WuBenjaminAssignment4

March 14, 2022

1 CS156 (Introduction to AI), Spring 2022

2 Homework 4 submission

2.0.1 Roster Name: Benjamin Wu

2.0.2 Student ID: 013607880

2.0.3 Email address: benjamin.wu01@sjsu.edu

2.1 References and sources

https://www.geeksforgeeks.org/add-a-new-column-in-pandas-data-frame-using-a-dictionary/

2.2 Solution

Load libraries and set random number generator seed

[ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.svm import LinearSVC
from sklearn.metrics import plot_confusion_matrix
from sklearn.model_selection import cross_val_score
```

[ ]:

```
np.random.seed(42)

colors = {"Breast": '#4287f5',
 "Bladder": '#19c5e3',
 "Colon": '#80d941',
 "Glioblastoma": '#179933',
 "Head&Neck": '#f07e78',
 "Kidney": '#f01e13',
 "Leukemia": '#f0841f',
 "LungAdeno": '#db5209',
 "LungSquamous": '#ce8ced',
 "Ovarian": '#551075',
"Rectal": '#e3d329',
"Uterine": '#cc3423'
}
```

Code the solution

Loading data

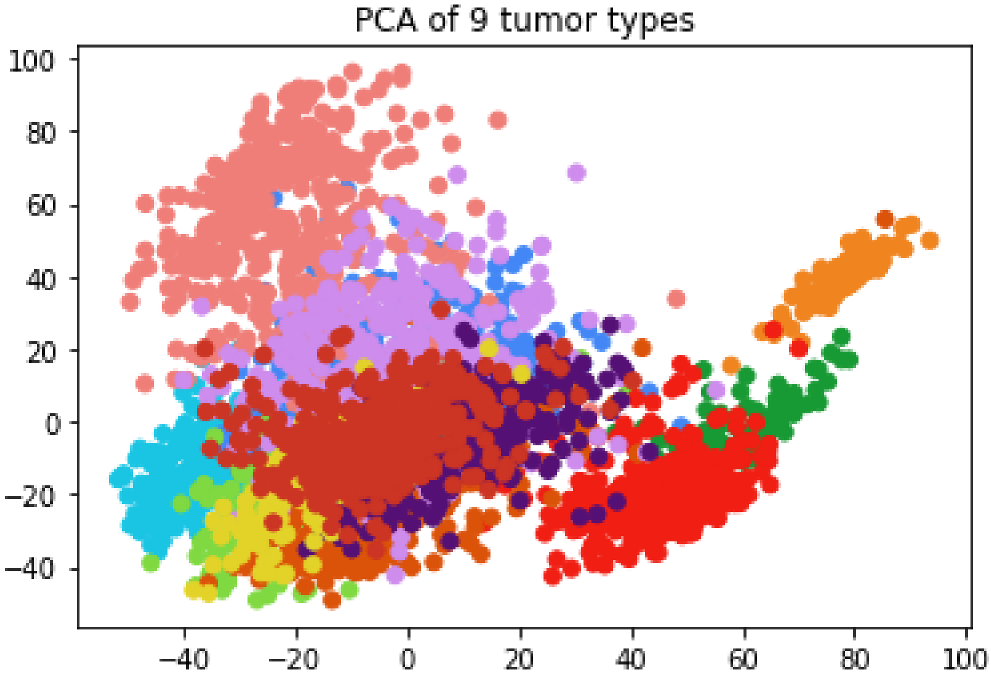
[ ]:

```
df = pd.read_csv("homework4_input_data.csv")
```

Plot PCA

[ ]:

```
pca = PCA(n_components = 2).fit_transform(df.iloc[:,1:-1])
pdf = pd.DataFrame(data=pca, columns=["X", "Y"])
classdf = df.Class
pdf = pd.concat([pdf, classdf], axis=1)
pdf["Color"] = pdf["Class"].map(colors)
plt.scatter(pdf.X, pdf.Y, color=pdf.Color)
plt.title("PCA of 9 tumor types")
plt.show()
```



Normalize data, break into sets, and perform cross-validation

[ ]:

```
X = df.iloc[:,1:-1]
Y = df.Class
```

```
scaler = StandardScaler()
X_rescaled = scaler.fit_transform(X)

