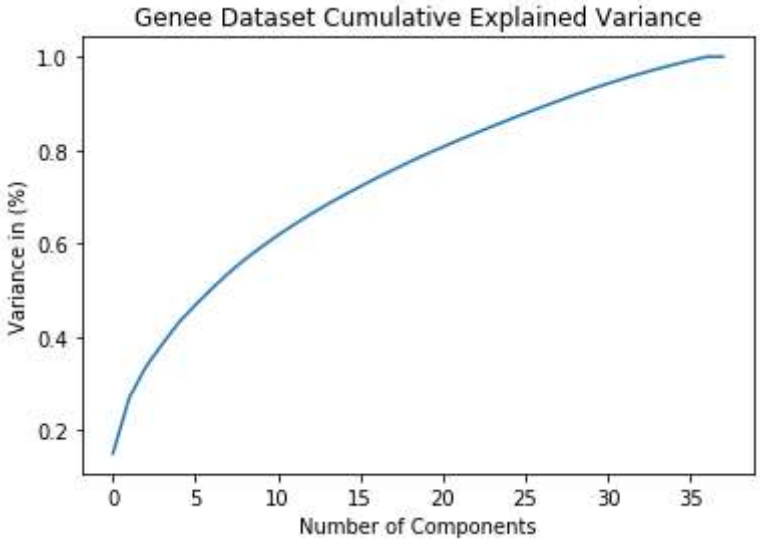
```
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

Restarting kernel...
```

```
In [1]: """
   ...: Created on Sat Mar 28 19:57:29 2020
   ...:
   ...: @author: ADITYA GATTU
   ...: """
   ...:
   ...:
   ...:
   ...: import numpy as np # linear algebra
   ...: import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
   ...: import matplotlib.pyplot as plt  # Graph Plots
   ...:
   ...:
   ...: # Importing the Datset
   ...: train_dataset = pd.read_csv('data_set_ALL_AML_train.csv')
   ...: test_dataset  = pd.read_csv('data_set_ALL_AML_independent.csv')
   ...:
   ...:
   ...: # Data Cleaning
   ...: # Remove Call Coulmn from Train Dataset
   ...: train_dataset1 = [col for col in train_dataset.columns if "call" not in col]
   ...: train_dataset  = train_dataset[train_dataset1]
   ...: train_dataset.head()
   ...:
   ...: # Remove Call Coulmn from Test Dataset
   ...: test_dataset1 = [col for col in test_dataset.columns if "call" not in col]
   ...: test_dataset  = test_dataset[test_dataset1]
   ...: test_dataset.head()
   ...:
   ...: # Dataset Transpose
   ...: # Transpose Train dataset
   ...: train_dataset.T.head()
   ...: train_dataset = train_dataset.T
   ...:
   ...: # Transpose Test dataset
   ...: test_dataset.T.head()
   ...: test_dataset = test_dataset.T
   ...:
   ...: # Drop Rows from Train dataset  Gene Description, Gene Accession Number
   ...: train_dataset2 = train_dataset.drop(['Gene Description','Gene Accession Number'],axis=0)
   ...:
   ...: # Drop Rows from Twat dataset  Gene Description, Gene Accession Number
   ...: test_dataset2  = test_dataset.drop(['Gene Description','Gene Accession Number'],axis=0)
   ...:
   ...: # Train dataset Convert to Numeric
   ...: train_dataset2.index = pd.to_numeric(train_dataset2.index)
   ...: train_dataset2.sort_index(inplace=True)
   ...: train_dataset2.head()
   ...:
   ...: # Test dataset Convert to Numeric
   ...: test_dataset2.index = pd.to_numeric(test_dataset2.index)
   ...: test_dataset2.sort_index(inplace=True)
   ...: test_dataset2.head()
   ...:
```

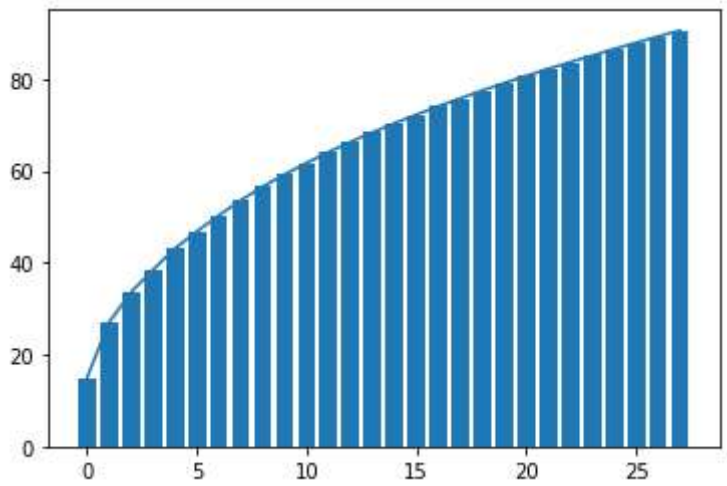
```
    ...:
    ...: # Import Response Variable
    ...: y = pd.read_csv('actual.csv')
    ...: y['cancer'].value_counts()
    ...:
    ...: # Replace (ALL, AML) with (0,1)
    ...: y = y.replace({'ALL':0,'AML':1})
    ...: labels = ['ALL', 'AML']
    ...:
    ...: # Train dataset
    ...: X_train = train_dataset2.reset_index(drop=True)
    ...: Y_train = y[y.patient <= 38].reset_index(drop=True)
    ...:
    ...: # Test dataset
    ...: X_test = test_dataset2.reset_index(drop=True)
    ...: Y_test = y[y.patient > 38].reset_index(drop=True)
