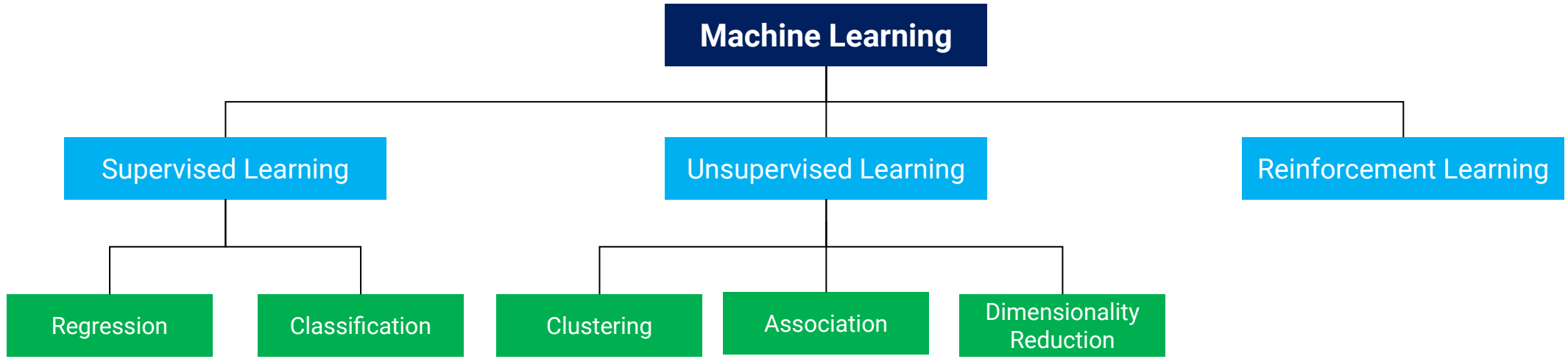
AI200: APPLIED MACHINE LEARNING

K-NEAREST NEIGHBOUR (K-NN)

OVERVIEW & LITERATURE OF MACHINE LEARNING



1. Spectrum of supervision

2. Types of problems

3. Types of algorithm / models

Regression	Classification	Clustering	Association	Dimensionality Reduction
Linear regression	Logistic Regression	Neural Networks	Apriori Algorithm	Neural Networks
Polynomial regression	KNN	K-Means	FP-Growth Algorithm	SVD
Neural Networks	Naïve Bayes	Hierarchical Clustering	Eclat Algorithm	PCA
Decision Tree	SVM	Hidden Markov Models		
Random Forest	Neural Networks	Gaussian Mixture		
XGBoost	Decision Tree			
	Random Forest			
	XGBoost			

4. Types of applications

Forecasting	Fraud Detection	Recommendations	Market Basket Analysis	Feature Elicitation	Real-time Decisions
Predictions	Image Classification	Targeted Marketing		Structure Discovery	Game AI
Process optimization	Customer Retention	Customer Segmentation		Meaningful Compression	Learning Tasks
New Insights	Diagnostics			Big Data Visualisation	Skill Acquisition
					Robot Navigation

