**Q1. K-NN:**

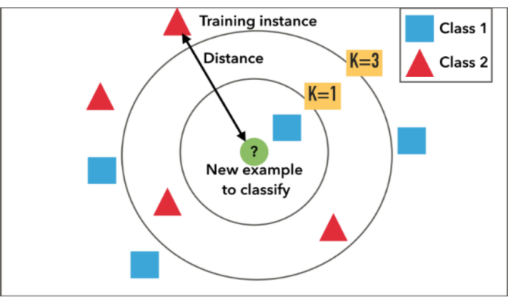
**Check out the provided program on k-NN, applied to the famous Iris Data set.Two parameters are interesting: (i) the split which is the size of the training subset v test subset (split = .67) means roughly 2/3rds training, 1/3rd testing, (ii) k which is the size of nearest neighbors. Note, the Accuracy of model is determined for test sets; it measures how well it classifications work for the unseen set.**

**Taking ideas about cross-validation into account; your job is first to systematically vary the size of the split ; exploring a decent number of other values for it >0.0 and <0.9 (to be chosen by you). Also to systematically vary k on five selected values between 1 and 20.**

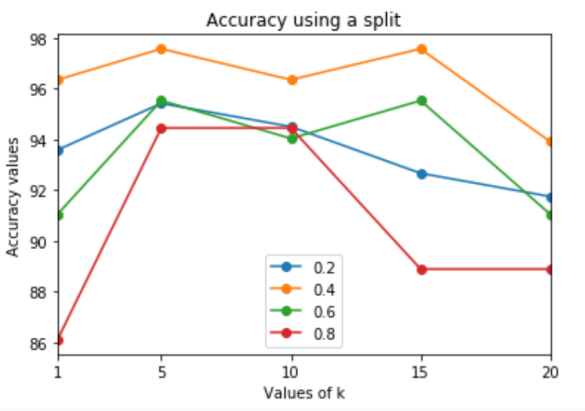
**Then plot the accuracy in a graph for these parameter changes Now, using this data set, describe an algorithm for doing a 5-fold cross validation on this data set. See can you implement it in Python.**

**Ans1.**

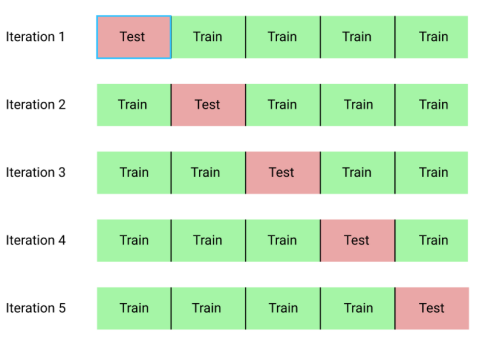
* **k-nearest neighbors algorithm**(k-NN) is a non-parametric method used for classification and regression algorithm. It is a supervised algorithm in which an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. It makes the use of distance metric to check the similarity among the neighbors and the test sample.*[1]*
* Split value determines the split percentage of training and test set.
* k in k-NN determines the number of data points closer to the test sample. Accuracy tells how well the test point is classified.



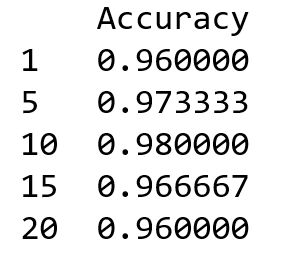
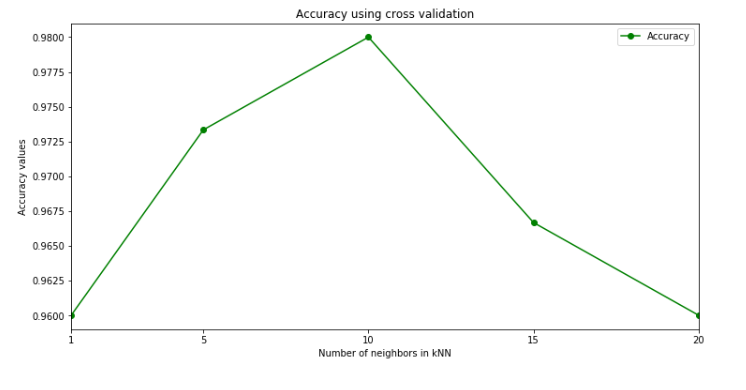
* Cross Validation is a model validation techniques for assessing how the results of a model will generalize to an independent data set. It define a data set to test the model in the training phase in order to limit problems like overfitting, underfitting and get an insight on how the model will generalize to an independent data set.
* Split values are considered as follows: 0.2,0.4,0.6,0.8
* For every split value, k values are as follows: 1,5,10,15,20
* When the value for k gets increased, accuracy gets decreased as the algorithm will look for more data points in the neighborhood and hence increase the (underfitting)generalisability at the cost of variance.
* When the value for the split is eventually increasing(training dataset gets increased), the larger the percentage of training sets, the better the classification because the classification accuracy will be relatively close to the optimum value and then it can obtain an acceptable optimum value.



