**Inputs :**

Images.csv : Contains the images which are represented by an image ID and the corresponding class.

EdgeHistogram.csv : Contains the feature data for the Edge Histogram feature for the images.

\*The given inputs Images and EdgeHistogram are read into data frames.

We have used LabelEncoder to encode the data and is stored in defined variable ‘label’.

\*In Images (df1), names of the class are duplicates, So that to know the actual count duplicates are removed.

\*Now the indexes of Images (df1) are discontinuous ,So they are removed.

\*The train\_test\_split function is imported from the module sklearn.preprocessing.

\*X-Input and Y- Output.

\*Train sets are x\_train and y\_train.

\*Test sets are x\_test and y\_test.

We have used test size =30% and train size=70%

\*Now for image classification **K-Nearest Neighbors(KNN)** is used.

\*KNeighborsClassifier (Class) is required to perform this, which is imported from the module sklearn.neighbors.

\* To measure the error in each of the nearest neighbour new variable ‘error\_rate’ is initialised with list datatype.

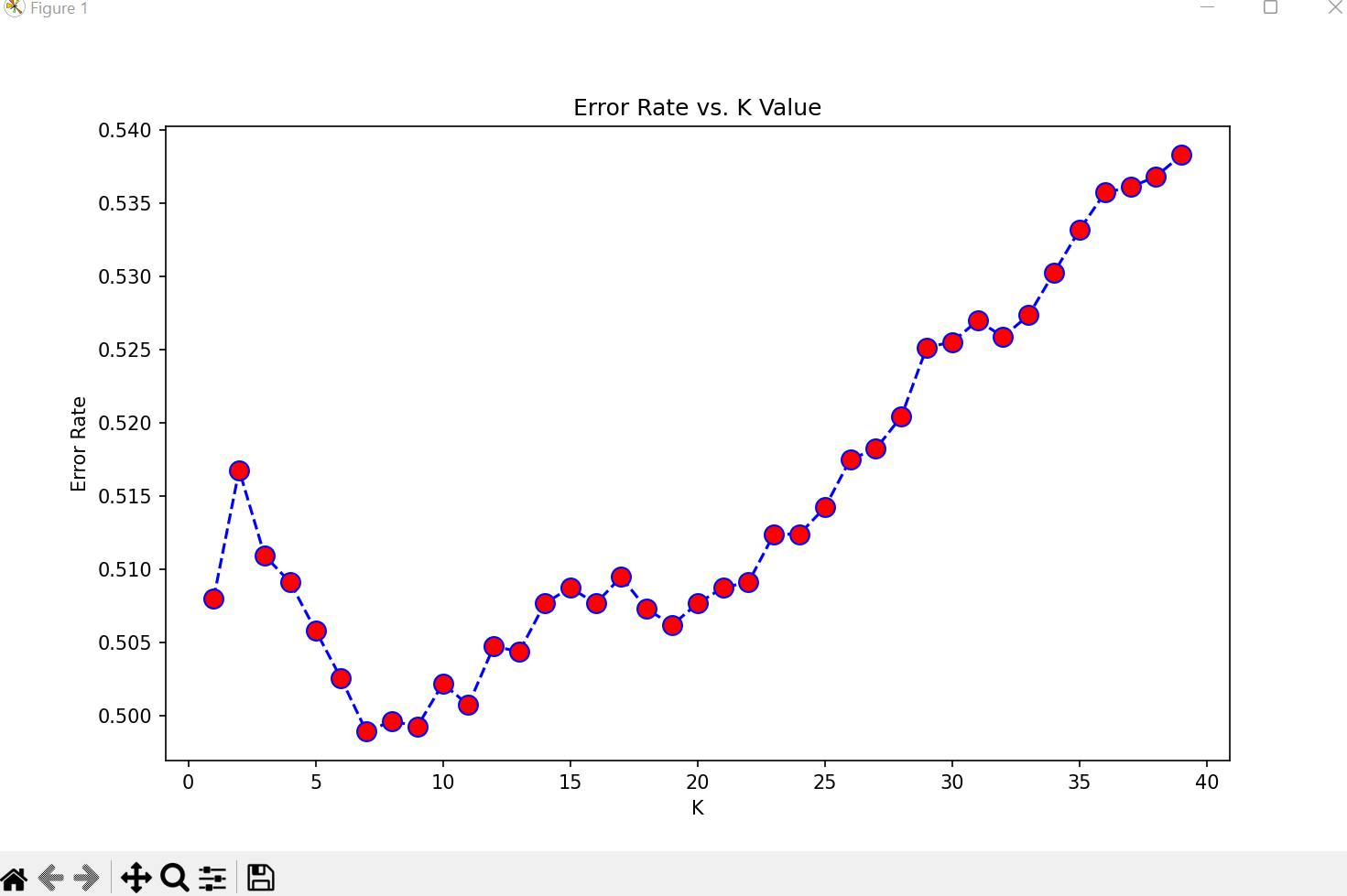
\* To check the neighbours from one to forty, a for loop is used.

\* Here forty different runs are to be executed for each amount of training images.

\*Here only one parameter used to optimize the classification method.

\*Comparing ‘pred\_i’ and ‘y\_test’, then appending into ‘error\_rate’ list , if both are equal then 0 is added else the error is appended

* Here we are plotting Error Rate vs. K Value’.



* \*Next we are presenting a classification report of the K Nearest Neighbour.

