

University of Science and Technology of Hanoi



Machine Learning and Data Mining 2
Labwork 3 - Classification I

BA12-118 Nguyen Phuc Minh
BA12-066 Nguyen Hoang Ha

Table of contents

I. K-nearest neighbor classification.....	3
1. Import libraries and datasets.....	3
2. Data cleaning and preprocessing.....	3
3. Data preprocessing.....	3
4. Implement K-nearest.....	3
Original datasets.....	4
After normalization.....	6
Use PCA and SVD.....	8
K-cross validation.....	11
Leave-one-out.....	12
II. SVM classifier.....	14
1. Social Network Ads.....	14
2. Epileptic Seizure Recognition.....	16

I. K-nearest neighbor classification

1. Import libraries and datasets

- Numpy, Pandas, Matplotlib, Seaborn, Sklearn

2. Data cleaning and preprocessing

- Two datasets: [Social Network Ads](#) and [Survey of Lung Cancer](#)
- Check the missing and duplicate values.

3. Data preprocessing

- Label encoding the categorical features.
- Scale the features using the standard scaler.

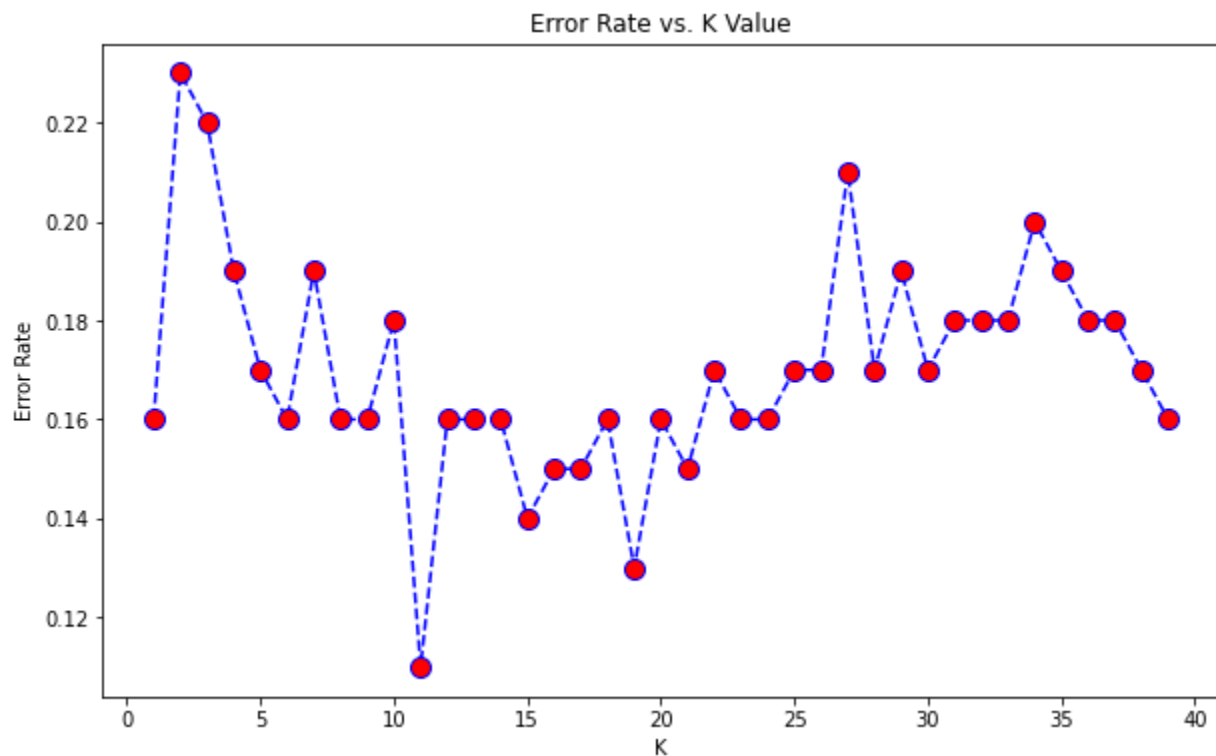
4. Implement K-nearest

K-nearest neighbors (KNN) is a non-parametric, supervised learning algorithm used for classification and regression. It is one of the simplest and most effective classification algorithms, and it is widely used in a variety of applications, including pattern recognition, data mining, and intrusion detection.

The KNN algorithm works by classifying a new data point based on the majority class of its k nearest neighbors in the training set. The k nearest neighbors are the k data points in the training set that are most similar to the new data point.

Now let's apply KNN to two selected datasets and analyze them.