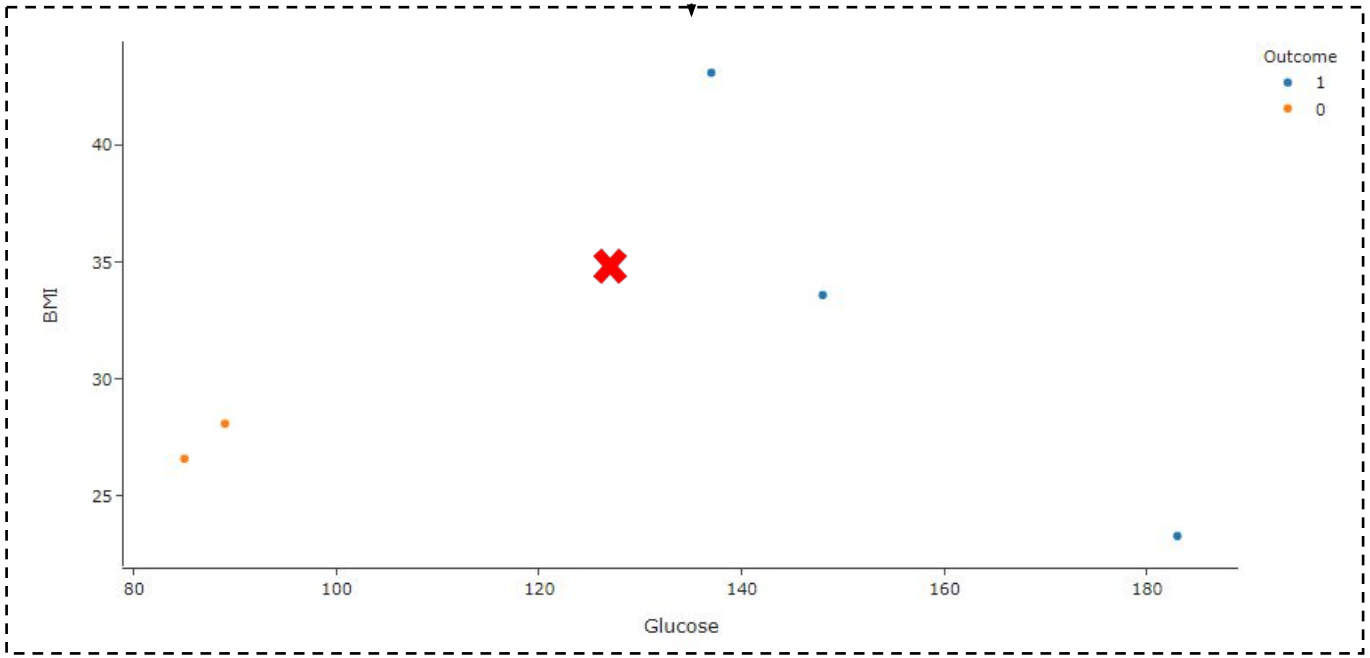
WHAT IS K-NN: LAYMAN INTUITION

- 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:
    - Say we will assume the same outcome as **three closest neighbours** (K = 3)
    - And imagine that with the health metrics 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



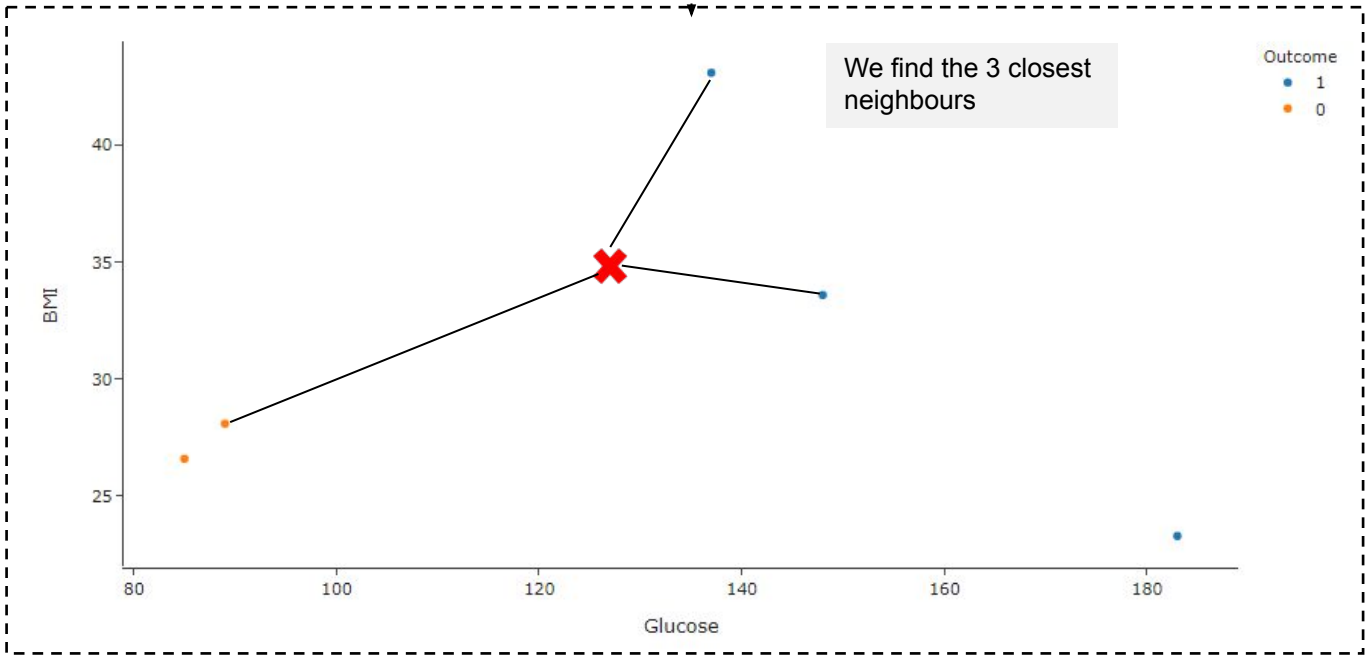
WHAT IS K-NN: LAYMAN INTUITION

- 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 feature similarity** to the point.
    - **Average the outcome of the 3 neighbours** to derive 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)

