Assignment2

October 8, 2023

1 Assignment 2

1.0.1 Name: Yelisetty

1.0.2 Roll Number: 21CS30036

1.1 Importing Libraries

[2]:
```
# Base libraries
import pandas as pd
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns; sns.set()
import matplotlib.pyplot as plt
%matplotlib inline

# Preprocessing
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler, scale

# Models
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.model_selection import GridSearchCV

# Metrics
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,␣
 ↪recall_score, f1_score
```

1.2 Target Data

1.2.1 Loading dataset

[3]:
```
cancer_targets = pd.read_csv("./data/Copy of actual.csv")
```

[4]:
```
cancer_targets.head()
```

```
[4]:    patient cancer
     0        1    ALL
     1        2    ALL
     2        3    ALL
     3        4    ALL
     4        5    ALL
```

```
[5]: cancer_targets.dtypes
```

```
[5]: patient     int64
     cancer     object
     dtype: object
```

1.2.2 Class distribution

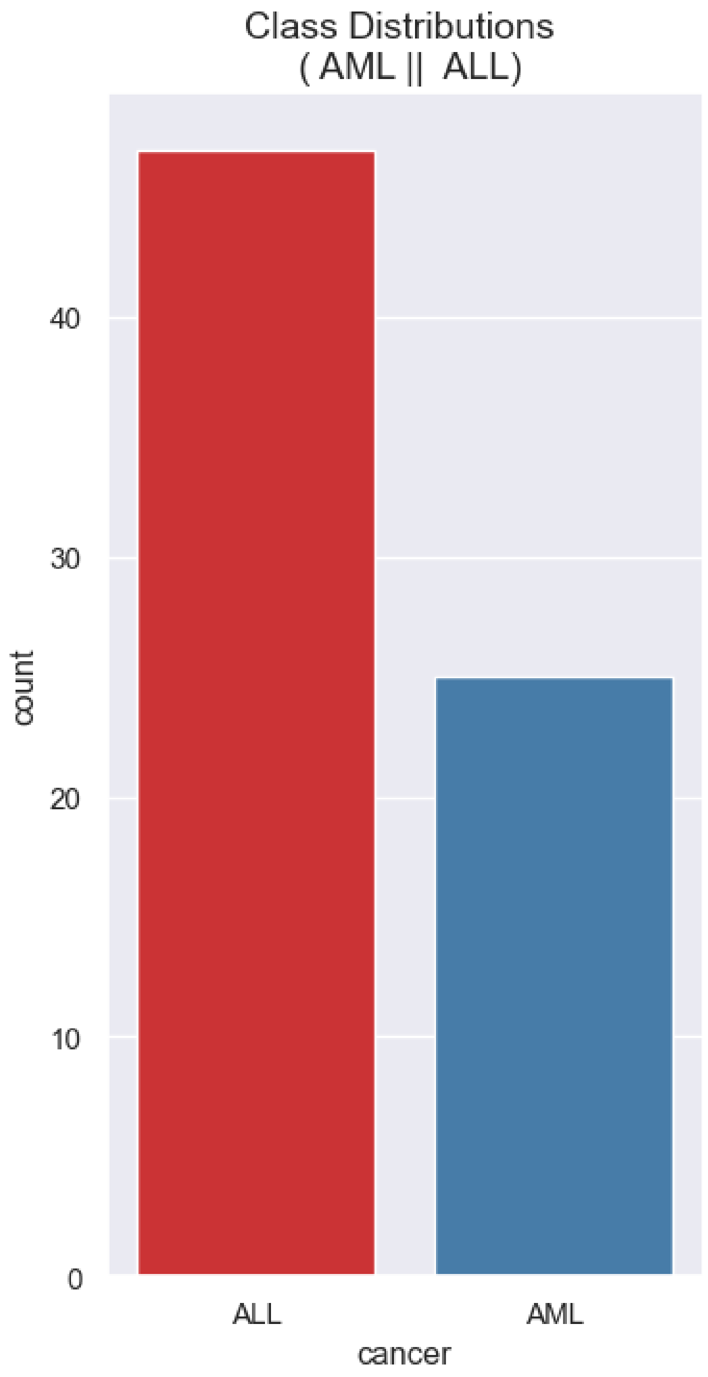
```
[6]: print(cancer_targets["cancer"].value_counts())
     print()
     print("Number of samples;", cancer_targets.shape)
```

```
cancer
ALL    47
AML    25
Name: count, dtype: int64

Number of samples; (72, 2)
```

```
[7]: plt.figure(figsize=(4, 8))
     colors = ["AML", "ALL"]
     sns.countplot(data=cancer_targets, x='cancer', palette="Set1")
     plt.title('Class Distributions \n ( AML ||  ALL)', fontsize=14)
```

```
[7]: Text(0.5, 1.0, 'Class Distributions \n ( AML ||  ALL)')
```


