AI200: APPLIED MACHINE LEARNING

K-NEAREST NEIGHBOUR (K-NN)

OVERVIEW & LITERATURE OF MACHINE LEARNING

- Machine Learning
  - Supervised Learning
    - Regression
    - Classification
  - Unsupervised Learning
    - Clustering
    - Association
    - Dimensionality Reduction
  - Reinforcement Learning

1. Spectrum of supervision	Supervised Learning		Unsupervised Learning			Reinforcement Learning
2. Types of problems	Regression	Classification	Clustering	Association	Dimensionality Reduction	
3. Types of algorithm / models	Linear regression Polynomial regression Neural Networks Decision Tree Random Forest XGBoost	Logistic Regression KNN Naïve Bayes SVM Neural Networks Decision Tree Random Forest XGBoost	Neural Networks K-Means Hierarchical Clustering Hidden Markov Models Gaussian Mixture	Apriori Algorithm FP-Growth Algorithm Eclat Algorithm	Neural Networks SVD PCA	
4. Types of applications	Forecasting Predictions Process optimization New Insights	Fraud Detection Image Classification Customer Retention Diagnostics	Recommendations Targeted Marketing Customer Segmentation	Market Basket Analysis	Feature Elicitation Structure Discovery Meaningful Compression Big Data Visualisation	Real-time Decisions Game AI Learning Tasks Skill Acquisition Robot Navigation

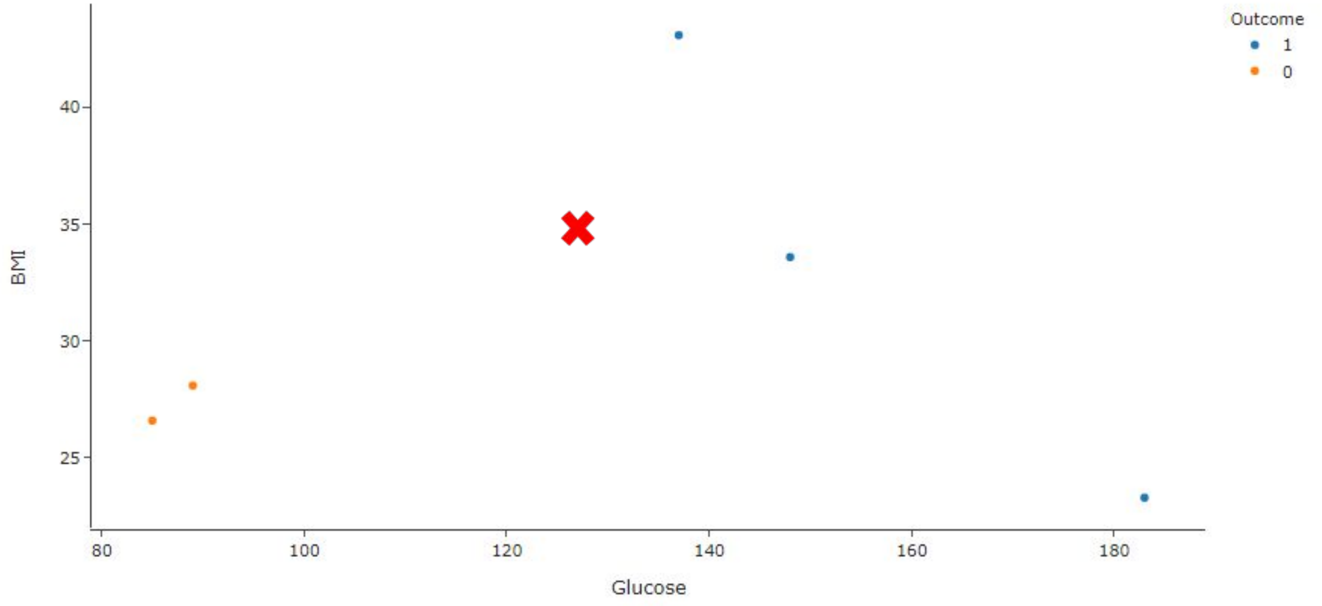
WHAT IS K-NN: LAYMAN INTUITION

- The idea behind K-Nearest Neighbours (K-NN) is very simple:
  - We find K number of “neighbours” with similar characteristics, and assume the same outcome as those neighbours
  - Let’s use an example for better understanding:
    - Assuming we say we will assume the same outcome as the **3 closest neighbour**…
    - Now let’s imagine that we got the biometrics of a new person, and we want to predict his likelihood of having diabetes (**glucose=130, BMI=35**)

Features **Outcome**

Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
148	72	35	0	33.6	0.627	50	1
85	66	29	0	26.6	0.351	31	0
183	64	0	0	23.3	0.672	32	1
89	66	23	94	28.1	0.167	21	0
137	40	35	168	43.1	2.288	33	1