Original datasets

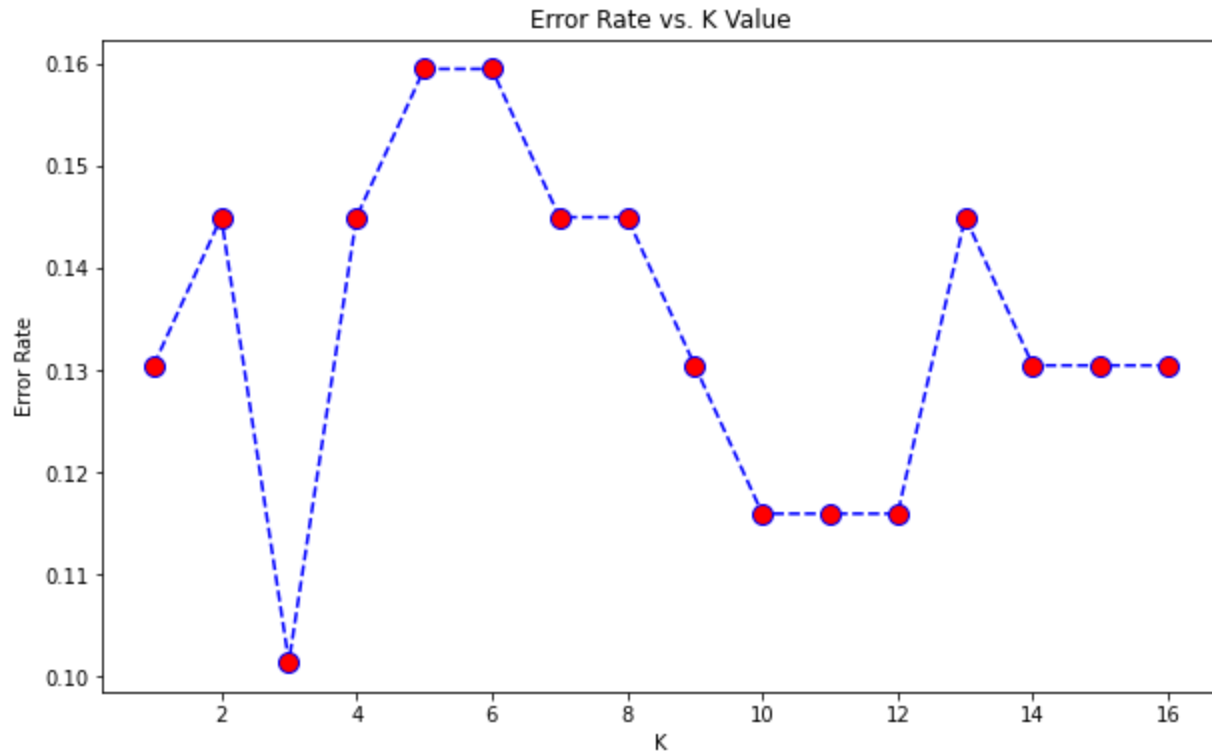


Social Network Ads

The graph shows that the error rate generally decreases as the value of k increases, until it reaches a minimum point, and then starts to increase again.

The error rate is minimum at $K = 11$

K = 11	Precision	Recall	F1-core	Support	Confusion Matrix
0	0.89	0.96	0.92	68	
1	0.89	0.75	0.81	32	
Accuracy			0.89	100	
Macro avg	0.89	0.85	0.87	100	
Weighted avg	0.89	0.89	0.89	100	



Survey Lung Cancer

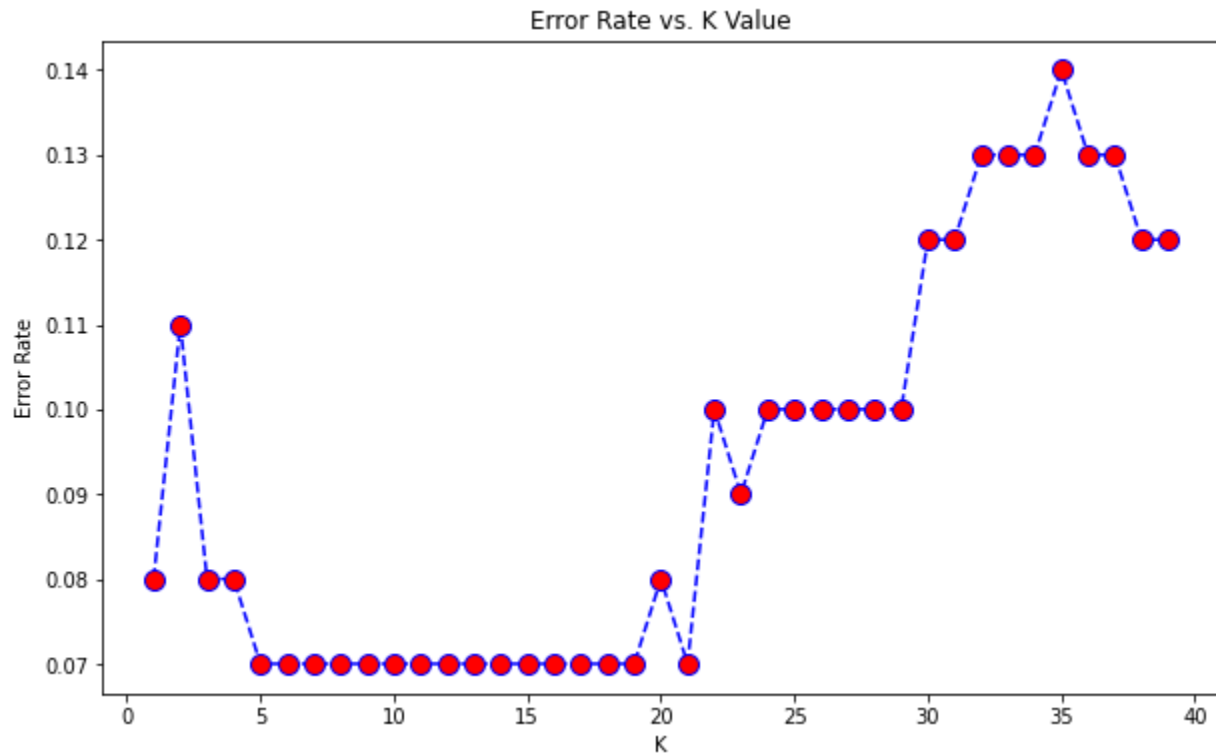
From $k = 14$ onwards the error remains around 0.13

The error rate is minimum at $K = 3$

K = 11	Precision	Recall	F1-core	Support	Confusion Matrix	
0	0.75	0.33	0.46	9		
1	0.91	0.98	0.94	60		
Accuracy			0.90	69		
Macro avg	0.83	0.66	0.70	69		
Weighted avg	0.89	0.90	0.88	69		