Class Distributions
(AML || ALL)
count
0
10
20
30
40
ALL
AML
cancer

1.3 Train and test dataset

[8]:
```python
cancer_train = pd.read_csv("./data/data_train.csv")
cancer_test = pd.read_csv("./data/data_test.csv")

print("Train shape:", cancer_train.shape)
print("Test shape:", cancer_test.shape)

cancer_train.head(4)
```

```
Train shape: (7129, 78)
Test shape: (7129, 70)
```

[8]:
```
                        Gene Description Gene Accession Number    1 call    2  \
0  AFFX-BioB-5_at (endogenous control)        AFFX-BioB-5_at -214    A -139
1  AFFX-BioB-M_at (endogenous control)        AFFX-BioB-M_at -153    A  -73
2  AFFX-BioB-3_at (endogenous control)        AFFX-BioB-3_at  -58    A   -1
3  AFFX-BioC-5_at (endogenous control)        AFFX-BioC-5_at   88    A  283

  call.1    3 call.2    4 call.3  …   29 call.33   30 call.34   31 call.35  \
0      A  -76      A -135      A  …   15       A -318       A  -32       A
1      A  -49      A -114      A  … -114       A -192       A  -49       A
2      A -307      A  265      A  …    2       A  -95       A   49       A
3      A  309      A   12      A  …  193       A  312       A  230       P

    32 call.36   33 call.37
0 -124       A -135       A
1  -79       A -186       A
2  -37       A  -70       A
3  330       A  337       A

[4 rows x 78 columns]
```

1.3.1 Preprocessing

We will be combining the train and test dataset for preprocessing and then we will split them again.

[9]:
```python
def rename_columns(df):
    for col in df.columns:
        if "call" in col:
            loc = df.columns.get_loc(col)
            patient = df.columns[loc-1]
            df.rename(columns={col: f'Call_{patient}'}, inplace=True)
```

[10]:
```python
rename_columns(df=cancer_train)
rename_columns(df=cancer_test)

cancer_train["Gene"] = cancer_train["Gene Description"] + \
    '_' + cancer_train["Gene Accession Number"]
cancer_test["Gene"] = cancer_test["Gene Description"] + \
    '_' + cancer_test["Gene Accession Number"]

cancer_train = cancer_train.T
cancer_train.columns = cancer_train.iloc[-1]
cancer_train = cancer_train[2:-1]
cancer_train['dataset'] = 'train'

cancer_test = cancer_test.T
cancer_test.columns = cancer_test.iloc[-1]
cancer_test = cancer_test[2:-1]
cancer_test['dataset'] = 'test'


df = pd.concat([cancer_train, cancer_test], axis=0, join='inner', sort=False)
df.shape
```

[10]:
```
(144, 7130)
```

> The columns labeled "Present," "Absent," and "Marginal" represent the detection calls made by the DNA Microarray manufacturer in the paper. These designations are based on comparing p-values of intensity calls to a predefined noise frequency cutoff. Consequently, it is advisable to remove rows where all values are designated as "Absent" (A) calls, as these are deemed unreliable.

Reference: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1409797/

1.3.2 Dropping the columns with all A in calls

[11]:
```
call_rows = [row for row in df.index if "Call" in row]
conditional = df.filter(call_rows, axis=0).apply(
    lambda x: x == 'A', axis=1).all()
print(conditional.value_counts())
df = df.loc[:, ~conditional]
```

```
False    5328
True     1802
Name: count, dtype: int64
```

1.3.3 Removing the call columns

[12]:
```
df.drop(call_rows, axis=0, inplace=True)
df['patient'] = df.index
df['patient'] = df['patient'].astype('int')
df.reset_index(drop=True)
```

```
[12]: Gene AFFX-BioC-5_at (endogenous control)_AFFX-BioC-5_at  \
      0                                                       88
      1                                                      283
      2                                                      309
      3                                                       12
      4                                                      168
      ..                                                     …
      67                                                     141
      68                                                      95
      69                                                     146
      70                                                     431
      71                                                       9

      Gene hum_alu_at (miscellaneous control)_hum_alu_at  \
      0                                               15091
      1                                               11038
      2                                               16692
      3                                               15763
      4                                               18128
      ..                                                …
      67                                              22818
      68                                              39323
      69                                              15689
      70                                              41570
      71                                              39538

      Gene AFFX-DapX-5_at (endogenous control)_AFFX-DapX-5_at  \
      0                                                        7
      1                                                       37
      2                                                      183
      3                                                       45
      4                                                      -28
      ..                                                     …
      67                                                       2
      68                                                     -26
      69                                                       6
      70                                                      94
      71                                                    -104

      Gene AFFX-DapX-M_at (endogenous control)_AFFX-DapX-M_at  \
      0                                                      311
      1                                                      134
      2                                                      378
      3                                                      268
      4                                                      118
      ..                                                     …
      67                                                      46
```

```
68                                                   73
69                                                  302
70                                                  235
71                                                  101

Gene AFFX-LysX-5_at (endogenous control)_AFFX-LysX-5_at  \
0                                                    21
1                                                   -21
2                                                    67
3                                                    43
4                                                    -8
..                                                  …
67                                                   26
68                                                   39
69                                                   25
70                                                   27
71                                                  106

Gene AFFX-HUMISGF3A/M97935_MA_at (endogenous control)_AFFX-
HUMISGF3A/M97935_MA_at  \
0                                                   -13
1                                                  -219
2                                                   104
3                                                  -148
4                                                   -55
..                                                  …
67                                                 -203
68                                                  -60
69                                                 -209
70                                                 -626
71                                                 -240

Gene AFFX-HUMISGF3A/M97935_MB_at (endogenous control)_AFFX-
HUMISGF3A/M97935_MB_at  \
0                                                   215
1                                                   116
2                                                   476
3                                                   155
4                                                   122
..                                                  …
67                                                   25
68                                                   60
69                                                  183
70                                                 -249
71                                                  113

Gene AFFX-HUMISGF3A/M97935_3_at (endogenous control)_AFFX-HUMISGF3A/M97935_3_at
```

```
\
0                                                   797
1                                                   433
2                                                  1474
3                                                   415
4                                                   483
..                                                  …
67                                                  264
68                                                  306
69                                                  657
70                                                  477
71                                                 1313

Gene AFFX-HUMRGE/M10098_5_at (endogenous control)_AFFX-HUMRGE/M10098_5_at  \
0                                                 14538
1                                                   615
2                                                  5669
3                                                  4850
4                                                  1284
..                                                  …
67                                                  104
68                                                  569
69                                                 3762
70                                                 -159
71                                                   34

Gene AFFX-HUMRGE/M10098_M_at (endogenous control)_AFFX-HUMRGE/M10098_M_at  \
0                                                  9738
1                                                   115
2                                                  3272
3                                                  2293
4                                                  2731
..                                                  …
67                                                 -159
68                                                  478
69                                                 2164
70                                                 -745
71                                                  -62

Gene  … Transcription factor Stat5b (stat5b) mRNA_U48730_at  \
0     …                                                 185
1     …                                                 169
2     …                                                 315
3     …                                                 240
4     …                                                 156
..    …                                                 …
67    …                                                  92
```

```
68    …                                                     63
69    …                                                    130
70    …                                                     84
71    …                                                     81

Gene Breast epithelial antigen BA46 mRNA_U58516_at  \
0                                                 511
1                                                 837
2                                                1199
3                                                 835
4                                                 649
..                                                …
67                                                532
68                                                297
69                                                639
70                                               1141
71                                                574

Gene TUBULIN ALPHA-4 CHAIN_X06956_at  \
0                                   389
1                                   442
2                                   168
3                                   174
4                                   504
..                                  …
67                                  239
68                                  358
69                                  548
70                                  197
71                                  618

Gene PTGER3 Prostaglandin E receptor 3 (subtype EP3) {alternative
products}_X83863_at  \
0                                                      793
1                                                      782
2                                                     1138
3                                                      627
4                                                      250
..                                                     …
67                                                     707
68                                                     423
69                                                     809
70                                                     466
71                                                     551

Gene HMG2 High-mobility group (nonhistone chromosomal) protein 2_Z17240_at  \
0                                                      329
```

```
1                                                    295
2                                                    777
3                                                    170
4                                                    314
..                                                   …
67                                                   354
68                                                    41
69                                                   445
70                                                   349
71                                                   194

Gene RB1 Retinoblastoma 1 (including osteosarcoma)_L49218_f_at  \
0                                                     36
1                                                     11
2                                                     41
3                                                    -50
4                                                     14
..                                                   …
67                                                   -22
68                                                     0
69                                                    -2
70                                                     0
71                                                    20

Gene GB DEF = Glycophorin Sta (type A) exons 3 and 4; partial_M71243_f_at  \
0                                                    191
1                                                     76
2                                                    228
3                                                    126
4                                                     56
..                                                   …
67                                                   260
68                                                  1777
69                                                   210
70                                                   284
71                                                   379

Gene GB DEF = mRNA (clone 1A7)_Z78285_f_at dataset patient
0                                      -37   train       1
1                                      -14   train       2
2                                      -41   train       3
3                                      -91   train       4
4                                      -25   train       5
..                                     …     …     …
67                                       5    test      65
68                                     -49    test      66
69                                      16    test      63
```

```
70                                           -73    test      64
71                                           -60    test      62

[72 rows x 5329 columns]
```

1.3.4 Combining features and labels

[13]:
```
df = pd.merge(left=df, right=cancer_targets,
              left_on='patient', right_on='patient')
print(df.shape)
df.head(5)
```

```
(72, 5330)
```

[13]:
```
  AFFX-BioC-5_at (endogenous control)_AFFX-BioC-5_at  \
0                                                 88
1                                                283
2                                                309
3                                                 12
4                                                168

  hum_alu_at (miscellaneous control)_hum_alu_at  \
0                                         15091
1                                         11038
2                                         16692
3                                         15763
4                                         18128

  AFFX-DapX-5_at (endogenous control)_AFFX-DapX-5_at  \
0                                                  7
1                                                 37
2                                                183
3                                                 45
4                                                -28

  AFFX-DapX-M_at (endogenous control)_AFFX-DapX-M_at  \
0                                                311
1                                                134
2                                                378
3                                                268
4                                                118

  AFFX-LysX-5_at (endogenous control)_AFFX-LysX-5_at  \
0                                                 21
1                                                -21
2                                                 67
3                                                 43
```

```
4                                              -8

  AFFX-HUMISGF3A/M97935_MA_at (endogenous control)_AFFX-HUMISGF3A/M97935_MA_at
\
0                                             -13
1                                            -219
2                                             104
3                                            -148
4                                             -55

  AFFX-HUMISGF3A/M97935_MB_at (endogenous control)_AFFX-HUMISGF3A/M97935_MB_at
\
0                                             215
1                                             116
2                                             476
3                                             155
4                                             122

  AFFX-HUMISGF3A/M97935_3_at (endogenous control)_AFFX-HUMISGF3A/M97935_3_at  \
0                                             797
1                                             433
2                                            1474
3                                             415
4                                             483

  AFFX-HUMRGE/M10098_5_at (endogenous control)_AFFX-HUMRGE/M10098_5_at  \
0                                           14538
1                                             615
2                                            5669
3                                            4850
4                                            1284

  AFFX-HUMRGE/M10098_M_at (endogenous control)_AFFX-HUMRGE/M10098_M_at  …  \
0                                            9738                        …
1                                             115                        …
2                                            3272                        …
3                                            2293                        …
4                                            2731                        …

  Breast epithelial antigen BA46 mRNA_U58516_at  \
0                                          511
1                                          837
2                                         1199
3                                          835
4                                          649

  TUBULIN ALPHA-4 CHAIN_X06956_at  \
0                             389
1                             442
2                             168
3                             174
4                             504

  PTGER3 Prostaglandin E receptor 3 (subtype EP3) {alternative
products}_X83863_at  \
0                                                     793
1                                                     782
2                                                    1138
3                                                     627
4                                                     250

  HMG2 High-mobility group (nonhistone chromosomal) protein 2_Z17240_at  \
0                                                     329
1                                                     295
2                                                     777
3                                                     170
4                                                     314

  RB1 Retinoblastoma 1 (including osteosarcoma)_L49218_f_at  \
0                                                      36
1                                                      11
2                                                      41
3                                                     -50
4                                                      14

  GB DEF = Glycophorin Sta (type A) exons 3 and 4; partial_M71243_f_at  \
0                                                     191
1                                                      76
2                                                     228
3                                                     126
4                                                      56

  GB DEF = mRNA (clone 1A7)_Z78285_f_at dataset patient cancer
0                                   -37   train       1    ALL
1                                   -14   train       2    ALL
2                                   -41   train       3    ALL
3                                   -91   train       4    ALL
4                                   -25   train       5    ALL

[5 rows x 5330 columns]
```

1.3.5 Separate train and test data

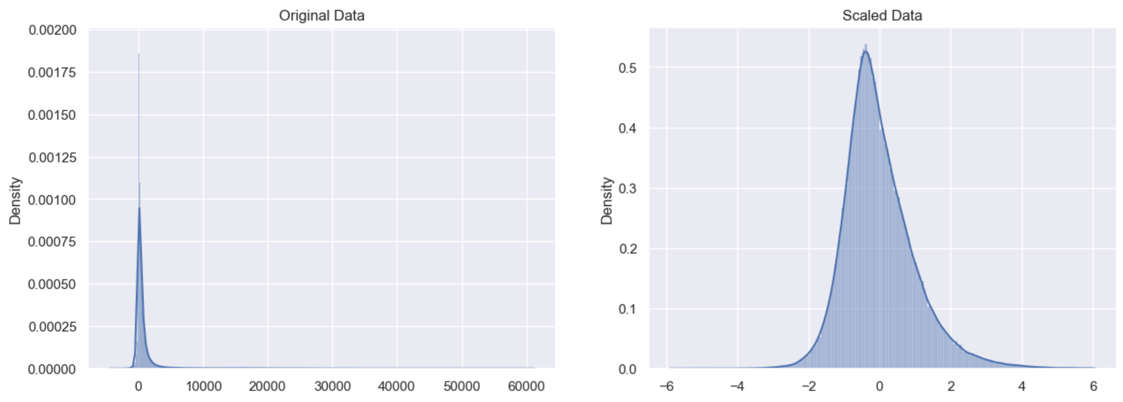
[14]:
```python
train = df[df['dataset'] == 'train'].iloc[:, 0:-3]
train_target = df[df['dataset'] == 'train'].iloc[:, -1]
test = df[df['dataset'] == 'test'].iloc[:, 0:-3]
test_target = df[df['dataset'] == 'test'].iloc[:, -1]

print(train.shape, train_target.shape)
print(test.shape, test_target.shape)
```

```
(38, 5327) (38,)
(34, 5327) (34,)
```

[15]:
```python
scaler = StandardScaler().fit(train)
train_scaled = pd.DataFrame(scaler.transform(train), columns=train.columns)
test_scaled = pd.DataFrame(scaler.transform(test), columns=test.columns)

fig, ax = plt.subplots(ncols=2, figsize=(15, 5))
sns.histplot(np.concatenate(train.values), ax=ax[0], kde=True, stat="density")
sns.histplot(np.concatenate(train_scaled.values),
             ax=ax[1], kde=True, stat="density")
ax[0].set_title('Original Data')
ax[1].set_title('Scaled Data')
plt.tight_layout
plt.show()
```



1.3.6 Feature Selection

[16]:
```python
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SelectFromModel

print("Original Shape:", train_scaled.shape)
```

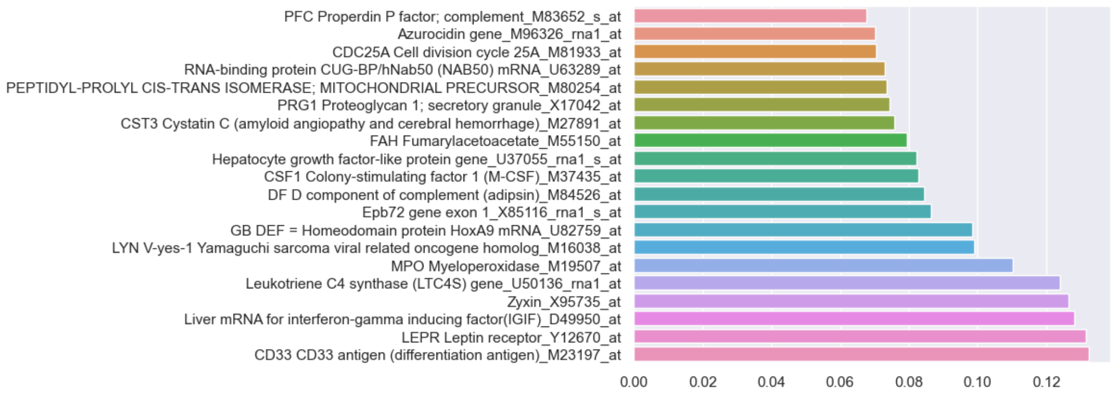
```
logistic_regression = LogisticRegression(penalty="l1", solver='saga',␣
↪max_iter=2000).fit(
    train_scaled, train_target)  # l1 for sparsity
log_coefficients = logistic_regression.coef_
selector_log = SelectFromModel(logistic_regression, prefit=True)
train_scaled_logreg = selector_log.transform(train_scaled)
test_scaled_logreg = selector_log.transform(test_scaled)
print("Features after selection using Logistic Regression:",
      train_scaled_logreg.shape)


log_coefficients_abs = abs(log_coefficients)
log_coefficients_abs_sort = np.sort(log_coefficients_abs).flatten()
sortedidx = log_coefficients_abs.argsort()
log_labels = train_scaled.columns.values[sortedidx].flatten()
sns.barplot(x=log_coefficients_abs_sort[-20:], y=log_labels[-20:])
```

```
Original Shape: (38, 5327)
Features after selection using Logistic Regression: (38, 146)
```

```
/home/karthi/anaconda3/envs/ml/lib/python3.11/site-
packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was
reached which means the coef_ did not converge
  warnings.warn(
/home/karthi/anaconda3/envs/ml/lib/python3.11/site-packages/sklearn/base.py:457:
UserWarning: X has feature names, but SelectFromModel was fitted without feature
names
  warnings.warn(
/home/karthi/anaconda3/envs/ml/lib/python3.11/site-packages/sklearn/base.py:457:
UserWarning: X has feature names, but SelectFromModel was fitted without feature
names
  warnings.warn(
```

```
[16]: <Axes: >
```



1.3.7 Feature Scaling using PCA

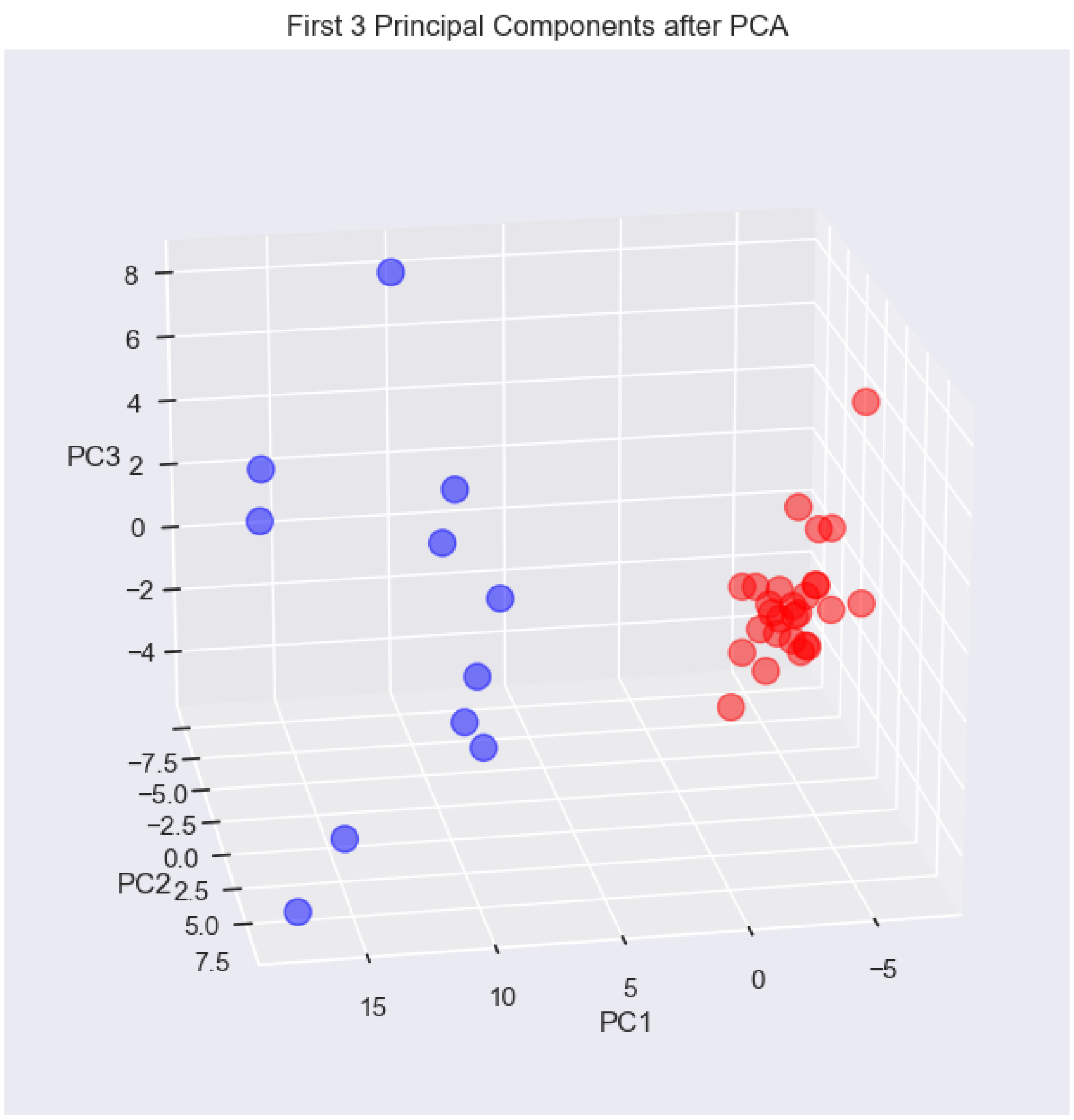
[17]:
```
pca = PCA(n_components=3)
pca.fit_transform(train_scaled_logreg)
print(pca.explained_variance_ratio_)

PCA_df = pd.DataFrame(data=pca.fit_transform(train_scaled_logreg),
                      columns=['pc1', 'pc2', 'pc3'])
```

```
[0.39758334 0.08187505 0.0603069 ]
```

[18]:
```
PCA_df = pd.concat([PCA_df, train_target], axis=1)

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
colors = {'ALL': 'red', 'AML': 'blue'}
ax.scatter(PCA_df.pc1, PCA_df.pc2, PCA_df.pc3,
           c=train_target.apply(lambda x: colors[x]),
           s=120, alpha=0.5
           )
plt.title('First 3 Principal Components after PCA')
ax.set_xlabel('PC1')
ax.set_ylabel('PC2')
ax.set_zlabel('PC3')
ax.view_init(20, 80)
plt.tight_layout
plt.show()
```



2 Models being tested

1. C-Support Vector Classification (SVM)

- Using Grid search for tuning hyperparameters

2. Random forest Classifier

- Using Grid search for tuning hyperparameters

3. Neural Networks

- Using only Grid search

2.1 SVM

2.1.1 Grid search for SVM

```
[19]: svc = SVC()
```

2.1.2 Grid search parameters

```
[20]: param_grid = {'C': [0.1, 1, 10, 100],
                    'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
                    'kernel': ['rbf', 'poly', 'linear', 'sigmoid'], }

      grid = GridSearchCV(svc, param_grid, cv=5)
```

2.1.3 Best parameters

```
[21]: grid.fit(train_scaled_logreg, train_target)
      print(f"Best parameters for SVM are: {grid.best_params_}")
```

```
Best parameters for SVM are: {'C': 0.1, 'gamma': 1, 'kernel': 'linear'}
```

2.1.4 Selecting the best svm model

```
[22]: svc = grid.best_estimator_
```

2.1.5 Fitting the model

```
[23]: svc.fit(train_scaled_logreg, train_target)
```

```
[23]: SVC(C=0.1, gamma=1, kernel='linear')
```

2.1.6 Evaluation of the model

```
[24]: # predicting the values
      test_pred = svc.predict(test_scaled_logreg)

      # calculating the accuracy, precision, recall, f1-score, roc_auc_score
      print(
          f"Accuracy of the best SVM model is: {accuracy_score(test_target,␣
        ↪test_pred)}")
      print(
          f"Precision of the best SVM model is: {precision_score(test_target,␣
        ↪test_pred, pos_label='AML')}")
      print(
          f"Recall of the best SVM model is: {recall_score(test_target, test_pred,␣
        ↪pos_label='AML')}")
      print(
    f"F1-score of the best SVM model is: {f1_score(test_target, test_pred,␣
↪pos_label='AML')}")
```

```
Accuracy of the best SVM model is: 0.9705882352941176
Precision of the best SVM model is: 0.9333333333333333
Recall of the best SVM model is: 1.0
F1-score of the best SVM model is: 0.9655172413793104
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/mnt/garuda/karthikeya/college/mlbio/Assignment2/Assignment2.ipynb Cell 46 line 1
      <a href='vscode-notebook-cell:/mnt/garuda/karthikeya/college/mlbio/
 ↪Assignment2/Assignment2.ipynb#Y600sZmlsZQ%3D%3D?line=8'>9</a> print(
     <a href='vscode-notebook-cell:/mnt/garuda/karthikeya/college/mlbio/
 ↪Assignment2/Assignment2.ipynb#Y600sZmlsZQ%3D%3D?line=9'>10</a>     f"Recall o␣
 ↪the best SVM model is: {recall_score(test_target, test_pred,␣
 ↪pos_label='AML')}")
     <a href='vscode-notebook-cell:/mnt/garuda/karthikeya/college/mlbio/
 ↪Assignment2/Assignment2.ipynb#Y600sZmlsZQ%3D%3D?line=10'>11</a> print(
     <a href='vscode-notebook-cell:/mnt/garuda/karthikeya/college/mlbio/
 ↪Assignment2/Assignment2.ipynb#Y600sZmlsZQ%3D%3D?line=11'>12</a>     f"F1-scor␣
 ↪of the best SVM model is: {f1_score(test_target, test_pred, pos_label='AML')} )
     <a href='vscode-notebook-cell:/mnt/garuda/karthikeya/college/mlbio/
 ↪Assignment2/Assignment2.ipynb#Y600sZmlsZQ%3D%3D?line=12'>13</a> print(
---> <a href='vscode-notebook-cell:/mnt/garuda/karthikeya/college/mlbio/
 ↪Assignment2/Assignment2.ipynb#Y600sZmlsZQ%3D%3D?line=13'>14</a>     f"ROC-AUC␣
 ↪score of the best SVM model is: {roc_auc_score(test_target, test_pred,␣
 ↪pos_label='AML')}")

File ~/anaconda3/envs/ml/lib/python3.11/site-packages/sklearn/utils/
 ↪_param_validation.py:189, in validate_params.<locals>.decorator.<locals>.
 ↪wrapper(*args, **kwargs)
    186 func_sig = signature(func)
    188 # Map *args/**kwargs to the function signature
--> 189 params = func_sig.bind(*args, **kwargs)
    190 params.apply_defaults()
    192 # ignore self/cls and positional/keyword markers

File ~/anaconda3/envs/ml/lib/python3.11/inspect.py:3212, in Signature.bind(self ␣
 ↪*args, **kwargs)
   3207 def bind(self, /, *args, **kwargs):
   3208     """Get a BoundArguments object, that maps the passed `args`
   3209     and `kwargs` to the function's signature.  Raises `TypeError`
   3210     if the passed arguments can not be bound.
   3211     """
-> 3212     return self._bind(args, kwargs)

File ~/anaconda3/envs/ml/lib/python3.11/inspect.py:3201, in Signature.
 ↪_bind(self, args, kwargs, partial)
```

```
   3199         arguments[kwargs_param.name] = kwargs
   3200     else:
-> 3201         raise TypeError(
   3202             'got an unexpected keyword argument {arg!r}'.format(
   3203                 arg=next(iter(kwargs))))
   3205 return self._bound_arguments_cls(self, arguments)

TypeError: got an unexpected keyword argument 'pos_label'
```