Plot features and outcome



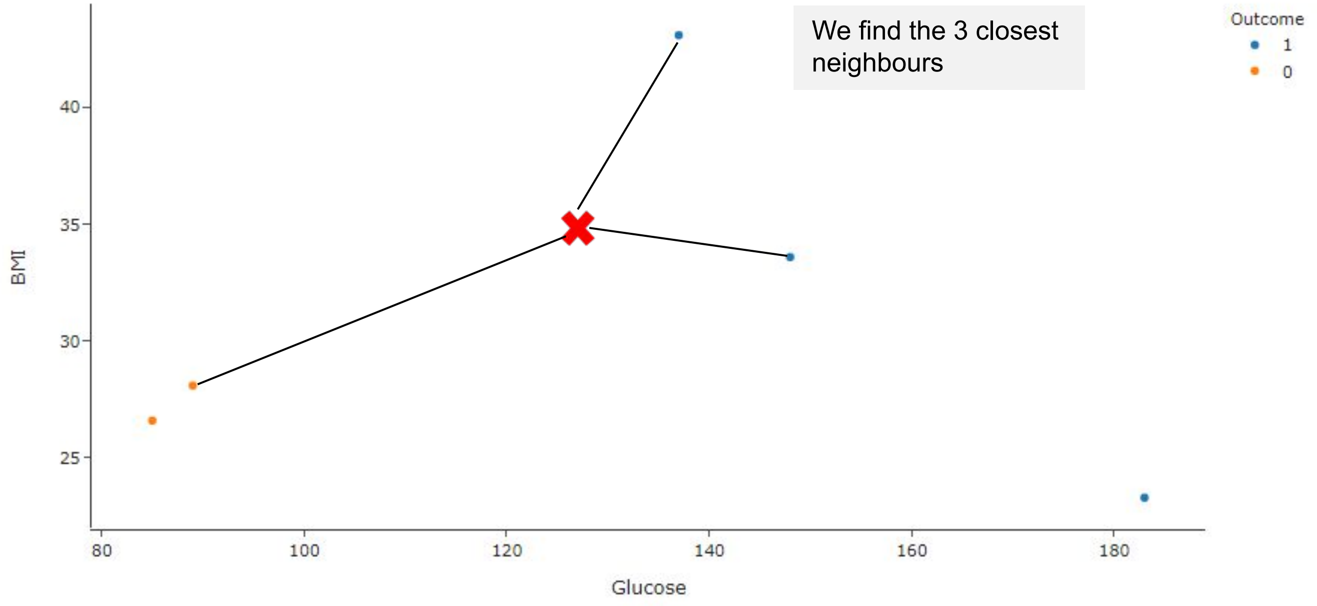
WHAT IS K-NN: LAYMAN INTUITION

- The idea behind K-Nearest Neighbours (K-NN) is very simple:
  - We find K number of “neighbours” with similar characteristics, and assume the same outcome as those neighbours
  - Let’s use an example for better understanding:
    - We will find the **3 closest neighbours in terms of features similarity** to the new person
    - Thereafter, we will average the outcome of the 3 neighbours and assume that as the outcome for the new person
    - **2 out of 3 of the neighbours had diabetes**, and if we assume that outcome; the **person has 2/3 probability** of contracting diabetes. Given that this person’s probability of having diabetes is higher than 0.5, we classify him as having diabetes

Features **Outcome**

Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
148	72	35	0	33.6	0.627	50	1
85	66	29	0	26.6	0.351	31	0
183	64	0	0	23.3	0.672	32	1
89	66	23	94	28.1	0.167	21	0
137	40	35	168	43.1	2.288	33	1

Plot features and outcome



Average outcome: (1 + 0 + 1) / 3 = 2/3

The spirit of this model is: Birds of the same feathers flock together (e.g. people of certain profiles tend to flock to certain schools)

WHY SHOULD YOU LEARN KNN?

- KNN is a very simple and intuitive model

- But despite its simplicity, KNN is very powerful and well-used in various commercial settings:
    - Amazon’s recommended products
    - Netflix recommended movies
    - Spotify recommended music

- In session 8, we will learn how to extend K-NN to build a recommender system that can recommend products for cross-selling

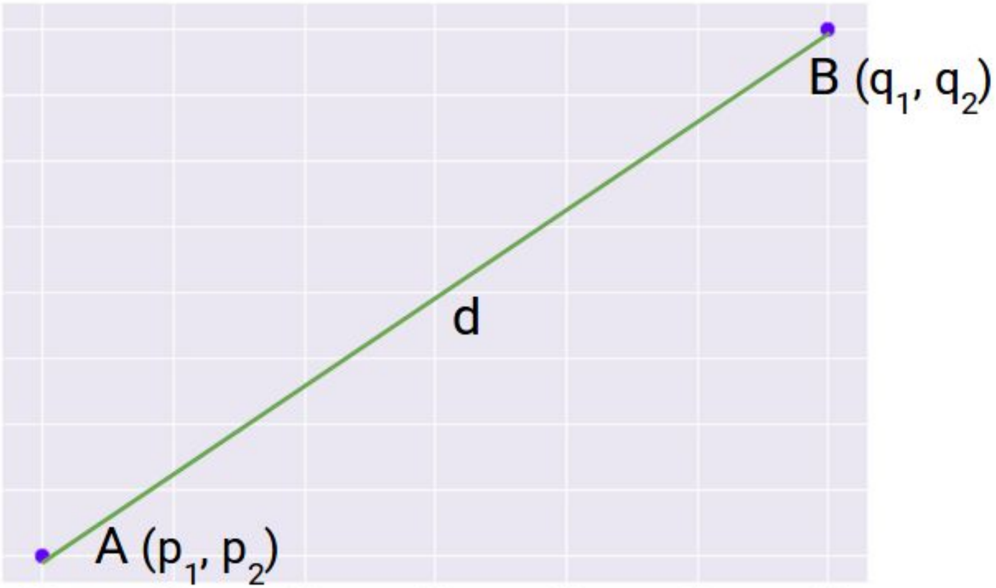
K-NEAREST NEIGHBOURS (K-NN)

MECHANISM BEHIND MODEL

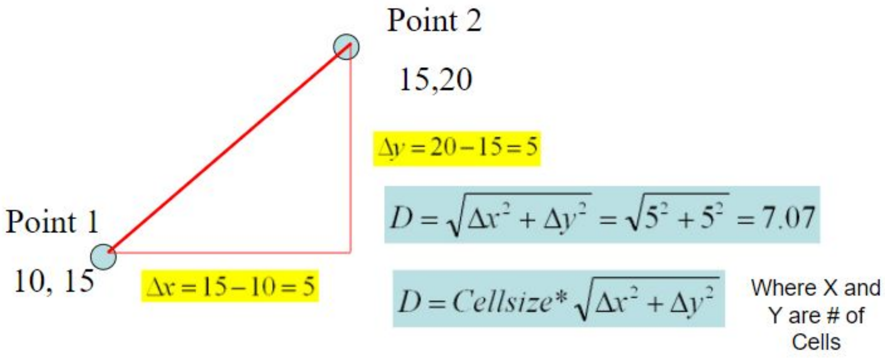
MECHANISM BEHIND MODEL: MEASURING EUCLIDEAN DISTANCE

- **Step 1**: When we are trying to predict the outcome of a new datapoint, we calculate its distance with other labelled data points using:
  - Euclidean Distance (if you remember your arithmetic class in secondary school)

$$d(p, q) = \sqrt{(p_1 - q_1)2 + (p_2 - q_2)2}$$



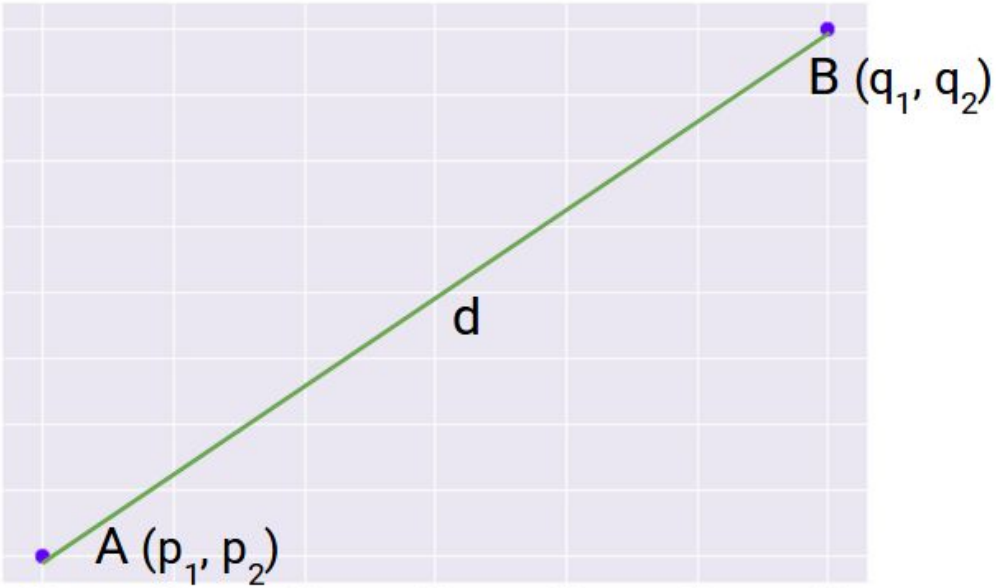
Example of Calculation of Euclidean Distance



MECHANISM BEHIND MODEL: MEASURING EUCLIDEAN DISTANCE

- **Step 1**: When we are trying to predict the outcome of a new datapoint, we calculate its distance with other labelled data points using:
  - Euclidean Distance (if you remember your arithmetic class in secondary school)

$$\boxed{d(p,q) = \sqrt{(p_1 - q_1)2 + (p_2 - q_2)2}}$$



Euclidean distance is only good for 2 dimensions (when you have 2 columns of data), so we can generalize this formula into one which can deal with multi-dimensions

