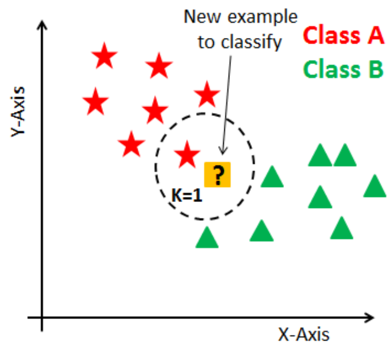
Predicting Search Volume in Excel using KNN

Understand KNN models with different combinations of hyperparameters



Overview

In this post, to predict time series data, I built KNN models with different combinations of parameters and two distance metrics in Excel and evaluated models' performance using RMSE.

Introduction

K-Nearest Neighbor (KNN) is a supervised learning technique. By using this method, we can classify or predict a new data point, based on similarity. To be more specific, predictions are made by searching through the entire training set for the k most similar cases (neighbors), and summarizing the output variables, which can be labels or numbers, for those k cases.

This is one of the assignments I finished in the course "Predictive Analytics" in my fourth quarter in graduate school. Instead of using a programming language, I chose to use Excel as it is one of the most widely used tools for data analysis.

Business Problem

First, I downloaded search data of the search term "Predictive Analytics" over the past 5 years from Google Trends. Then, to predict the value for the latest week (Jan 10, 2021 – Jan 16, 2021), I built KNN models with different combinations of parameters and distance metrics. Finally, to find the best models, Root Mean Squared Error (RMSE) was used to calculate accuracy.

	A	B
1	Date	Trends
2	2016-01-31	66
3	2016-02-07	56
4	2016-02-14	63
5	2016-02-21	60
6	2016-02-28	42
7	2016-03-06	70
8	2016-03-13	25
9	2016-03-20	43
10	2016-03-27	58

	A	B
251	2020-11-08	44
252	2020-11-15	43
253	2020-11-22	41
254	2020-11-29	53
255	2020-12-06	66
256	2020-12-13	37
257	2020-12-20	36
258	2020-12-27	28
259	2021-01-03	37
260	2021-01-10	41

Downloaded Data from Google Trends

Data Modeling

When building a KNN model, we can alter dimensionality of the model, distance function, or the number of nearest neighbors to find the best result.

1. Dimensionality of the model (n):

I used n=2 and 3 to see whether a higher dimensional model could yield a more accurate prediction. The models I used have the following form, where n+1 is the dimensionality of the model.

$$x_{i+1} = m(x_i, x_{i-1}, \ldots, x_{i-n})$$

Thus, since n=2, and 3, three-dimensional and four-dimensional models were built. In the three-dimensional model, I had three inputs, while in the four-dimensional model, I had four inputs.

1	n=2 (Three dimension)			
2	X1	X2	X3	Output
3	66	56	63	60
4	56	63	60	42

....

257	37	36	28	37
258	36	28	37	?

1	n=3 (Four dimension)				
2	X1	X2	X3	X4	Output
3	66	56	63	60	42
4	56	63	60	42	70

...

256	66	37	36	28	37
257	37	36	28	37	?

2. Distance function:

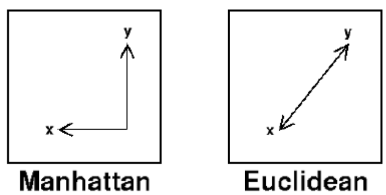
Another element that can affect the performance of the models is the distance function. Euclidean and Manhattan are two popular distance metrics.

Euclidean distance: the distance between two points is the length of a line segment between the two points.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan distance: the distance between two points is measured along axes at right angles. It is called the Manhattan distance since it is the distance a car would drive in a city (e.g., Manhattan.)

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$



For each dimensional model, I calculated both Euclidean and Manhattan distance.

Three-dimensional model

257	37	36	28	37
258	36	28	37	?

Euclidean distance

1	n=2 (Three dimension)				
2	X1	X2	X3	Output	Euclidean Distance
3	66	56	63	60	=SQRT((A3-A258)^2+(B3-B258)^2+(C3-C258)^2)
4	56	63	60	42	
5	63	60	42	70	
6	60	42	70	25	43.13930922

Manhattan distance

	A	B	C	D	E
1	n=2 (Three dimension)				
2	X1	X2	X3	Output	Manhattan Distance
3	66	56	63	60	=ABS(A3-A258)+ABS(B3-B258)+ABS(C3-C258)
4	56	63	60	42	
5	63	60	42	70	
6	60	42	70	25	71

Four-dimensional model

256	66	37	36	28	37
257	37	36	28	37	?