X_train, X_test, Y_train, Y_test = train_test_split(X_rescaled, Y, test_size=0.
 ↪2, random_state=0, stratify=Y)

model = LinearSVC(multi_class='ovr', class_weight='balanced')

results = cross_val_score(model, X_train, Y_train)
```

Cross-validation results

```
[ ]: print("Individual cross-validation accuracies: " + str(results))
     total = 0
     for i in results:
         total = total + i
     total = total / len(results)

     print("Mean cross validation accuracy: {:.5f}".format(total))
```

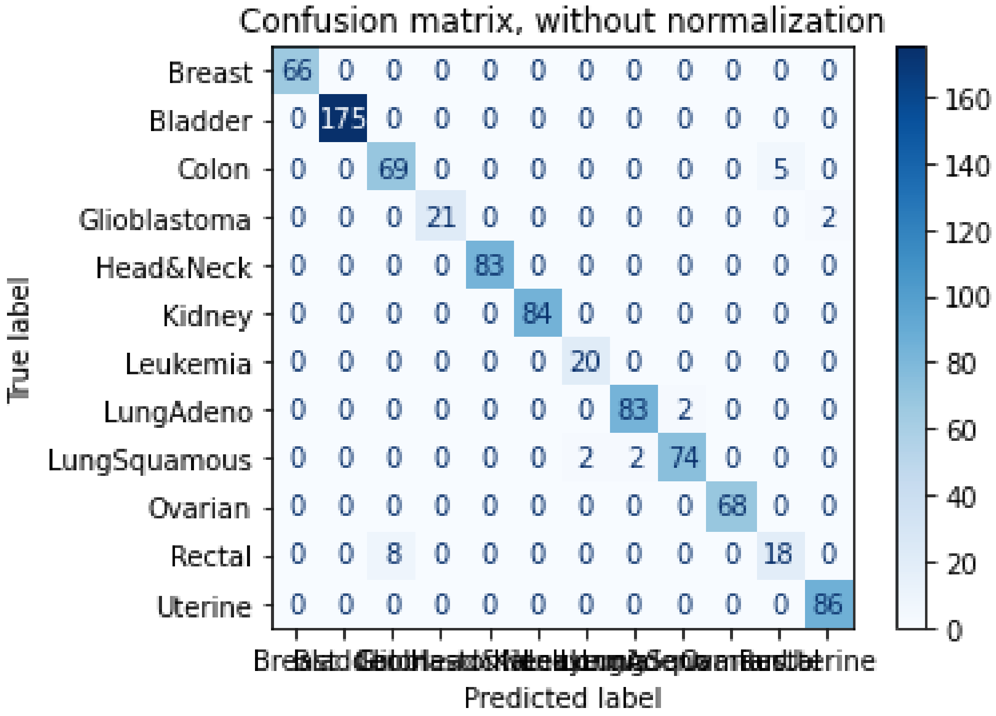
```
Individual cross-validation accuracies: [0.96 0.97 0.96 0.97 0.97]
Mean cross validation accuracy: 0.96396
```

Training final model, computing accuracy, and plotting confusion matrices

```
[ ]: model.fit(X_train, Y_train)

     print('Accuracy of linear SVC on training set: {:.2f}'.format(model.
      ↪score(X_train, Y_train)))
     print('Accuracy of linear SVC on test set: {:.2f}'.format(model.score(X_test,␣
      ↪Y_test)))


     np.set_printoptions(precision=2)
     titles_options = [("Confusion matrix, without normalization", None),
                       ("Normalized confusion matrix", 'true')]
     for title, normalize in titles_options:
         disp = plot_confusion_matrix(model, X_test, Y_test,
                                      display_labels=colors.keys(),
                                      cmap=plt.cm.Blues,
                                      normalize=normalize)
         disp.ax_.set_title(title)