After normalization

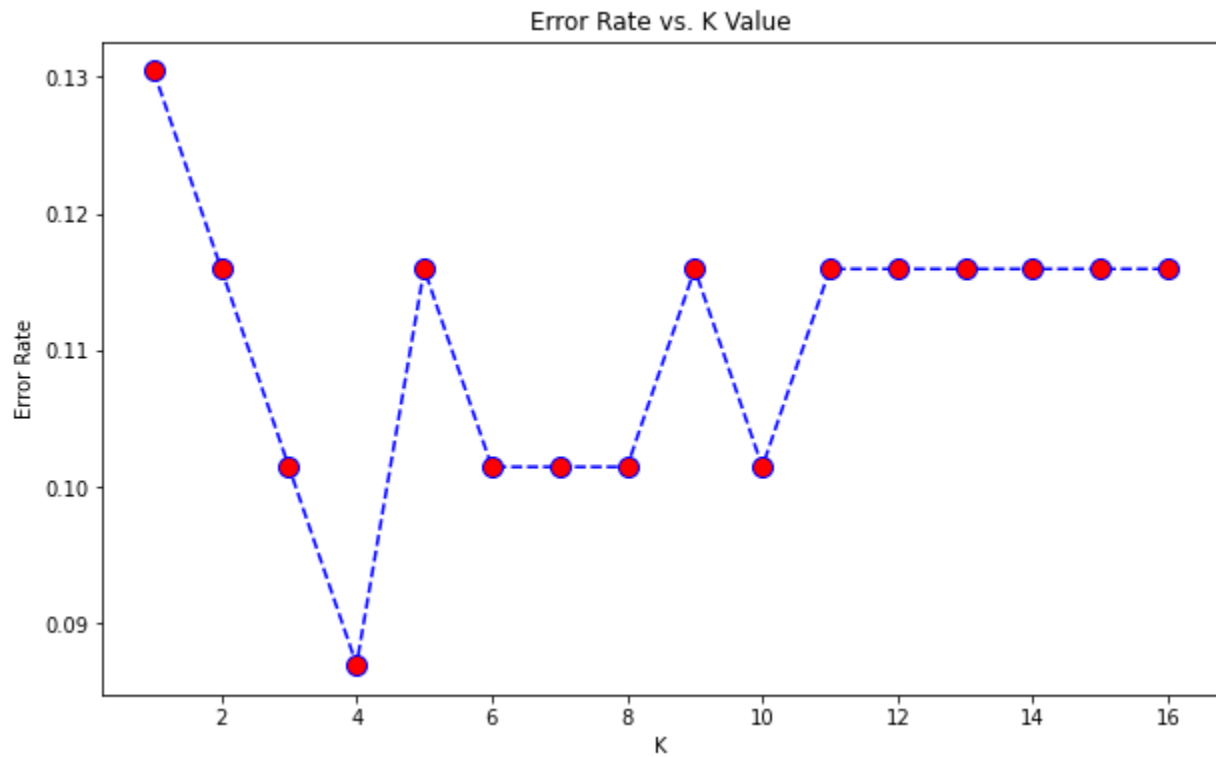
Data normalization can be important for KNN because it relies on the distance between data points to make predictions. If the features are not normalized, then features with larger scales will have a greater impact on the distance calculation than features with smaller scales. This can lead to inaccurate predictions.



Social Network Ads

The error rate is minimum at $K = 5$ and stays the same up to $K = 19$ then starts to increase.

K = 5	Precision	Recall	F1-core	Support	Confusion Matrix
0	0.96	0.94	0.95	68	
1	0.88	0.91	0.89	32	
Accuracy			0.93	100	
Macro avg	0.92	0.92	0.92	100	
Weighted avg	0.93	0.93	0.93	100	



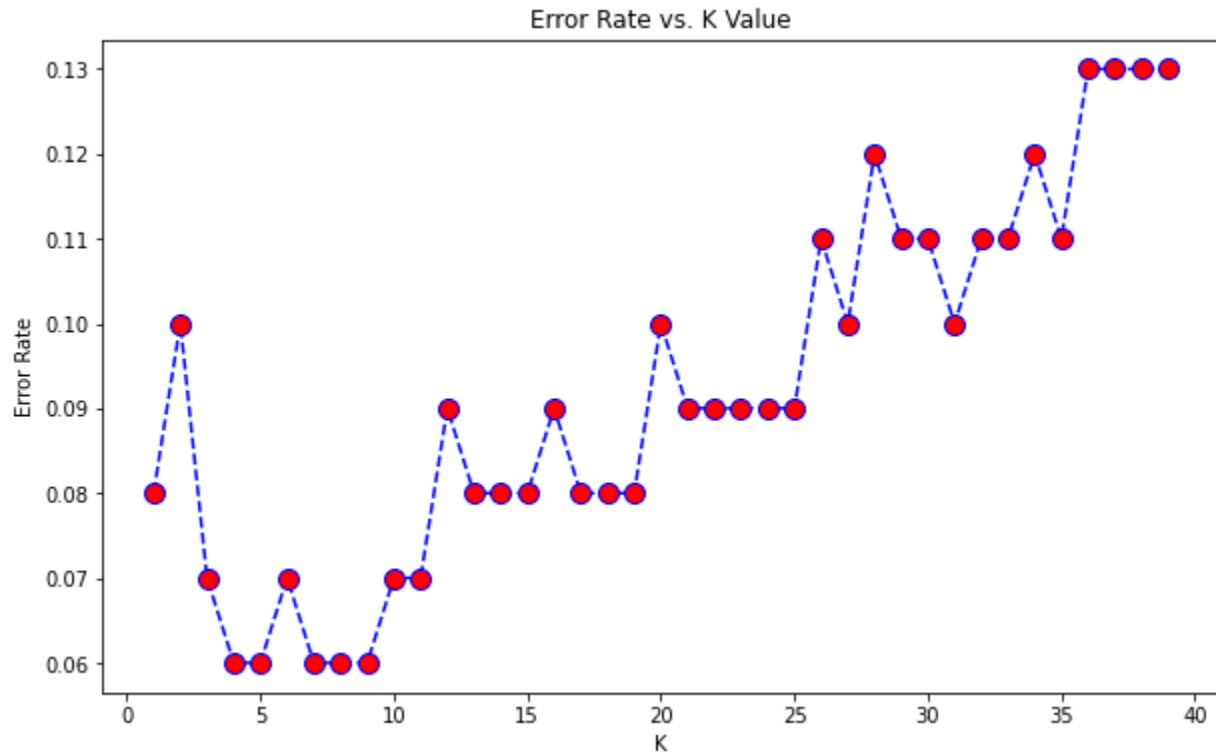
Survey Lung Cancer

Error rate is minimum at K = 4, from k = 14 onwards the error remains around less than 0.12

K = 4	Precision	Recall	F1-core	Support	Confusion Matrix	
0	0.62	0.89	0.73	9		
1	0.98	0.92	0.95	60		
Accuracy			0.91	69		
Macro avg	0.80	0.90	0.84	69		
Weighted avg	0.93	0.91	0.92	69		

Use PCA and SVD

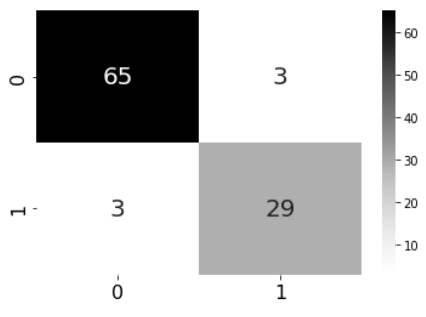
Reduce the dimension of the dataset to 2 components before applying K-nn

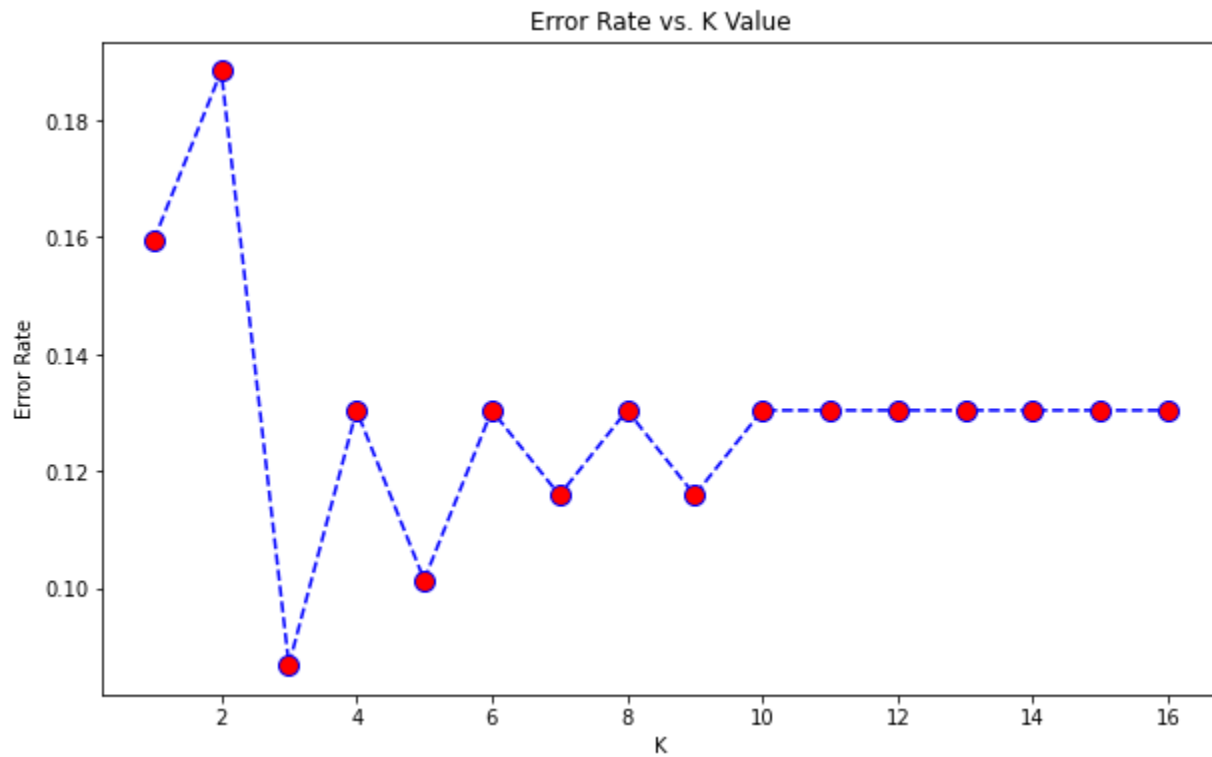


Social Network Ads

→ Applying K-nn classification on this dataset after PCA slightly decreases the error rate.

The error rate is minimum at K = 4 and then increases when K increases.