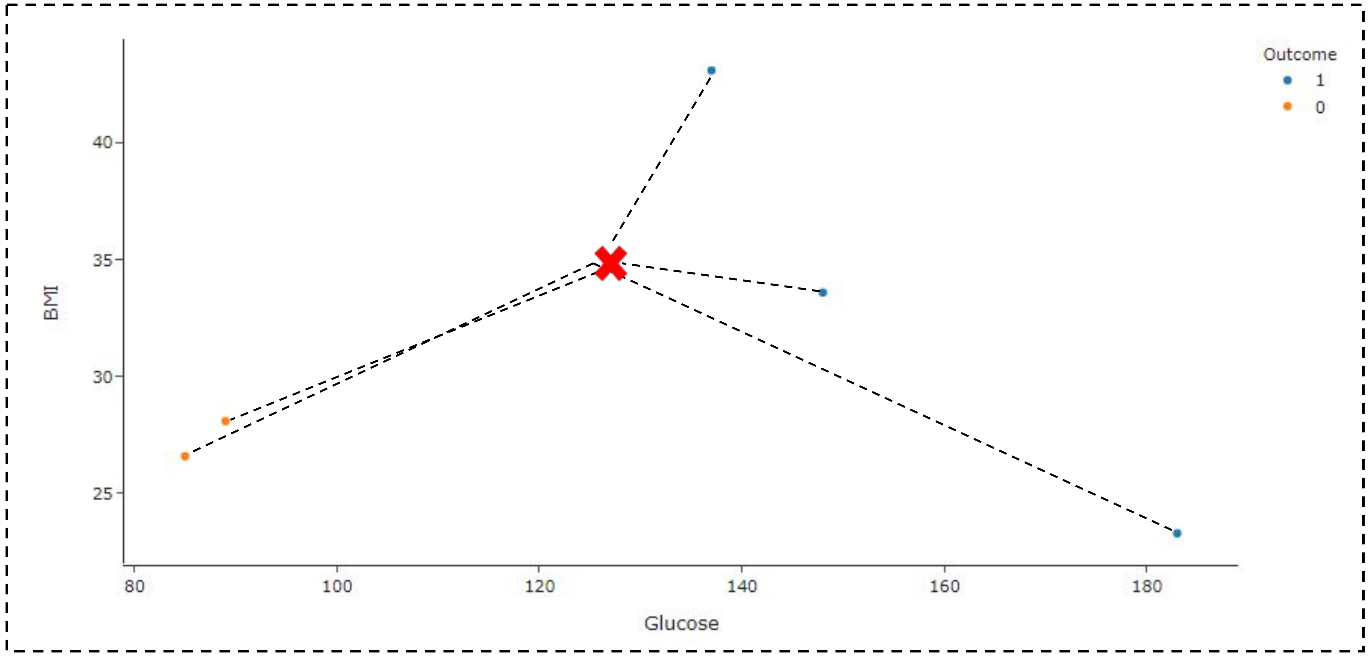
Minkowski Distance

$$d(i,j) = \sqrt[q]{\underbrace{|x_{i1} - x_{j1}|^q}_{1^{st}\text{ dimension}} + \underbrace{|x_{i2} - x_{j2}|^q}_{2^{nd}\text{ dimension}} + \ldots + \underbrace{|x_{ip} - x_{jp}|^q}_{p^{th}\text{ dimension}}}$$

Don’t be frighten by the formulas, the mechanism is the same as Euclidean distance, and that is all you need to know.