    ...:
    ...: Y_test = Y_test.iloc[:,1].values
    ...: Y_train =Y_train.iloc[:,1].values
    ...:
    ...: # Feature Scaling
    ...: from sklearn.preprocessing import StandardScaler
    ...: sc= StandardScaler()
    ...: X_train= sc.fit_transform(X_train)
    ...: X_test= sc.fit_transform(X_test)
    ...:
    ...: # Feature Extraction With PCA
    ...: # Applying PCA
    ...: from sklearn.decomposition  import PCA
    ...: pca = PCA(n_components = None)
    ...: X_train_pca = pca.fit_transform(X_train)
    ...: X_train_pca
    ...:
    ...: #Eigenvalues (sum of squares of the distance between the projected data points and the
origin along the eigenvector)
    ...: print(pca.explained_variance_)
    ...:
    ...: #Explained variance ratio (i.e. how much of the change in the variables is explained by
change in the respective principal component): eigenvalue/(n variables)
    ...: print(pca.explained_variance_ratio_)
    ...:
    ...:
    ...: #Plotting the Cumulative Summation of the Explained Variance
    ...: plt.figure()
    ...: plt.plot(np.cumsum(pca.explained_variance_ratio_))
    ...: plt.xlabel('Number of Components')
    ...: plt.ylabel('Variance in (%)') #for each component
    ...: plt.title('Genee Dataset Cumulative Explained Variance')
    ...: plt.show()
    ...:
    ...: ## Calculating Explained Variance up to 90% of the variance
    ...: total = sum(pca.explained_variance_)
    ...: k = 0
    ...: current_variance = 0
    ...: while current_variance/total < 0.90:
    ...:       current_variance += pca.explained_variance_[k]
    ...:       k = k + 1
    ...:
    ...: k
    ...:
    ...: #Applying PCA for selecting N Components
    ...: from sklearn.decomposition  import PCA
    ...: pca = PCA(n_components = k )
    ...: X_train_pca = pca.fit_transform(X_train)
    ...: X_test_pca  = pca.transform(X_test)
```

```
    ...:
    ...: var_exp = pca.explained_variance_ratio_.cumsum()
    ...: var_exp = var_exp*100
    ...: plt.bar(range(k), var_exp);
    ...:
    ...: var1=np.cumsum(np.round(pca.explained_variance_ratio_, decimals=4)*100)
    ...: print(var1)
    ...: plt.plot(var1)
    ...:
    ...:
    ...: # Applying Data Model
    ...: # Fitting Logistic Regression to the train Set
    ...: from sklearn.linear_model import LogisticRegression
    ...: classifier =LogisticRegression(random_state=0)
    ...: classifier.fit(X_train_pca,Y_train)
    ...:
    ...: # Predicting the test set Results
    ...: Y_pred = classifier.predict(X_test_pca)
    ...:
    ...: # Making the Confusion Matrix
    ...: from sklearn.metrics import confusion_matrix
    ...: from sklearn.metrics import accuracy_score
    ...:
