In [32]:
```
#Rina Patel Surhil Gohel Machine Learning Techniques Final Project April 19,2022
```

In [33]:
```
#1. Load data in Python (pd.read_csv) command
```

In [34]:
```
import pandas as pd
```

In [35]:
```
data= r'P1_HCV_EG_DATA.csv'
df = pd.read_csv(data)
df
print(df.columns)
```

```
Index(['Age ', 'Gender', 'BMI', 'Fever', 'Nausea/Vomting', 'Headache ',
       'Diarrhea ', 'Fatigue & generalized bone ache ', 'Jaundice ',
       'Epigastric pain ', 'WBC', 'RBC', 'HGB', 'Plat', 'AST 1', 'ALT 1',
       'ALT4', 'ALT 12', 'ALT 24', 'ALT 36', 'ALT 48', 'ALT after 24 w',
       'RNA Base', 'RNA 4', 'RNA 12', 'RNA EOT', 'RNA EF',
       'Baseline histological Grading', 'Baselinehistological staging'],
      dtype='object')
```

In [36]:
```
#Before proceeding to our next step, we should preprocess the data so that we have a be
#in our machine learning algorithm. In this particular project, I am assigned to only f
#outputs of 3 and 4, so I need to reduce my dataset to only have these two outcomes.
```

In [37]:
```
# remove special character
df.columns = df.columns.str.replace(' ', '')

# print file after removing special character
print("\n\n", df)
#This action allowed me to remove the spaces from the column heading so I can more easi

df.Baselinehistologicalstaging
```

```
      Age  Gender  BMI  Fever  Nausea/Vomting  Headache  Diarrhea  \
0      56       1   35      2               1         1         1
1      46       1   29      1               2         2         1
2      57       1   33      2               2         2         2
3      49       2   33      1               2         1         2
4      59       1   32      1               1         2         1
...   ...     ...  ...    ...             ...       ...       ...
1380   44       1   29      1               2         2         2
1381   55       1   34      1               2         2         1
1382   42       1   26      2               2         1         1
1383   52       1   29      2               1         1         2
1384   55       2   26      1               2         2         2

      Fatigue&generalizedboneache  Jaundice  Epigastricpain  ...  ALT36  \
0                               2         2               2  ...      5
1                               2         2               1  ...     57
2                               1         1               1  ...      5
3                               1         2               1  ...     48
4                               2         2               2  ...     94
```

```
...                          ...       ...              ...  ...    ...
1380                           1         1                1  ...     63
1381                           1         1                1  ...     97
1382                           1         2                1  ...     87
1383                           2         2                1  ...     48
1384                           1         2                1  ...     64

      ALT48  ALTafter24w  RNABase    RNA4    RNA12  RNAEOT   RNAEF  \
0         5            5   655330  634536   288194       5       5
1       123           44    40620  538635   637056  336804   31085
2         5            5   571148  661346        5  735945  558829
3        77           33  1041941  449939   585688  744463  582301
4        90           30   660410  738756  3731527  338946  242861
...     ...          ...      ...     ...      ...     ...     ...
1380     44           45   387795   55938        5       5       5
1381     64           41   481378  152961   393339   73574  236273
1382     39           24   612664  572756   806109  343719  160457
1383     81           43   139872   76161   515730    2460  696074
1384     71           34  1190577  628730        5       5       5

      BaselinehistologicalGrading  Baselinehistologicalstaging
0                              13                            2
1                               4                            2
2                               4                            4
3                              10                            3
4                              11                            1
...                           ...                          ...
1380                           15                            4
1381                           10                            2
1382                            6                            2
1383                           15                            3
1384                           13                            3

[1385 rows x 29 columns]
```

Out[37]:

```
0       2
1       2
2       4
3       3
4       1
       ..
1380    4
1381    2
1382    2
1383    3
1384    3
Name: Baselinehistologicalstaging, Length: 1385, dtype: int64
```

In [38]:

```
df1=df[(df.Baselinehistologicalstaging==3)|(df.Baselinehistologicalstaging==4)]
df1
#This allows us to have a dataset with only 3 or 4 for our outcome (Baseline histologic
```

Out[38]:

	Age	Gender	BMI	Fever	Nausea/Vomting	Headache	Diarrhea	Fatigue&generalizedboneache	J
2	57	1	33	2	2	2	2	1	
3	49	2	33	1	2	1	2	1	
5	58	2	22	2	2	2	1	2	
6	42	2	26	1	1	2	2	2	

	Age	Gender	BMI	Fever	Nausea/Vomting	Headache	Diarrhea	Fatigue&generalizedboneache	J
7	48	2	30	1	1	2	2	1	
...	...	...	...	...	...	...	...	...	
1378	33	1	24	1	1	1	2	2	
1379	53	1	31	2	2	1	2	2	
1380	44	1	29	1	2	2	2	1	
1383	52	1	29	2	1	1	2	2	
1384	55	2	26	1	2	2	2	1	

717 rows × 29 columns

In [39]:
```
#Data preprocessing must be performed before continuing the project
from sklearn import preprocessing
#I am getting rid of some features that are not applicable for the case of this project
#age, gender, and BMI. I am also removing WBC, RBC, HGB, and platelet from the dataset,
#ALT levels, and RNA sequences for the project
df_new=df1.drop(columns=['Age','Gender','BMI','WBC','RBC','HGB','Plat'])
df_new
```

Out[39]:

	Fever	Nausea/Vomting	Headache	Diarrhea	Fatigue&generalizedboneache	Jaundice	Epigastricpa
2	2	2	2	2	1	1	
3	1	2	1	2	1	2	
5	2	2	2	1	2	2	
6	1	1	2	2	2	2	
7	1	1	2	2	1	1	
...	...	...	...	...	...	...	
1378	1	1	1	2	2	2	
1379	2	2	1	2	2	2	
1380	1	2	2	2	1	1	
1383	2	1	1	2	2	2	
1384	1	2	2	2	1	2	

717 rows × 22 columns

In [40]:
```
#Split my dataframe into features and outcome (target)
df_features=df_new.iloc[:,:21]
df_features
df_outcome=df_new.Baselinehistologicalstaging
df_outcome
```

Out[40]:
```
2       4
3       3
5       4
6       4
7       3
       ..
1378    4
1379    4
1380    4
1383    3
1384    3
Name: Baselinehistologicalstaging, Length: 717, dtype: int64
```

In [41]:
```
#2. Split the data in to training (80%) and testing data (20%).
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import svm, datasets
from sklearn.metrics import plot_roc_curve,auc
#Split my dataset into training and testing samples
data_train,data_test,target_train,target_test=train_test_split(df_features,
                                                               df_outcome,
                                                               test_size=0.20,
                                                               random_state=42)
```

In [42]:
```
#3 Train at least 3 different SVM/Neural network models by using different values of ke
#other parameters.

#SVM Classification with Linear Kernel
svc = svm.SVC(kernel='linear', C=1,gamma= 1/2)
svc.fit(data_train, target_train)
# perform prediction and get accuracy score
svc.score(data_test,target_test)
```

Out[42]:
```
0.4652777777777778
```

In [43]:
```
## here we train SVM with RBF kernel
svc_rbf = svm.SVC(kernel='rbf', C=1,gamma= 1/2)
svc_rbf.fit(data_train, target_train)

# perform prediction and get accuracy score
svc_rbf.score(data_test,target_test)
```

Out[43]:
```
0.4791666666666667
```

In [44]:
```
from sklearn.neural_network import MLPClassifier
mlp=MLPClassifier(max_iter=1000)
mlp.fit(data_train, target_train)
score = mlp.score(data_test, target_test)
print(score)
```

```
0.5069444444444444
```

In [45]:
```
#From our first split, it looks like nerual networks worked the past with classificatio
#performed particularly well. SVM, using RBF kernel, had a score of correctly identifyi
#score of 47%. Neural network was able to perform classification with a score of %50.
```

In [46]:
```
#Calculate and compare the accuracy of the models using the test data.
```

In [47]:
```
#5.Starting from the data in step 1, create a new training and testing data split. This
#should be different from the original data split performed in step 2. After creating t
#and testing data split, repeat steps 3-4 on this new data.
#Split my dataset into training and testing samples
data_train1,data_test1,target_train1,target_test1=train_test_split(df_features,
                                                    df_outcome,
                                                    test_size=0.20,
                                                    random_state=67)
```

In [48]:
```
#SVM Classification with Linear Kernel
svc = svm.SVC(kernel='linear', C=1,gamma= 1/2)
svc.fit(data_train1, target_train1)
# perform prediction and get accuracy score
svc.score(data_test1,target_test1)
```

Out[48]:
```
0.4305555555555556
```

In [49]:
```
## here we train SVM with RBF kernel
svc_rbf = svm.SVC(kernel='rbf', C=1,gamma= 1/2)
svc_rbf.fit(data_train1, target_train1)

# perform prediction and get accuracy score
svc_rbf.score(data_test1,target_test1)
```

Out[49]:
```
0.4652777777777778
```

In [50]:
```
mlp=MLPClassifier(max_iter=1000)
mlp.fit(data_train1, target_train1)
score = mlp.score(data_test1, target_test1)
print(score)
```

```
0.4583333333333333
```

In [ ]:
```
#From our second split, it looks like SVM worked the past with classification of the tw
#particularly well. SVM, using RBF kernel, had a score of correctly identifying the out
#was 43%, and neural network was able to predict the outcome at 46%
```