MECHANISM BEHIND MODEL: MEASURING EUCLIDEAN DISTANCE

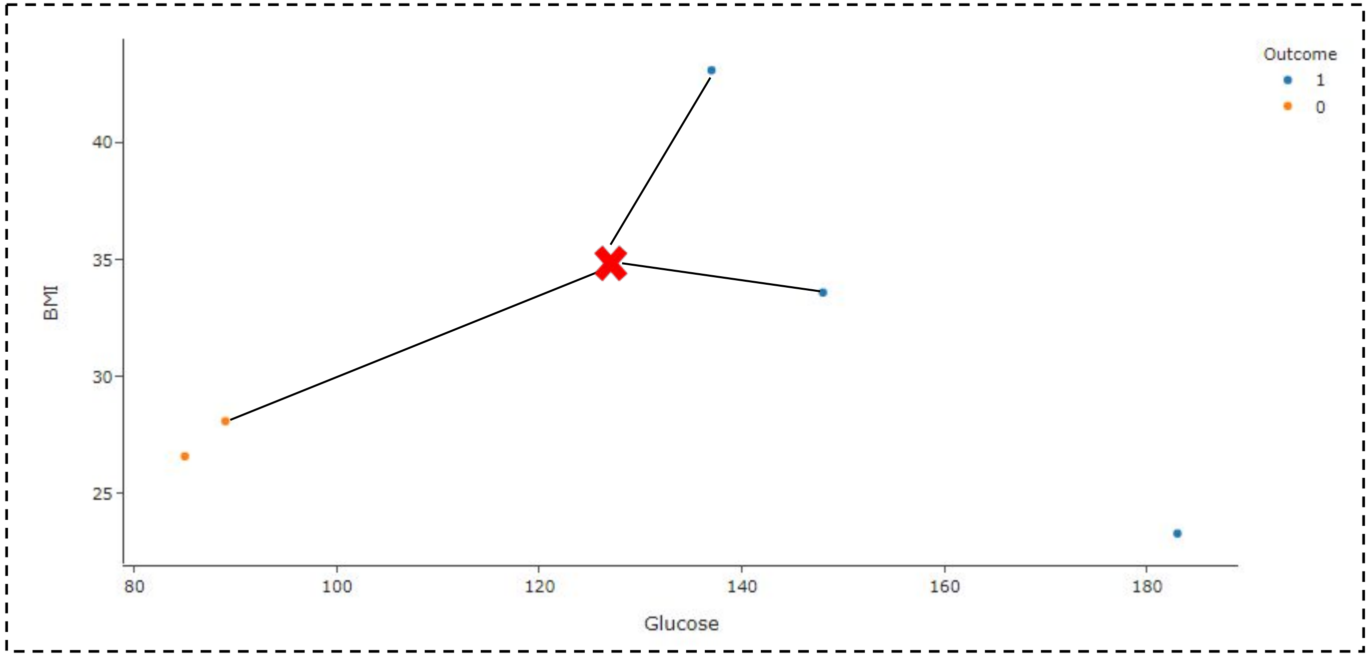
- **Step 1**: When we are trying to predict the outcome of a new datapoint, we calculate its distance with other labelled datapoints using:
  - Euclidean Distance (if you remember your arithmetic class in secondary school)
- **Step 2**: We measure the distance between the new datapoint and every other labelled datapoints using the Euclidean distance as well



Labelled Datapoint #	Labelled Datapoint Outcome	Distance from new datapoint
A	1	13
B	0	50.7
C	1	49.4
D	0	46.5
E	1	8.3

MECHANISM BEHIND MODEL: MEASURING EUCLIDEAN DISTANCE

- **Step 1**: When we are trying to predict the outcome of a new datapoint, we calculate its distance with other labelled data points using:
    - Euclidean Distance (if you remember your arithmetic class in secondary school)
- **Step 2**: We measure the distance between the new datapoint and every other labelled data points using the Euclidean distance as well
- **Step 3**: If we let k=3 (meaning we want to assume the outcome of the 3 closest neighbours), we will average the outcome of the 3 closest neighbours



Labelled Datapoint #	Labelled Datapoint Outcome	Distance from new datapoint
A	1	13
B	0	50.7
C	1	49.4
D	0	46.5
E	1	8.3

Average outcome: (1 + 0 + 1) / 3 = 2/3

MECHANISM BEHIND MODEL: SCALING

- But hang on, if you look at the diagram, you will realize that the range of values of BMI (23 – 43) and Glucose (89 – 183) is quite different
- This **difference in range will heavily skew distance-based models** such as K-NN, as it will heavily favor one feature over the others
- To resolve this, we should scale the features before even calculating the Euclidean distance. There are various methods to perform scaling, but here we use the **Standardization / Z-score Normalization** here

$$x_i' = \frac{x_i - \bar{x}}{\sigma}$$

Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
148	72	35	0	33.6	0.627	50	1
85	66	29	0	26.6	0.351	31	0
183	64	0	0	23.3	0.672	32	1
89	66	23	94	28.1	0.167	21	0
137	40	35	168	43.1	2.288	33	1

After scaling the data

Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0.528565	72	35	0	0.383768	0.627	50	1
-1.170394	66	29	0	-0.626149	0.351	31	0
1.472431	64	0	0	-1.102252	0.672	32	1
-1.062524	66	23	94	-0.409738	0.167	21	0
0.231921	40	35	168	1.754370	2.288	33	1

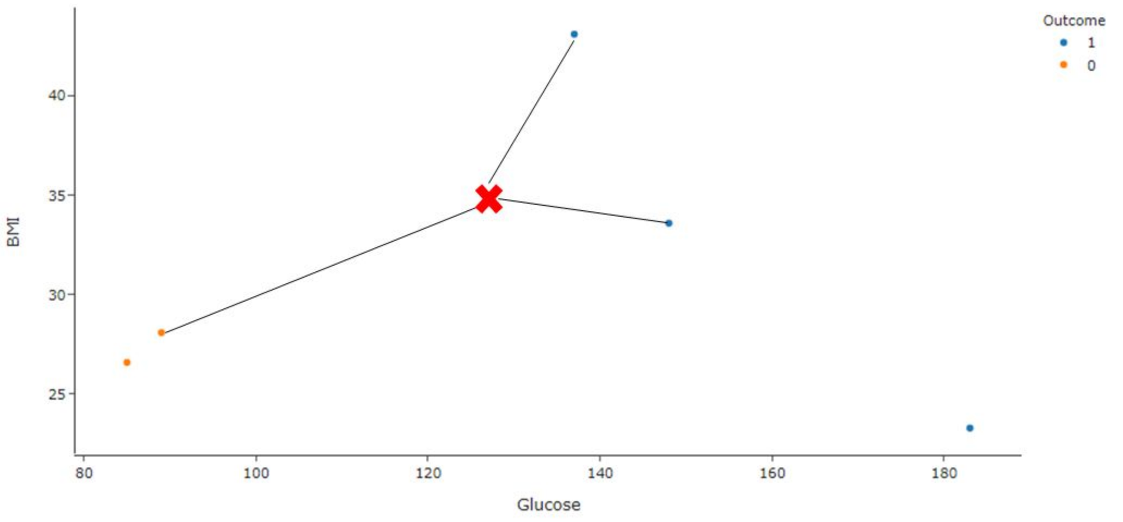
MECHANISM BEHIND MODEL: SCALING

- After scaling, you can observe that the 3 nearest neighbours changed!

Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0.528565	72	35	0	0.383768	0.627	50	1
-1.170394	66	29	0	-0.626149	0.351	31	0
1.472431	64	0	0	-1.102252	0.672	32	1
-1.062524	66	23	94	-0.409738	0.167	21	0
0.231921	40	35	168	1.754370	2.288	33	1

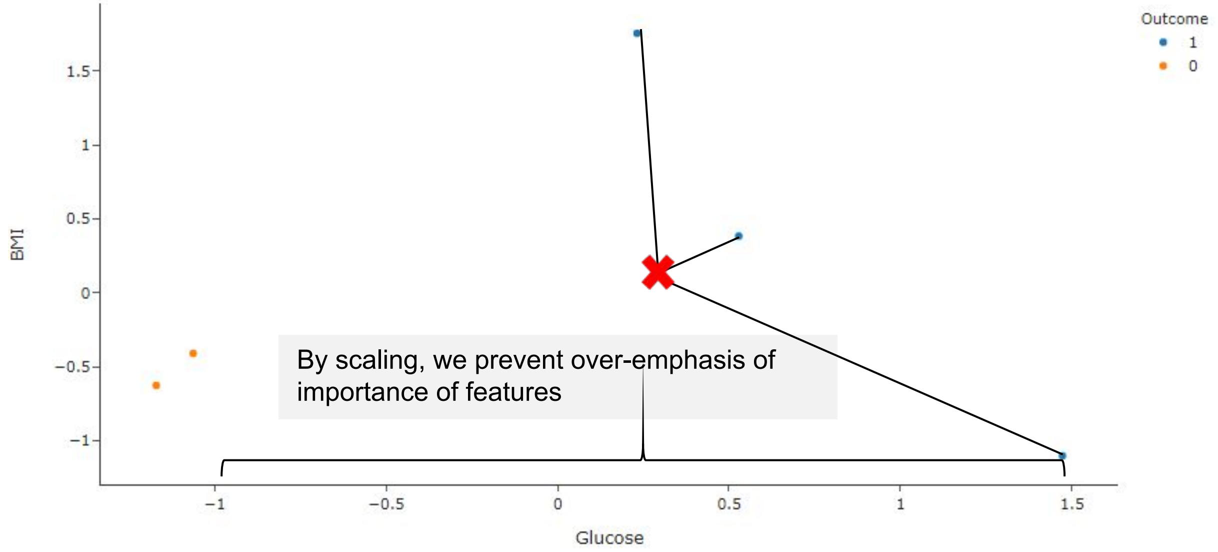
As you can see, after scaling the features, the nearest 3 neighbours all have an outcome of 1. So scaling your features will have an impact on the results and accuracy of your predictions.

BEFORE SCALING



Average outcome: (1 + 0 + 1) / 3 = 2/3

AFTER SCALING



Average outcome: (1 + 1 + 1) / 3 = 1

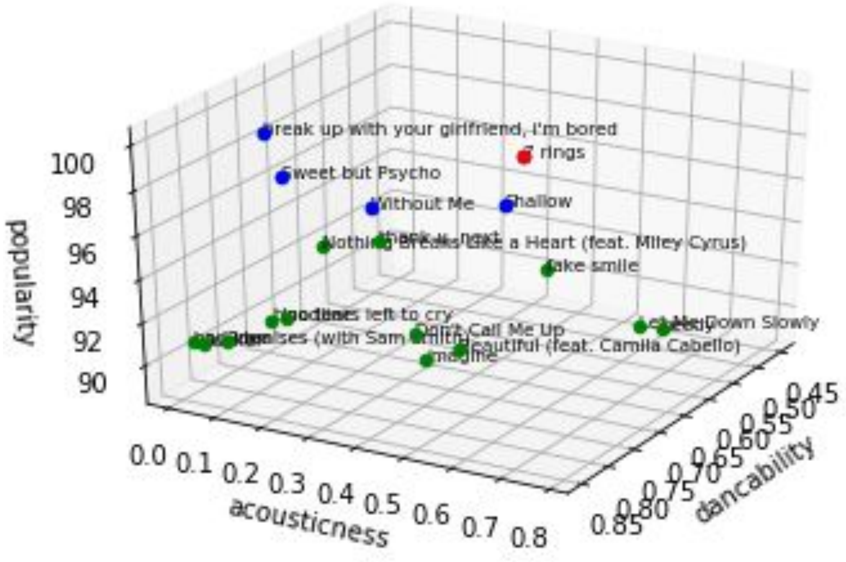
K-NEAREST NEIGHBOURS (K-NN)

K-NN WITH MULTIPLE FEATURES

K-NN WITH MULTIPLE FEATURES

- Even though we cannot visualize anything beyond 2-dimensions (the image below is an attempt at visualizing 3-dimensions), adding new features does not in anyway change the process which the Euclidean distance is calculated between each points

- During the code-implementation component, we will show how you can easily implement a K-NN model with more than 2 dimensions

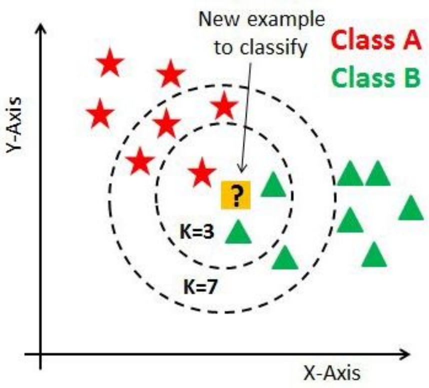


K-NEAREST NEIGHBOURS (K-NN)

STRATEGY FOR SELECTING THE K VALUE

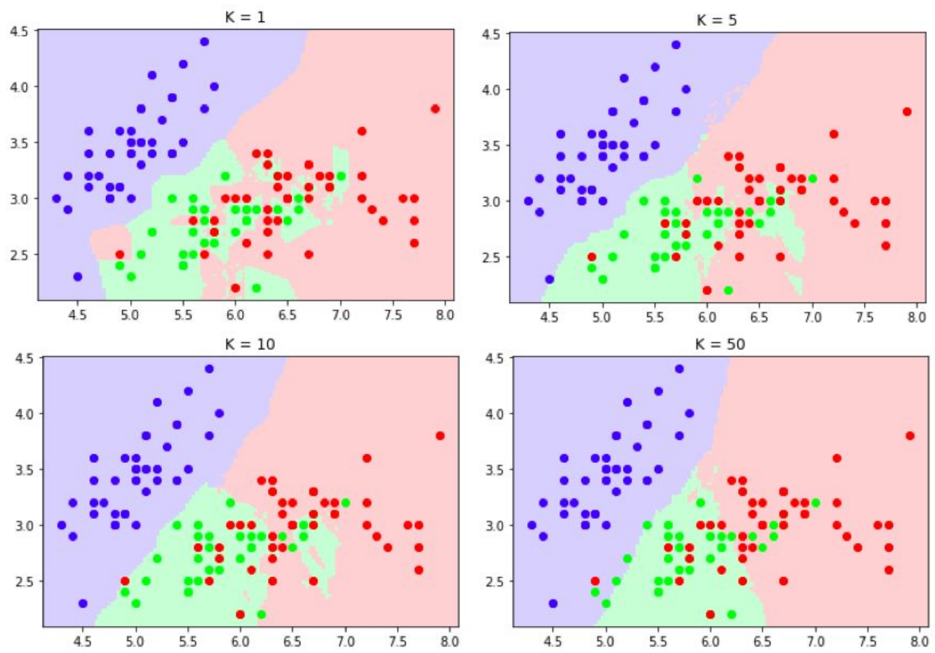
STRATEGY FOR SELECTING THE K VALUE: WHY IS THERE A NEED FOR THIS?

- The results of K-NN is greatly influenced by the K value that we choose



- In fact, the effect of a **small k vs large k** is well documented:
  - With a small k value, we have highly variable and unstable decision boundaries. Any small changes in the training dataset will lead to large changes in classification
  - With a large k value the decision boundary is smoother and more stable. There is less emphasis on individual data points, meaning that it would not be able to cater for outliers. In the most extreme case, the model becomes general to the point that everything is simply classified in the same category

Example where K-NN is used to classify 3 types of outcomes

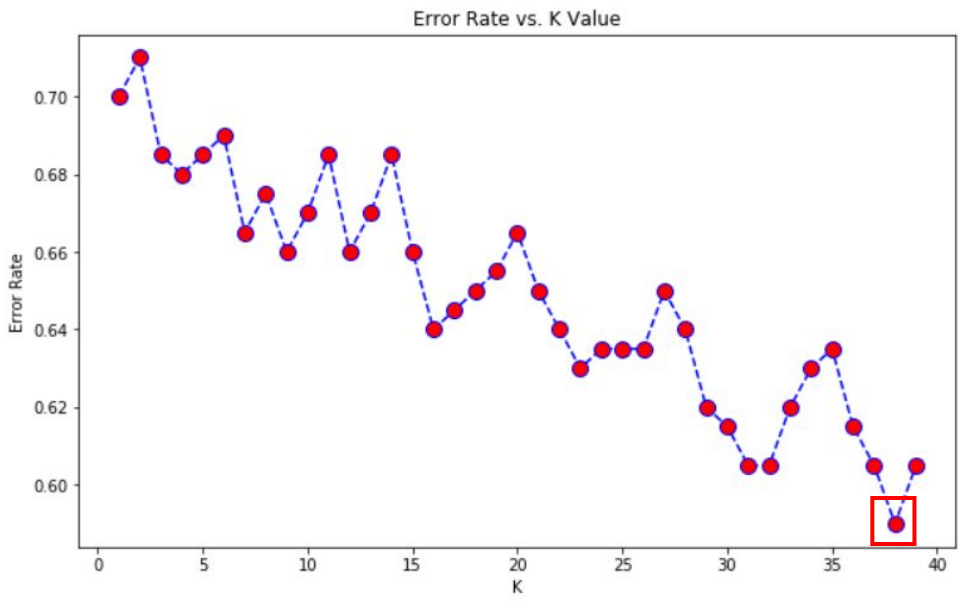


STRATEGY FOR SELECTING THE K VALUE: WHY IS THERE A NEED FOR THIS?

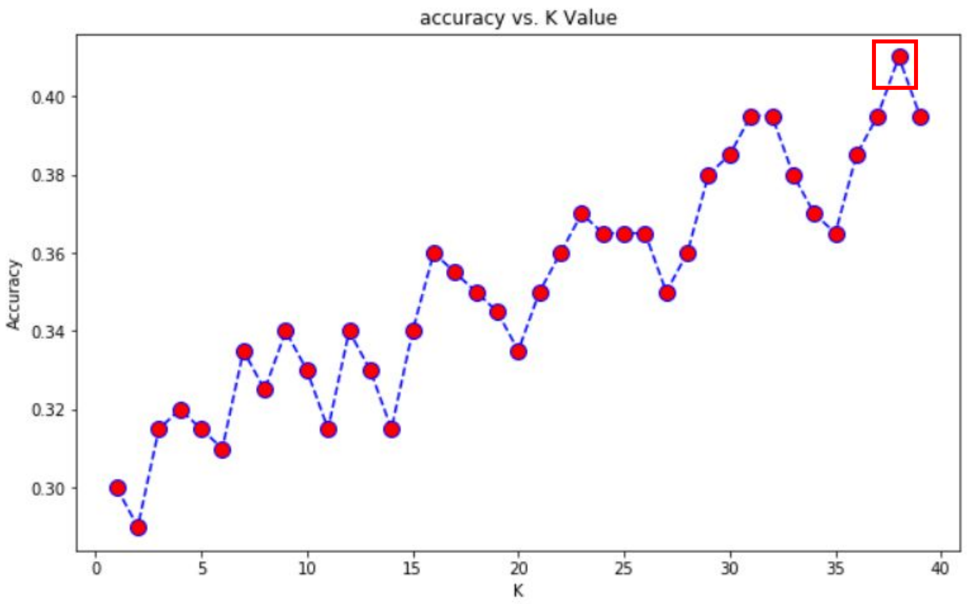
- To select an optimal K Value:

  - Initialize a random K value and start computing
  - Choosing a small value of K leads to unstable decision boundaries
  - The substantial K value is better for classification as it leads to smoothening the decision boundaries.
  - **Derive a plot between error rate and K denoting values in a defined range. Then choose the K value as having a minimum error rate**

Minimum error:- 0.59 at K = 37



Maximum accuracy:- 0.41 at K = 37



In the example above, both plots shows that k=37 is the optimal k value for the dataset

K-NEAREST NEIGHBOURS (K-NN)

ADDITIONAL CONSIDERATIONS

ADDITIONAL CONSIDERATIONS

- **Can be used for both classification and regression problems**: You should know that K-NN can be used for both classification and regression problems. The mechanism (finding the k neighbors with the nearest Euclidean distance to each other). Except that in the case of regression, the predicted outcome would be the mean values of the k nearest neighbors)