    ...: # Confusion Matirx
    ...: logit_cm = confusion_matrix(Y_test, Y_pred)
    ...: print(logit_cm)
    ...:
    ...: # Logit Accuuracy
    ...: logit_ac=accuracy_score(Y_test, Y_pred)
    ...: print(logit_ac)
    ...:
    ...:
    ...: # Applyig Random Forest
    ...: from sklearn.ensemble import RandomForestClassifier
    ...:
    ...: classifier = RandomForestClassifier(max_depth=2, random_state=0,oob_score=True)
    ...: classifier.fit(X_train_pca, Y_train)
    ...: print(classifier.oob_score_)
    ...: # Predicting the Test set results
    ...: Y_pred = classifier.predict(X_test_pca)
    ...:
    ...: # Making the Confusion Matrix
    ...: from sklearn.metrics import confusion_matrix
    ...: from sklearn.metrics import accuracy_score
    ...:
    ...: # Confusion Matirx
    ...: rf_cm = confusion_matrix(Y_test, Y_pred)
    ...: print(rf_cm)
    ...:
    ...: # Logit Accuuracy
    ...: rf_ac=accuracy_score(Y_test, Y_pred)
    ...: print(rf_ac)
[1.09735759e+03 8.76976485e+02 4.83272186e+02 3.57658145e+02
 3.39170426e+02 2.72510227e+02 2.55597771e+02 2.40859971e+02
 2.18576598e+02 1.93681882e+02 1.83741623e+02 1.72634428e+02
 1.61466953e+02 1.52866654e+02 1.42171050e+02 1.38629132e+02
 1.35241393e+02 1.25441555e+02 1.24851480e+02 1.20423734e+02
 1.12430908e+02 1.11931494e+02 1.06610477e+02 1.04779826e+02
 1.03617441e+02 1.00934278e+02 9.86526997e+01 9.56118071e+01
 9.54129156e+01 9.14541425e+01 8.49250449e+01 8.17690665e+01
 7.56789591e+01 7.30684864e+01 6.76502725e+01 6.28995436e+01
 6.11190279e+01 2.16415626e-28]
[1.49877930e-01 1.19778111e-01 6.60056806e-02 4.88492199e-02
 4.63241532e-02 3.72196529e-02 3.49097368e-02 3.28968370e-02
 2.98533570e-02 2.64532179e-02 2.50955698e-02 2.35785408e-02
```

```
 2.20532785e-02 2.08786432e-02 1.94178295e-02 1.89340717e-02
 1.84713718e-02 1.71329025e-02 1.70523096e-02 1.64475646e-02
 1.53558984e-02 1.52876881e-02 1.45609395e-02 1.43109079e-02
 1.41521484e-02 1.37856801e-02 1.34740603e-02 1.30587329e-02
 1.30315682e-02 1.24908759e-02 1.15991269e-02 1.11680809e-02
 1.03362895e-02 9.97974912e-03 9.23972538e-03 8.59086723e-03
 8.34768304e-03 2.95582098e-32]
```




```
[14.99 26.97 33.57 38.45 43.08 46.8  50.29 53.58 56.57 59.22 61.73 64.09
 66.3  68.39 70.33 72.22 74.07 75.78 77.49 79.13 80.67 82.2  83.66 85.09
 86.51 87.89 89.24 90.55]
[[19  1]
 [ 2 12]]
0.9117647058823529
0.8157894736842105
[[20  0]
 [12  2]]
0.6470588235294118
C:\Users\ADITYA GATTU\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432:
FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this
warning.
  FutureWarning)
C:\Users\ADITYA GATTU\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```



In [2]: