

Student's Name

Instructor's Name

Course

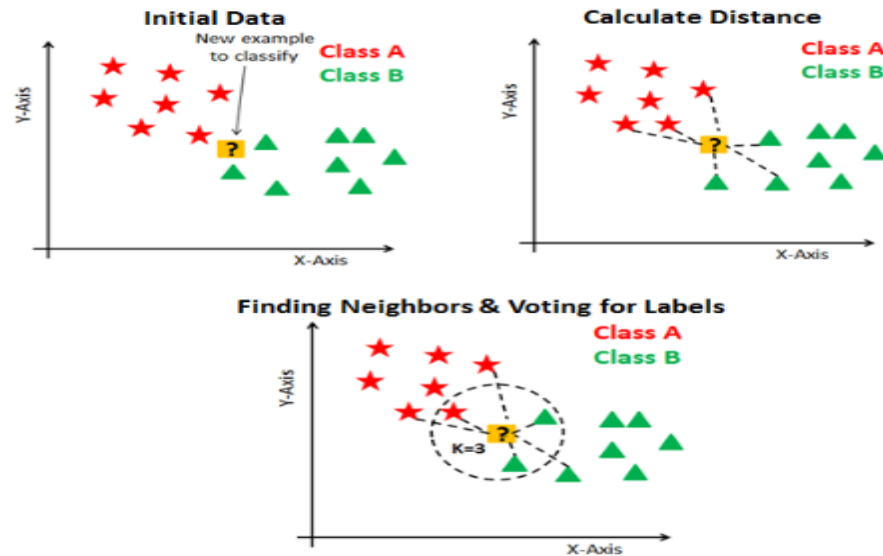
Date

K-Nearest Neighbors

K-Nearest Neighbor (KNN) is one of the most common machine learning algorithms. It is applied in image and video recognition, finance, political science, and healthcare. K-Nearest Neighbor is used in financial institutions to predict customers' credit ratings. It also indicates whether it is safe or risky to disburse loans to customers. In political science, KNN classifies voters into two categories: those who will vote and those who will not (Navlani). K Nearest Neighbor is based on feature similarity. This algorithm is used in both regression and classification problems.

The K-Nearest Neighbors algorithm is non-parametric implying that the model structure is determined from the dataset. KNN has no assumptions for underlying data distribution (Navlani). This attribute is helpful as most real-world datasets have no mathematical theoretical assumptions. Also, KNN is a lazy learning algorithm (Harrison). This algorithm does not need any training data points to generate a model. All training data is used in the testing phase, making it slower than the training phase (Navlani). The testing phase is also costly in terms of memory and time consumption.

In K-Nearest Neighbors, K is the number of nearest neighbors. KNN calculates distance, finds closest neighbors, and votes for labels. These steps are demonstrated in the figure below.



The value of K (number of neighbors) is chosen during model building. It is a controlling variable for the prediction model. Various datasets have different requirements. Therefore, selecting an optimal K value varies across different datasets. However, it is essential to note that choosing a small value of K , means that the noise will have a higher influence on the result than selecting a large K value. However, a higher value of K is computationally expensive. Generally, an odd number of K is chosen if the number of classes is even. The images below show the KNN classifier building using Scikit-learn.

	WTT	PTI	EQW	SBI	LQE	QWG	FDJ	PJF	HQE	NXJ	TARGET CLASS
0	0.913917	1.162073	0.567946	0.755464	0.780862	0.352608	0.759697	0.643798	0.879422	1.231409	1
1	0.635632	1.003722	0.535342	0.825645	0.924109	0.648450	0.675334	1.013546	0.621552	1.492702	0
2	0.721360	1.201493	0.921990	0.855595	1.526629	0.720781	1.626351	1.154483	0.957877	1.285597	0
3	1.234204	1.386726	0.653046	0.825624	1.142504	0.875128	1.409708	1.380003	1.522692	1.153093	1
4	1.279491	0.949750	0.627280	0.668976	1.232537	0.703727	1.115596	0.646691	1.463812	1.419167	1

This example uses KNN to create a model that predicts a class for a new data point based on the features. This is what our dataset looks like.

We will start with $K=1$

```

In [ ]: from sklearn.neighbors import KNeighborsClassifier
In [ ]: knn = KNeighborsClassifier(n_neighbors=1)
In [ ]: knn.fit(X_train,y_train)
Out[ ]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
metric_params=None, n_jobs=1, n_neighbors=1, p=2,
weights='uniform')
In [ ]: pred = knn.predict(X_test)

```

Predictions and evaluations

```
from sklearn.metrics import classification_report,confusion_matrix
```

```
print(confusion_matrix(y_test,pred))
```

```
[[125  18]
 [ 13 144]]
```

```
print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.91	0.87	0.89	143
1	0.89	0.92	0.90	157
avg / total	0.90	0.90	0.90	300

Using the elbow method to choose a suitable K value

```

In [ ]: error_rate = []
In [ ]: for i in range(1,40):
In [ ]:     knn = KNeighborsClassifier(n_neighbors=i)
In [ ]:     knn.fit(X_train,y_train)
In [ ]:     pred_i = knn.predict(X_test)
In [ ]:     error_rate.append(np.mean(pred_i != y_test))

```

Retraining the model with $K=23$ has a lower error rate. The accuracy is higher (95%) than when $K=1$ (90% accuracy).

Works Cited

Harrison, Onel. "Machine Learning Basics With The K-Nearest Neighbors Algorithm." *Tds*, 10 Sept.

2018,

[towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a](https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761)

[6e71d01761](https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761). Accessed 10 Apr. 2022

Navlani, Avinash. "KNN Classification Tutorial Using Scikit-Learn." *datacamp*, 2 Aug. 2018,

www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn.

Accessed 10 Apr. 2022.

Academic Experts

**Your paper can be even better than this one.
Get help from real experts in academic writing.**

**REQUEST
HELP**

**GET A FREE
QUOTE**