2.1.7 Plotting the confusion matrix

[ ]:
```
# confusion matrix
cm = confusion_matrix(test_target, test_pred)

plt.figure(figsize=(5, 3))
sns.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
plt.title('Confusion Matrix')
plt.show()
```

2.2 Random Forest

[ ]:
```
# using Random Forest and performing grid search to find the best parameters
rfc = RandomForestClassifier(random_state=0)

# defining parameter range
param_grid = {'n_estimators': [60, 70, 80, 90, 100],
              'max_features': [0.6, 0.7, 0.8, 0.9],
              'min_samples_leaf': [8, 10, 12, 14],
              'min_samples_split': [3, 5, 7]}


grid = GridSearchCV(rfc, param_grid, cv=3, scoring='accuracy')
```

2.2.1 Best parameters

[ ]:
```
grid.fit(train_scaled_logreg, train_target)
print(f"Best parameters for Random Forest are: {grid.best_params_}")
```

2.2.2 Selecting the best random forest model

[ ]:
```
rfc = grid.best_estimator_
```

2.2.3 Fitting the model

```
[ ]: rfc.fit(train_scaled_logreg, train_target)
```

2.2.4 Evaluation of the model

```
[ ]: # predicting the values
     test_pred = rfc.predict(test_scaled_logreg)

     # calculating the accuracy, precision, recall, f1-score, roc_auc_score
     print(
         f"Accuracy of the best Random Forest model is: {accuracy_score(test_target,␣
      ↪test_pred)}")
     print(
         f"Precision of the best Random Forest model is:␣
      ↪{precision_score(test_target, test_pred, average='macro')}")
     print(
         f"Recall of the best Random Forest model is: {recall_score(test_target,␣
      ↪test_pred, average='macro')}")
     print(
         f"F1 Score of the best Random Forest model is: {f1_score(test_target,␣
      ↪test_pred, average='macro')}")
```

2.2.5 Plotting the confusion matrix

```
[ ]: # confusion matrix
     cm = confusion_matrix(test_target, test_pred)

     plt.figure(figsize=(5, 3))
     sns.heatmap(cm, annot=True)
     plt.xlabel('Predicted')
     plt.ylabel('Truth')
     plt.title('Confusion Matrix')
     plt.show()
```

2.3 Neural Network

2.3.1 Grid search for MLP

```
[ ]: mlp = MLPClassifier(max_iter=1000)
```

2.3.2 Grid Parameters

```
[ ]: param_grid = {'hidden_layer_sizes': [(50, 50, 50), (50, 100, 50), (100,)],
                   'activation': ['tanh', 'relu', 'logistic'],
                   'solver': ['sgd', 'adam'],
                   'alpha': [0.0001, 0.05],
                   'learning_rate': ['constant', 'adaptive']}


     grid = GridSearchCV(mlp, param_grid, cv=3, scoring='accuracy')
```

2.3.3 Best parameters

```
[ ]: grid.fit(train_scaled_logreg, train_target)
     print(f"Best parameters for Neural Network are: {grid.best_params_}")
```

2.3.4 Selecting the best neural network model

```
[ ]: mlp = grid.best_estimator_
```

2.3.5 Fitting the model

```
[ ]: mlp.fit(train_scaled_logreg, train_target)
```

2.3.6 Evaluating the model

```
[ ]: # predicting the values
     test_pred = mlp.predict(test_scaled_logreg)

     # calculating the accuracy, precision, recall, f1-score, roc_auc_score
     print(
         f"Accuracy of the best Neural Network model is:␣
       ↪{accuracy_score(test_target, test_pred)}")
     print(
         f"Precision of the best Neural Network model is:␣
       ↪{precision_score(test_target, test_pred, average='macro')}")
     print(
         f"Recall of the best Neural Network model is: {recall_score(test_target,␣
       ↪test_pred, average='macro')}")
     print(
         f"F1 Score of the best Neural Network model is: {f1_score(test_target,␣
       ↪test_pred, average='macro')}")
```

2.3.7 Plotting the confusion matrix

```
[ ]: # confusion matrix
     cm = confusion_matrix(test_target, test_pred)

     plt.figure(figsize=(5, 3))
     sns.heatmap(cm, annot=True)
     plt.xlabel('Predicted')
```

```python
plt.ylabel('Truth')
plt.title('Confusion Matrix')
plt.show()
```