K = 5	Precision	Recall	F1-core	Support	Confusion Matrix
0	0.96	0.96	0.96	68	
1	0.91	0.91	0.91	32	
Accuracy			0.94	100	
Macro avg	0.93	0.93	0.93	100	
Weighted avg	0.94	0.94	0.94	100	



Survey Lung Cancer

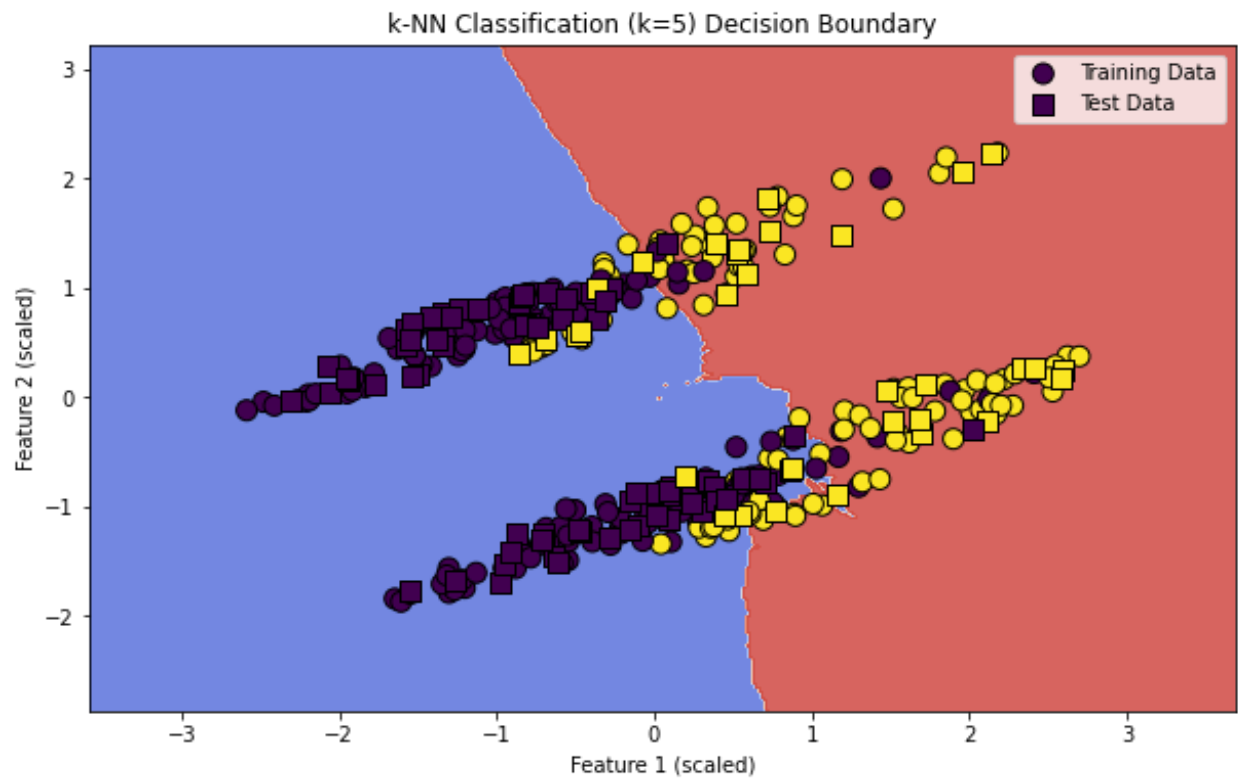
The error rate is minimum at K = 3, then change it up and down and gradually keep it at about 0.13

K = 3	Precision	Recall	F1-core	Support	Confusion Matrix
0	0.80	0.44	0.57	9	
1	0.92	0.98	0.95	60	
Accuracy			0.91	69	
Macro avg	0.86	0.71	0.76	69	
Weighted avg	0.91	0.91	0.90	69	

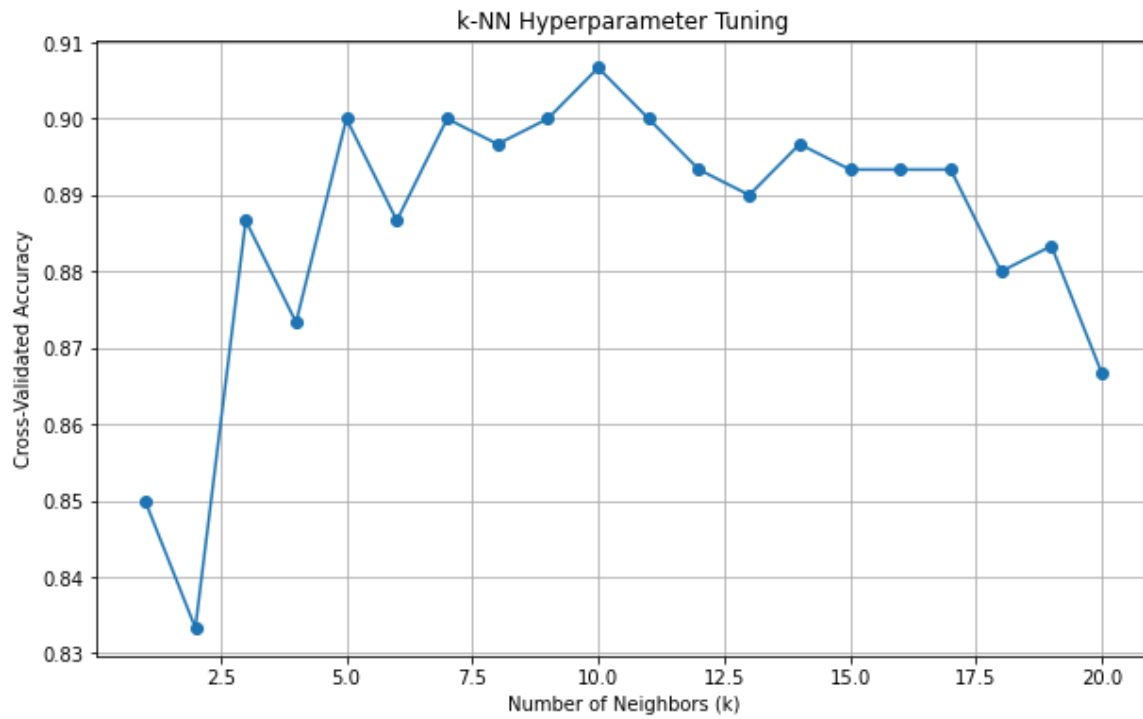
→ Using PCA for this data does not have much impact on the error rate of classification.

- If the error rate curve after PCA reaches a lower minimum point compared to without PCA, it suggests PCA helped improve classification accuracy.
- If the error rate remains similar or even increases after PCA, it might indicate that the discarded information was relevant for classification, or the chosen number of PCs wasn't optimal.

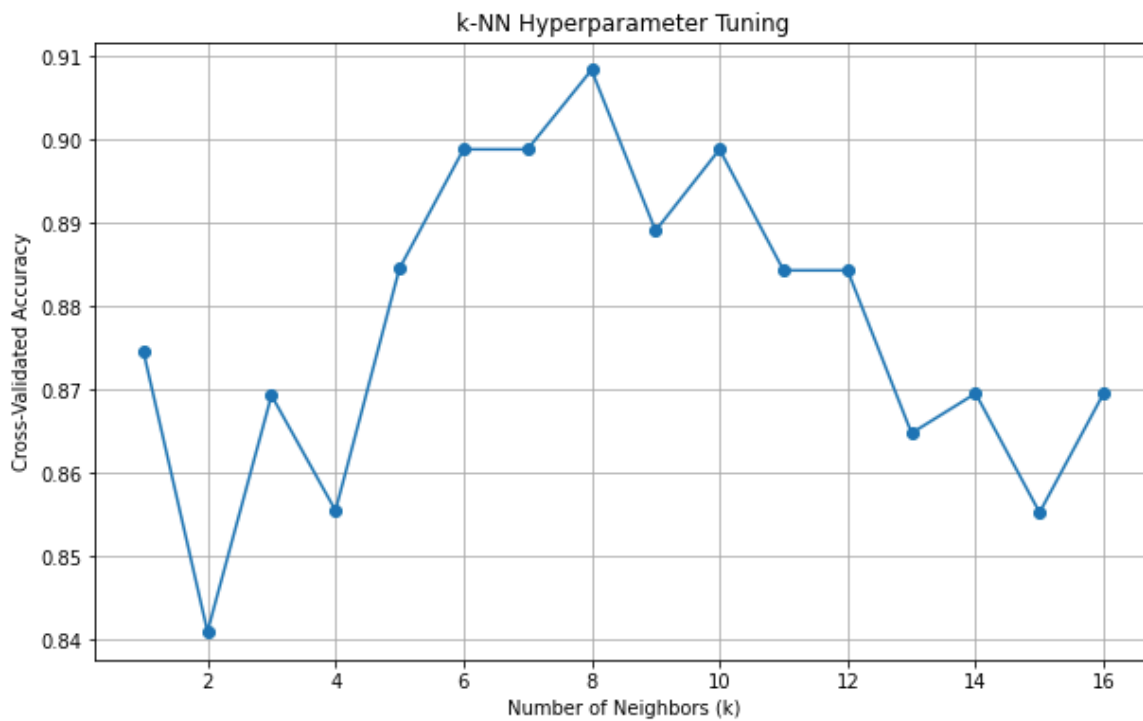
The figure shows the distribution of data after PCA and classification.



K-cross validation



Social Network Ads



Survey Lung Cancer

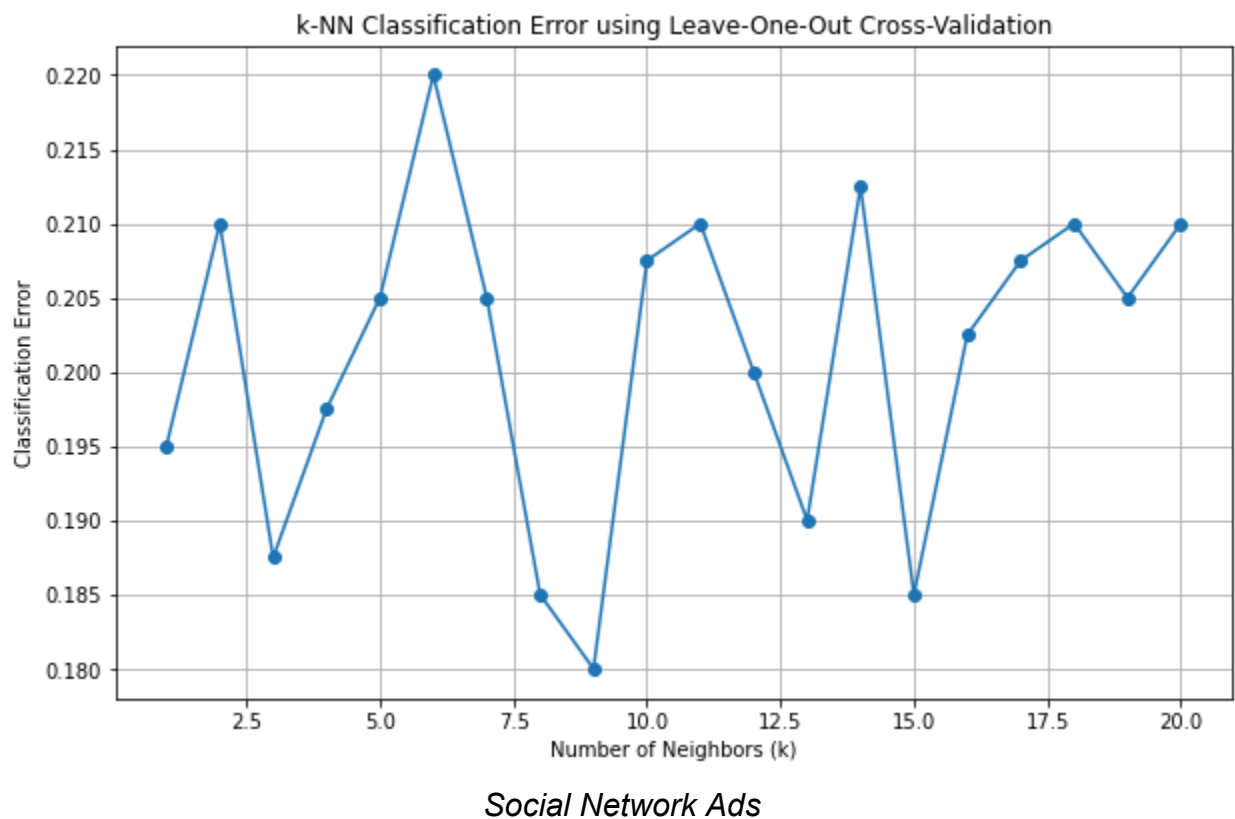
The accuracy generally increases as the number of neighbors (K) increases, reaches a peak, and then starts to decrease. This is a common pattern observed in KNN models.

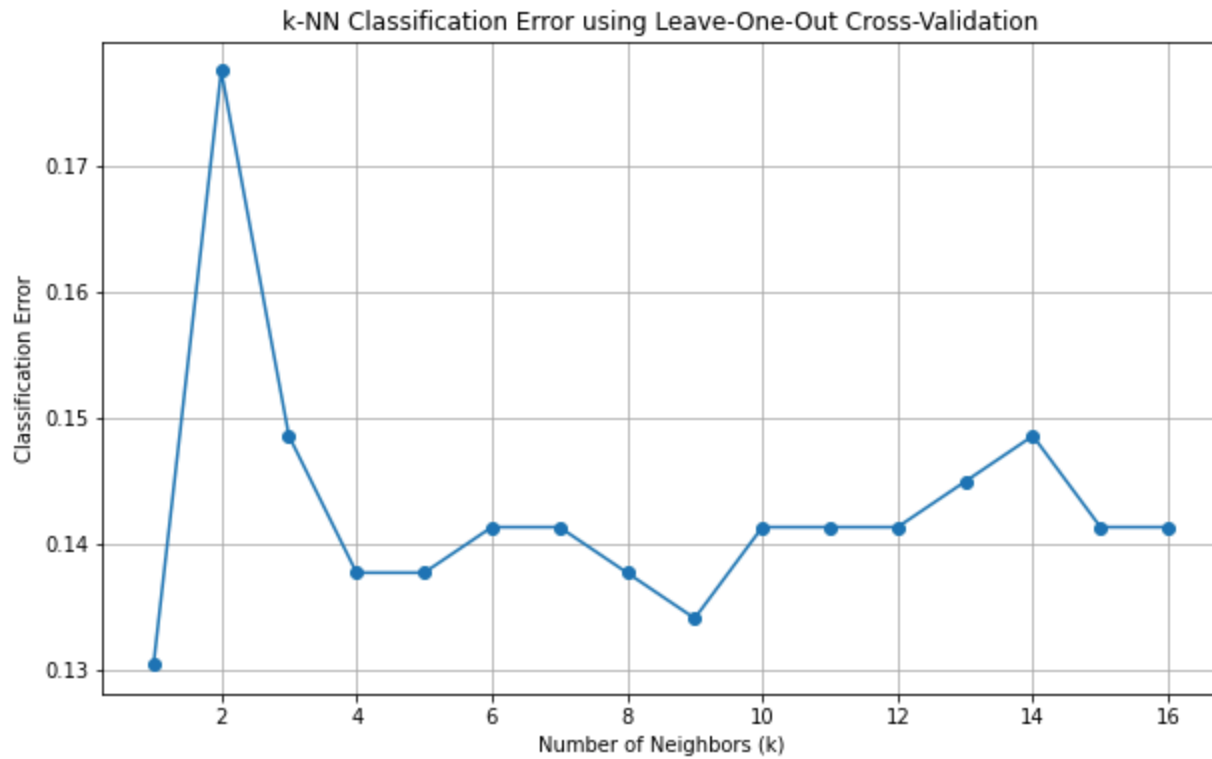
The peak accuracy on the graph is considered the optimal value for K (K=10)

Small values of k: When K is too small, the model is too sensitive to noise in the data. If a single noisy data point is one of the k nearest neighbors of a test point, it can cause the model to make an incorrect prediction.

Large values of k: When K is too large, the model becomes less sensitive to the local variations in the data, but it can also start to overfit the training data. This means that the model will be good at classifying the data it was trained on, but it may not generalize well to new data.

Leave-one-out





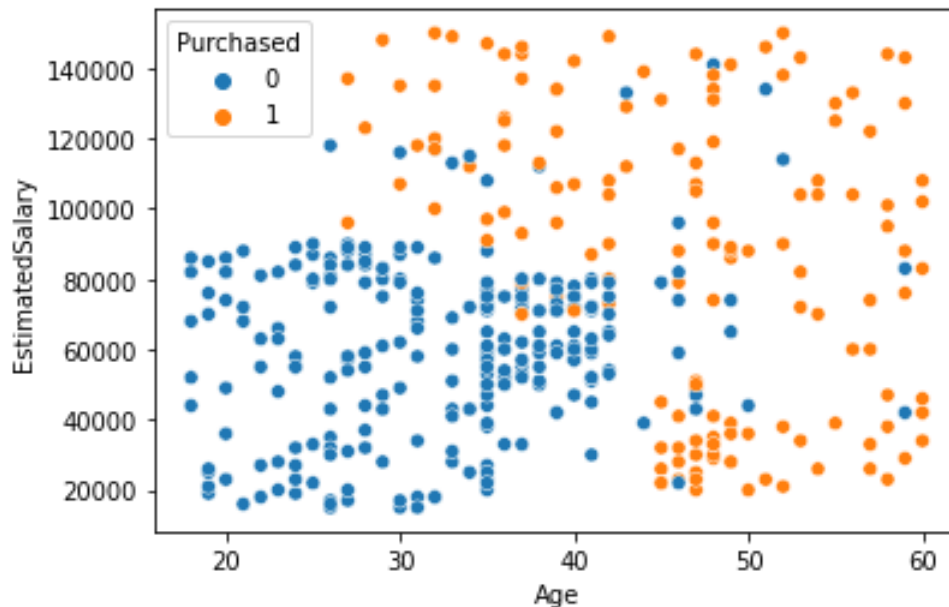
Survey Lung Cancer

Due to the nature of LOOCV using almost the entire dataset for training each time, the error rate can be highly variable compared to other cross-validation techniques like k-fold cross-validation. This can be seen in the fluctuating pattern of the error rate across different k values in the graph.