     plt.show()
```

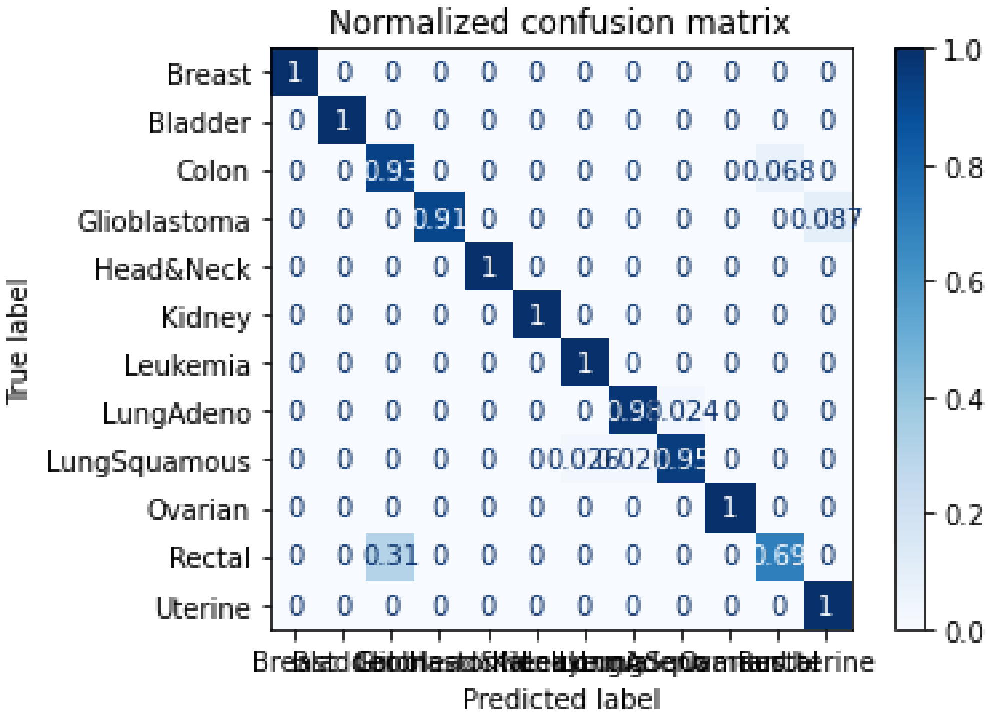
```
Accuracy of linear SVC on training set: 1.00
Accuracy of linear SVC on test set: 0.98
```

```
C:\Users\benja\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
plot_confusion_matrix is deprecated; Function `plot_confusion_matrix` is
deprecated in 1.0 and will be removed in 1.2. Use one of the class methods:
ConfusionMatrixDisplay.from_predictions or
ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
C:\Users\benja\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
plot_confusion_matrix is deprecated; Function `plot_confusion_matrix` is
deprecated in 1.0 and will be removed in 1.2. Use one of the class methods:
ConfusionMatrixDisplay.from_predictions or
ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```



Confusion matrix, without normalization

Breast	66	0	0	0	0	0	0	0	0	0	0	0
Bladder	0	175	0	0	0	0	0	0	0	0	0	0
Colon	0	0	69	0	0	0	0	0	0	0	5	0
Glioblastoma	0	0	0	21	0	0	0	0	0	0	0	2
Head&Neck	0	0	0	0	83	0	0	0	0	0	0	0
Kidney	0	0	0	0	0	84	0	0	0	0	0	0
Leukemia	0	0	0	0	0	0	20	0	0	0	0	0
LungAdeno	0	0	0	0	0	0	0	83	2	0	0	0
LungSquamous	0	0	0	0	0	0	2	2	74	0	0	0
Ovarian	0	0	0	0	0	0	0	0	0	68	0	0
Rectal	0	0	8	0	0	0	0	0	0	0	18	0
Uterine	0	0	0	0	0	0	0	0	0	0	0	86


Normalized confusion matrix
Breast 1 0 0 0 0 0 0 0 0 0 0 0
Bladder 0 1 0 0 0 0 0 0 0 0 0 0
Colon 0 0 0.93 0 0 0 0 0 0 0 0.068 0
Glioblastoma 0 0 0 0.91 0 0 0 0 0 0 0 0.087
Head&Neck 0 0 0 0 1 0 0 0 0 0 0 0
Kidney 0 0 0 0 0 1 0 0 0 0 0 0
Leukemia 0 0 0 0 0 0 1 0 0 0 0 0
LungAdeno 0 0 0 0 0 0 0 0.98 0.024 0 0 0
LungSquamous 0 0 0 0 0 0 0.026 0.026 0.95 0 0 0
Ovarian 0 0 0 0 0 0 0 0 0 1 0 0
Rectal 0 0 0.31 0 0 0 0 0 0 0 0.69 0
Uterine 0 0 0 0 0 0 0 0 0 0 0 1
True label
Uterine
Predicted label
1.0
0.8
0.6
0.4
0.2
0.0