Euclidean distance

1	n=3 (Four dimension)					
2	X1	X2	X3	X4	Output	Euclidean Distance
3	66	56	63	60	42	=SQRT((A3-A257)^2+(B3-B257)^2+(C3-C257)^2+(D3-D257)^2)
4	56	63	60	42	70	
5	63	60	42	70	25	
6	60	42	70	25	43	
7	42	70	25	43	58	35.0142828

Manhattan distance

1	n=3 (Four dimension)					
2	X1	X2	X3	X4	Output	Manhattan Distance
3	66	56	63	60	42	=ABS(A3-A257)+ABS(B3-B257)+ABS(C3-C257)+ABS(D3-D257)
4	56	63	60	42	70	
5	63	60	42	70	25	
6	60	42	70	25	43	
7	42	70	25	43	58	48

3. The number of nearest neighbors (k):

I used k=1,3,5, and 7 to search through the dataset for the k most similar cases (neighbors). Normally, we will create error rates plot in such as R programming language to find the optimal k value. To use different k values, I first sorted distances and ranked observations. Then, I summarized the output for those k cases. For instance, if k=5, I will calculate the average of outputs from rank 1 to 5, since the outputs are numbers.

Three-dimensional model & Euclidean distance

1	n=2 (Three dimension)									
2	X1	X2	X3	Output	Euclidean Distance	Rank	k=1	k=3	k=5	k=7
3	39	26	30	38	7.874007874	1	38	40.33333333	50	51.42857143
4	26	30	38	46	10.24695077	2				
5	37	36	28	37	12.08304597	3				
6	46	36	39	70	12.9614814	4				
7	30	38	46	59	14.73091986	5				
8	46	36	46	53	15.65247584	6				
9	48	36	30	57	16.03121954	7				

Three-dimensional model & Manhattan distance

1	n=2 (Three dimension)									
2	X1	X2	X3	Output	Manhattan Distance	Rank	k=1	k=3	k=5	k=7
3	39	26	30	38	12	1	38	40.33333333	52.125	53
4	26	30	38	46	13	2				
5	37	36	28	37	18	3				
6	35	46	36	39	20	4				
7	46	36	39	70	20	4				
8	36	30	57	73	22	5				
9	35	43	43	73	22	5				
10	36	44	43	41	22	5				
11	53	34	36	54	24	6				
12	30	38	46	59	25	7				

Four-dimensional model & Euclidean distance

1	n=3 (Four dimension)										
2	X1	X2	X3	X4	Output	Euclidean Distance	Rank	k=1	k=3	k=5	k=7
3	39	26	30	38	46	10.44030651	1	46	56.33333333	56.4	57.28571429
4	35	46	36	39	70	13.11487705	2				
5	36	44	43	41	53	17.49285568	3				
6	40	53	34	36	54	18.30300522	4				
7	26	30	38	46	59	18.38477631	5				
8	49	46	36	46	53	19.72308292	6				
9	44	43	41	53	66	22.86919325	7				

Four-dimensional model & Manhattan distance

1	n=3 (Four dimension)										
2	X1	X2	X3	X4	Output	Manhattan Distance	Rank	k=1	k=3	k=5	k=7
3	39	26	30	38	46	15	1	46	56.66666667	59.2	58
4	35	46	36	39	70	22	2				
5	40	53	34	36	54	27	3				
6	36	44	43	41	53	28	4				
7	48	36	30	57	73	33	5				
8	26	30	38	46	59	36	6				
9	34	36	54	30	56	36	6				
10	49	46	36	46	53	39	7				

Data Analysis

There was a total of 16 models (2x2x4=16) and the prediction results are shown in the table below. From the table, we can tell that since the real value of the latest week is **41**, models with three-dimensional, Euclidean distance or Manhattan distance, and k=3 are the most accurate. Besides, from the table below, we can tell that overall, three-dimensional models seem to perform better than four-dimensional models, since the predicted results of three-dimensional models are closer to 41. Also, from my Excel work, I found that the Manhattan distance is more likely to have more than one observation in a rank.

Real Value=41	K=1	K=3	K=5	K=7
Three-dimensional & Euclidean	38	40.333	50	51.43
Three-dimensional & Manhattan	38	40.333	52.125	53
Four-dimensional & Euclidean	46	56.333	56.4	57.286
Four-dimensional & Manhattan	46	56.667	59.2	58

Model Accuracy

To report the accuracy of the models, I used the real values for the latest month and the predicted values for the latest month (Dec 20, 2020 – Jan 16, 2021) to calculate Root Mean Squared Error (RMSE.)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

Three-dimensional model & Euclidean distance

k=1	Error	Error^2		k=3	Error	Error^2
2020/12/20	12.5	156.25		2020/12/20	14.8	219.04
2020/12/27	42	1764		2020/12/27	10.66666667	113.7777778
2021/1/3	1	1		2021/1/3	7.66666667	58.77777783
2021/1/10	3	9		2021/1/10	0.66666667	0.444444449
RMSE		21.96730525		**RMSE**		9.900000002

k=5	Error	Error^2		k=7	Error	Error^2
2020/12/20	19.42857143	377.4693878		2020/12/20	20	400
2020/12/27	15.4	237.16		2020/12/27	13.71428571	188.0816325
2021/1/3	13	169		2021/1/3	16.71428571	279.3673468
2021/1/10	9	81		2021/1/10	10.42857143	108.7551021
RMSE		14.70229053		**RMSE**		15.62213239