II. SVM classifier

1. [Social Network Ads](#)

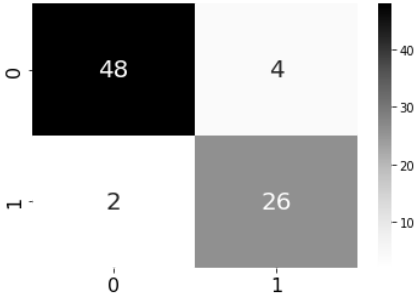
→ Data distribution: visualizing the relationship between 'Age' and 'Estimated Salary', with the colors representing the 'Purchased' column shows that the data set is not linearly separable. Few data points for the two classes overlap, suggesting that a straight line cannot effectively separate the clusters.



→ Since this dataset is non-linear separate, we use the 'rbf' kernel to process it. Then use GridSearchCV to find values that match the dataset for the following two parameters:

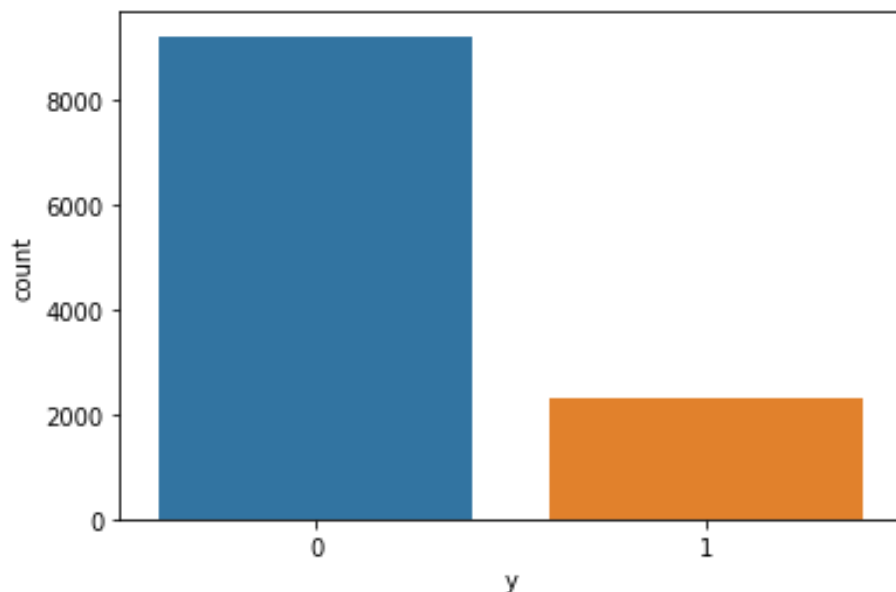
- ◆ C (Regularization Parameter): Controls the trade-off between achieving a low error on the training data and minimizing the complexity of the model (i.e., avoiding overfitting).
- ◆ Gamma (Kernel Coefficient): Defines how far the influence of a single training example reaches.

→ Implement SVM:

	Precision	Recall	F1-core	Support	Confusion Matrix
0	0.96	0.92	0.94	52	
1	0.87	0.93	0.80	28	
Accuracy			0.93	80	
Macro avg	0.91	0.93	0.92	80	
Weighted avg	0.93	0.93	0.93	80	

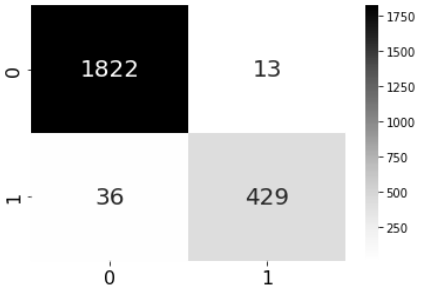
2. [Epileptic Seizure Recognition](#)

- Handle multi-class: This dataset includes 5 classes (labeled from 1 to 5) for detecting seizures. However, we must reclassify the data since the SVM model is designed for binary classification problems. Classes 2, 3, 4, and 5 consist of subjects who do not have epileptic seizures, while only class 1 consists of subjects with epileptic seizures.
- Therefore, to make the data compatible with the SVM model, we have reclassified the data into two categories: class 0 (comprising original classes 2, 3, 4, and 5, representing non-epileptic subjects) and class 1 (comprising original class 1, representing epileptic subjects)



- Data distribution: With a dataset containing 11,000 rows, we decided to assess the linear separability of the data by running three different models: LogisticRegression (82.89%), LinearSVC (82.34%), and SVC with an RBF kernel (97.86%). The significantly higher accuracy of the SVC model with the RBF kernel compared to the logistic regression and linear SVC models suggests that the data is non-linearly separable.
- Use GridSearchCV to find 'C' and 'gamma' values that match the dataset.

→ Implement SVM:

	Precision	Recall	F1-core	Support	Confusion Matrix
0	0.98	0.99	0.99	1835	
1	0.97	0.92	0.95	465	
Accuracy			0.98	2300	
Macro avg	0.98	0.96	0.87	2300	
Weighted avg	0.98	0.98	0.98	2300	