* **5 fold Cross- Validation Algorithm**: *[2][3]*



* Cross Validation has a single feature known as ‘k’ that indicates the number of partitions that a given data set is to be split into.That’s why it is called k-fold cross-validation.
* Dataset gets divided into K equal folds, here it is given as 5 fold, so the dataset is divided into 5 partitions. For every iteration(till k), consider 1 partition of the dataset as the test set and remaining all portions as training set.
* Then a model is fitted on the training set and tested for the test set.
* Evaluation score is calculated for every iteration collectively using metric like mean.
* The difference between the Hold Out and Cross validation is that here, split value is specified. Instead it automatically divides the dataset but in ‘K’ equal parts.
* The results of a k-fold cross-validation run are often summarized with the mean of the model skill scores.
* Every partition of the dataset becomes test set once and remains in the training set for (K-1) times.(Bias gets reduced and the variance gets increased as the value for ‘K’ increases.)
* **Implementation of Cross Validation** in Python is done using the cross\_val\_score from sklearn.model\_selection module.
* KNeighborsClassifier is a classifier implementing the k-NN votes.
* cross\_val\_score is a method which accepts the classifier, dataset and number of partitions.
* For number of neighbors as [1,5,10,15,20] and number of folds/parts as 5, mean accuracy for every neighbor is calculated. Graph is shown below.
* Accuracy value increases gradually as the k neighbors increases till a certain point but then decreases.
* Cross validation is much better than the train\_test method as
* It trains the model on multiple train-test splits. This shows how well the model performs on unseen data whereas Hold-out is dependent on just one train-test split. That makes the hold-out method score dependent on how the data is split into train and test sets.

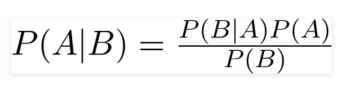
=====================================================================================

**Q2. Bayes Classifiers**

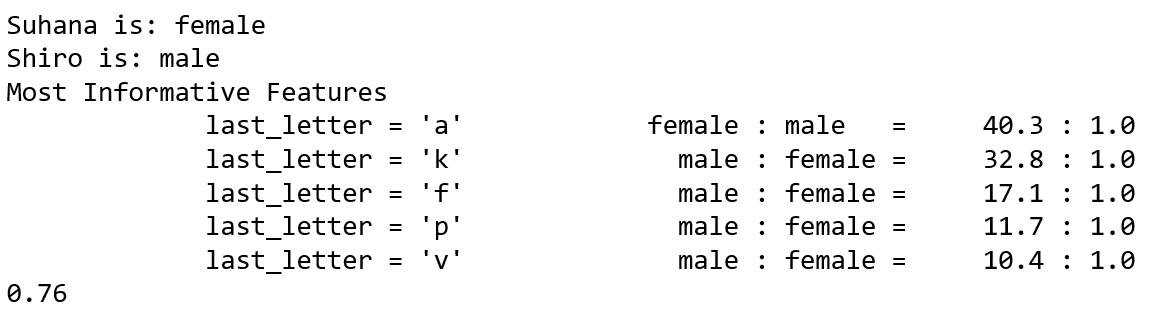
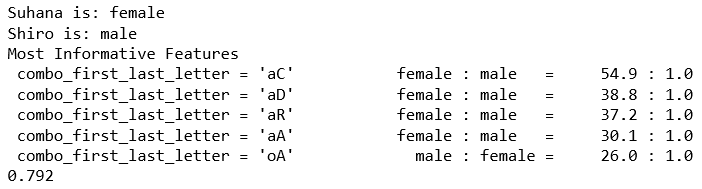
**Have a look at the nltk Bayes classifier that does the prediction of male/female names based on the last letter in the name.**