Three-dimensional model & Manhattan distance

k=1	Error	Error^2		k=3	Error	Error^2
2020/12/20	12	144		2020/12/20	13	169
2020/12/27	8	64		2020/12/27	12	144
2021/1/3	20	400		2021/1/3	14	196
2021/1/10	3	9		2021/1/10	0.666666667	0.44444444
RMSE		12.41974235		**RMSE**		11.285438

k=5	Error	Error^2		k=7	Error	Error^2
2020/12/20	15.83333333	250.6944444		2020/12/20	18	324
2020/12/27	15.57142857	242.4693878		2020/12/27	17.09090909	292.099174
2021/1/3	16.42857143	269.8979592		2021/1/3	14.92307692	222.698225
2021/1/10	11.125	123.765625		2021/1/10	12	144
RMSE		14.88982384		**RMSE**		15.6747998

Four-dimensional model & Euclidean distance

k=1	Error	Error^2		k=3	Error	Error^2
2020/12/20	12	144		2020/12/20	15.25	232.5625
2020/12/27	2	4		2020/12/27	5.666666667	32.1111111
2021/1/3	20	400		2021/1/3	10.33333333	106.777778
2021/1/10	5	25		2021/1/10	15.33333333	235.111111
RMSE		11.9687092		**RMSE**		12.3142448

k=5	Error	Error^2		k=7	Error	Error^2
2020/12/20	23.83333333	568.0277778		2020/12/20	20.25	410.0625
2020/12/27	11.8	139.24		2020/12/27	15.14285714	229.306122
2021/1/3	7.2	51.84		2021/1/3	6.428571429	41.3265306
2021/1/10	15.4	237.16		2021/1/10	16.28571429	265.22449
RMSE		15.78185491		**RMSE**		15.3779033

Four-dimensional model & Manhattan distance

k=1	Error	Error^2		k=3	Error	Error^2
2020/12/20	14	196		2020/12/20	17	289
2020/12/27	9.5	90.25		2020/12/27	13	169
2021/1/3	20	400		2021/1/3	7.666666667	58.7777778
2021/1/10	5	25		2021/1/10	15.66666667	245.444444
RMSE		13.33463535		RMSE		13.8041862
k=5	Error	Error^2		k=7	Error	Error^2
2020/12/20	19.4	376.36		2020/12/20	22.28571429	496.653061
2020/12/27	14.71428571	216.5102041		2020/12/27	15.2	231.04
2021/1/3	7.2	51.84		2021/1/3	6.428571429	41.3265306
2021/1/10	18.2	331.24		2021/1/10	17	289
RMSE		15.62010086		RMSE		16.2636065

RMSE	K=1	K=3	K=5	K=7
Three-dimensional & Euclidean	21.96730525	9.900000002	14.70229053	15.62213239
Three-dimensional & Manhattan	12.41974235	11.285438	14.88982384	15.6747998
Four-dimensional & Euclidean	11.9687092	12.3142448	15.78185491	15.3779033
Four-dimensional & Manhattan	13.33463535	13.80411862	15.62010086	16.2636065

The RMSE is the square root of the variance of the residuals. The lower value of RMSE, the better fit of the model. Thus, from the table above, we can tell that the model with three-dimensional, Euclidean distance, and k=3 is the best, since it has the lowest RMSE. Also, the second-best model is three-dimensional, Manhattan distance, and k=3. Overall, the conclusions are the same as those I made when predicting the value for the latest week (Jan 10, 2021 – Jan 16, 2021).

Conclusion

Normally, Excel is not a good tool for serious data analysis and does not scale to process large datasets. Yet, in this case, Excel acted as a user-friendly tool since no programming skill is required to build a machine learning model. Also, by thinking step by step, we can clearly understand the concept behind a KNN model, making it accessible for even non-programmers to apply this algorithm in real-world tasks.

See/ Download Excel file: KNN in Excel_Kuan-Pei (Yuki) Lai.xlsx

References

Bhalla, D. *K NEAREST NEIGHBOR : STEP BY STEP TUTORIAL*. Retrieved from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

Fiori, L. (2020, May 22). *Distance metrics and K-Nearest Neighbor (KNN).* Retrieved from https://medium.com/@luigi.fiori.lf0303/distance-metrics-and-k-nearest-neighbor-knn-1b840969c0f4

Luy, O. A. (2020, June 28). *Machine Learning Made Simple With Excel.* Retrieved from https://towardsdatascience.com/machine-learning-made-simple-with-excel-1bffd7901502

Manhattan. Retrieved from www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Manhattan_Distance_Metric.htm

Martin, K.G. *Assessing the Fit of Regression Models.* Retrieved from https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/

Root-Mean-Square Error in R Programming. Retrieved from https://www.geeksforgeeks.org/root-mean-square-error-in-r-programming/

Teknomo, K. *KNN for Smoothing and Prediction.* Retrieved from https://people.revoledu.com/kardi/tutorial/KNN/KNN_TimeSeries.htm