_2 = iris_test_1_2.iloc[:,:4]
         y_test_1_2 = iris_test_1_2.iloc[:,4]
         x_test_1_2 = StandardScaler().fit_transform(x_test_1_2)
         fisher_proj_1_2 = pd.DataFrame(fisher_c12.transform(x_test_1_2),columns=['projected']
         fisher_proj_1_2['class'] = iris_test_1_2.iloc[:,4].values

In [26]: sns.distplot(fisher_proj_1_2.loc[fisher_proj_1_2['class'] == 'versicolor']['projected
         sns.distplot(fisher_proj_1_2.loc[fisher_proj_1_2['class'] == 'virginica']['projected']
         plt.legend()
         plt.show()
```

```
/home/jishnu/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning:
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
/home/jishnu/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning:
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



1.1.2 Finding fisher discriminant value

In [28]:
```
mean_1 = np.mean(fisher_proj_1_2.loc[fisher_proj_1_2['class'] == 'versicolor'])
mean_2 = np.mean(fisher_proj_1_2.loc[fisher_proj_1_2['class'] == 'virginica'])

sd_1 = np.std(fisher_proj_1_2.loc[fisher_proj_1_2['class'] == 'versicolor'])
sd_2 = np.std(fisher_proj_1_2.loc[fisher_proj_1_2['class'] == 'virginica'])

fd_3_4 = (mean_1 - mean_2)**2 / (sd_1**2 + sd_2**2)
np.round(fd_3_4,4)
```

Out[28]:
```
projected    10.7755
dtype: float64
```

1.2 Observations

Using scatter plots, we found that “versicolor” and “virginica” species are the most similar of the three species.

We combined these two species into one meta-class, and created the first fisher projection by discriminating the meta-class and setosa classes. We projected the test data points to this projection vector, and the histogram shows good seperation between the two classes.

We then created the second fisher projection by discriminating the two most similar species, versicolor and virginica. We projected the test data(filtering those data points for these two species) on the projection vector. The histogram of the projected values show a clear seperation for the two classes, which was not evident in the PCA projections.

We have therefore, found the two vectors which can be used to discriminate all three classes in the IRIS dataset. The first vector can be used to discriminate setosa and the combination of versicolor and virginica, and the second vector can be used to discriminate versicolor and virginica classes effectively.