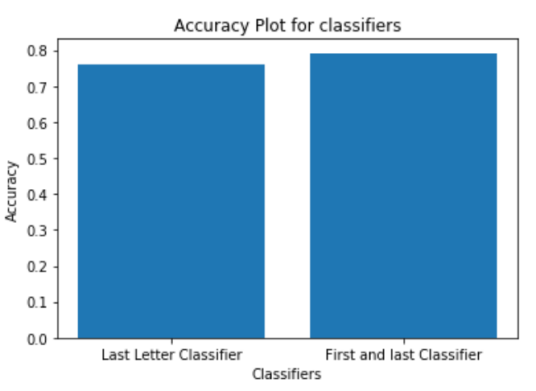
**Think of a new feature that you could extract from the data-set; define a function for it (modifying gender\_features) and see what is learned from it in the classification.**

**Ans2. Naive Bayes Classifier**: A Naive Bayes classifier is a probabilistic machine learning model used for classification task based on the Bayes theorem.



* Accuracy by extracting only the last letter from the corpus comes out to be 76%.
* Features extracted are the combination of first letter and the last letter from the corpus of words and the accuracy comes to be 79.2%. Graph indicating the accuracy of both the classifiers is shown below:



=====================================================================================

**Q3. SVMs**

**So, this is just a quick use of an SVM; to show you how easy it is. It involves learning digit recognition.**

**NB to install packages: sklearn, matplotlib, scipy.**

1. **Now run three different training configurations and then test the outputs for 5-10 different digits; report results**

**Ans3.**

**Support Vector Machines**, SVM are supervised learning models for classification and regression problems. They are used for both linear and non-linear problems.[4]

The idea behind SVM is to create hyperplane that separates the classes in the case of classification. The data points lying on the hyperplane or closer to the hyperplane are called the support vectors. They influence the position and orientation of the hyperplane.The goal is to maximize the margin between the points on either sides of the hyperplane. Thus model can easily predict target labels for the unseen data points.

**Digit Recognition**: It is the working of a machine to train itself to recognize the digits for online handwriting recognition as the handwritten digits can vary in size, shape, width, orientation so the issue is wrongly classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8 etc. *[5]*

**Hyper parameter C**: The C parameter determines the influence of misclassification, it shows how much misclassification can be avoided in the svm while classifying the training example.

Larger values of C, smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.

Smaller values for C cause the svm to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

**Hyper parameter gamma**: gamma is a parameter used for non linear hyperplanes.

Larger values of gamma, model tries to exactly fit the training data set.

Increasing gamma can lead to overfitting as the classifier tries to perfectly fit the training data.

3 different training configurations (combination of parameters C and gamma) are taken as follows:

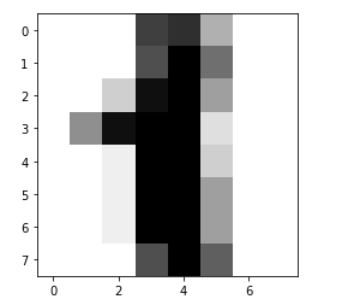
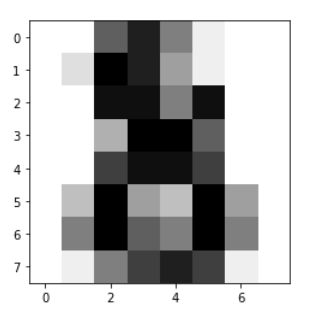
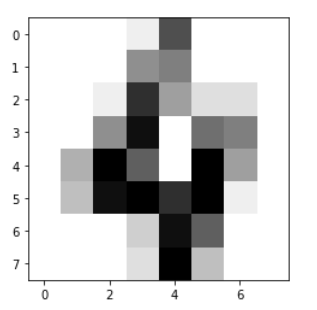
C = 0.01 and gamma = 0.5

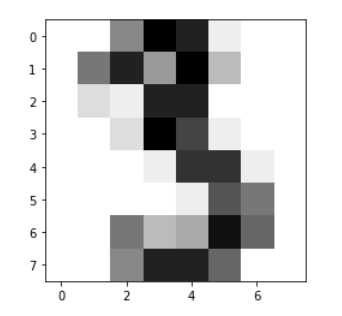
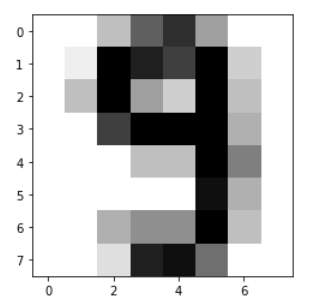
C = 10 and gamma = 0.005

C = 1000 and gamma = 0.99999

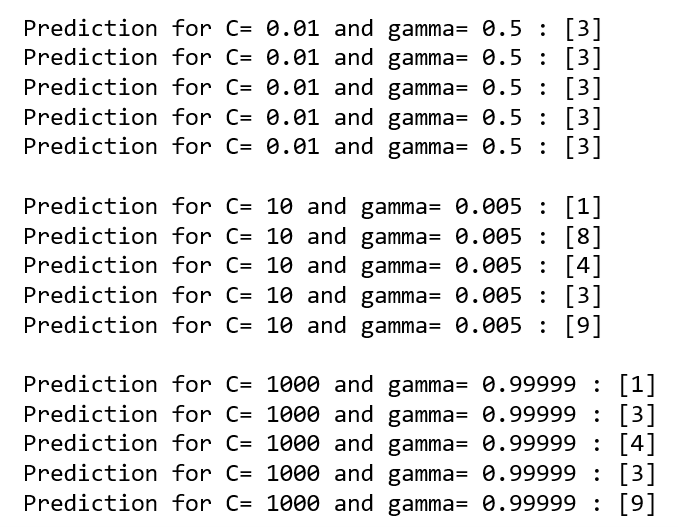
5 digits considered are as follows: 1,3 4, 8,9.

The original images of the digits taken are as follows:

The predicted digit by SVM using every combination of C and gamma are shown below:



First training configuration: Low C and low gamma

It recognizes the digits incorrectly because:

Low value of C implies the probability of the outliers is increased, and the general decision boundary is found. It does not avoid the misclassification of the data point.

Second training configuration: High C and low gamma

It recognizes the digits correctly because:

High C allows data samples to be placed in different classes. It avoids the misclassification of the data points in the classes . High C, low gamma implies the decision boundary is found more carefully.

Low gamma means distance from data points that are far from the hyperplane whereas high value only allows the data points closer to hyperplane.

Third training configuration: High C and high gamma

It recognizes the digits slightly incorrect because:

High C allows data samples to be placed in different classes. It avoids the misclassification of the data points in the classes . High C, high gamma implies the general decision boundary is found.

High gamma considers the data points closer to the hyper plane only leaving behind the data points far away from hyperplane.

=====================================================================================

**REFERENCES**

[1] D.Soni,”k-Nearest-Neighbors (kNN) ”, 31, May 2018[Online]. Available: <http://blog.xnextcon.com/?p=213> [Accessed Oct. 26, 2019]

[2] J.Brownlee,”A Gentle Introduction to k-fold Cross-Validation”, 23, May 2018[Online]. Available: <https://machinelearningmastery.com/k-fold-cross-validation/> [Accessed Oct. 26, 2019]

[3] R. Shaikh, “Cross Validation Explained: Evaluating estimator performance”, 26, Nov 2018[Online]. Available: <https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85> [Accessed Oct. 26, 2019]

[4] L.Schultebraucks, “Introduction to Support Vector Machines”, 22 Sep 2017[Online]. Available: <https://medium.com/@LSchultebraucks/introduction-to-support-vector-machines-9f8161ae2fcb> [Accessed Oct. 26, 2019]

[5] H.Beniwal, “Handwritten Digit Recognition using Machine Learning”, 22 Dec 2018[Online]. Available: <https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64> [Accessed Oct. 26, 2019]