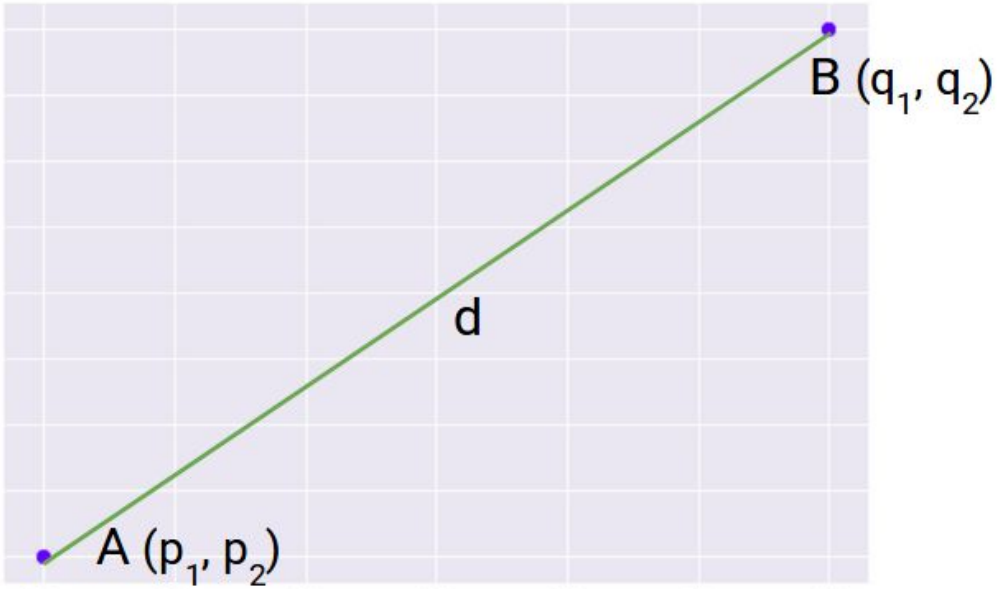
K-NEAREST NEIGHBOURS (K-NN)

MECHANISM BEHIND MODEL

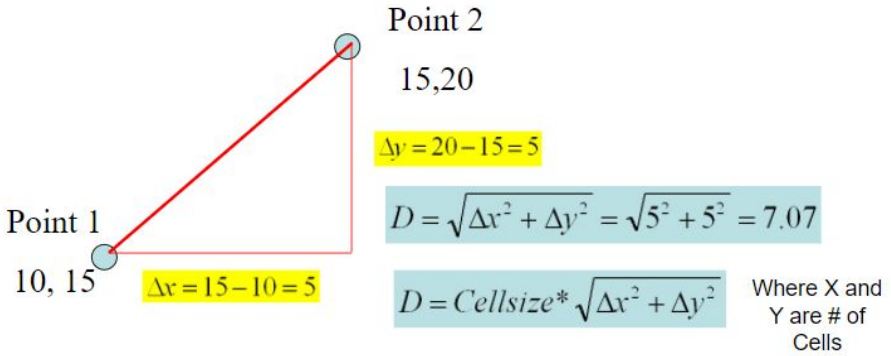
MECHANISM BEHIND MODEL: MEASURING EUCLIDEAN DISTANCE

- **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 *(remember what you've learnt back in secondary school?)*

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



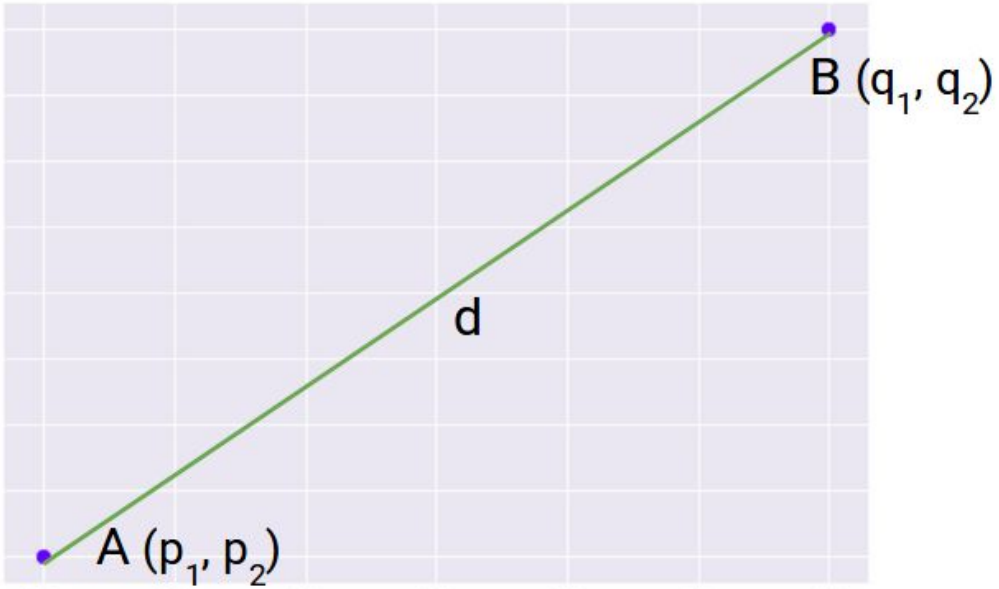
Example of Calculation of Euclidean Distance



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 *(remember what you've learnt back in secondary school?)*

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



Euclidean distance is only good for 2 dimensions (i.e. 2 columns of data), so we generalize this formula to deal with more than 2 dimensions

Minkowski Distance

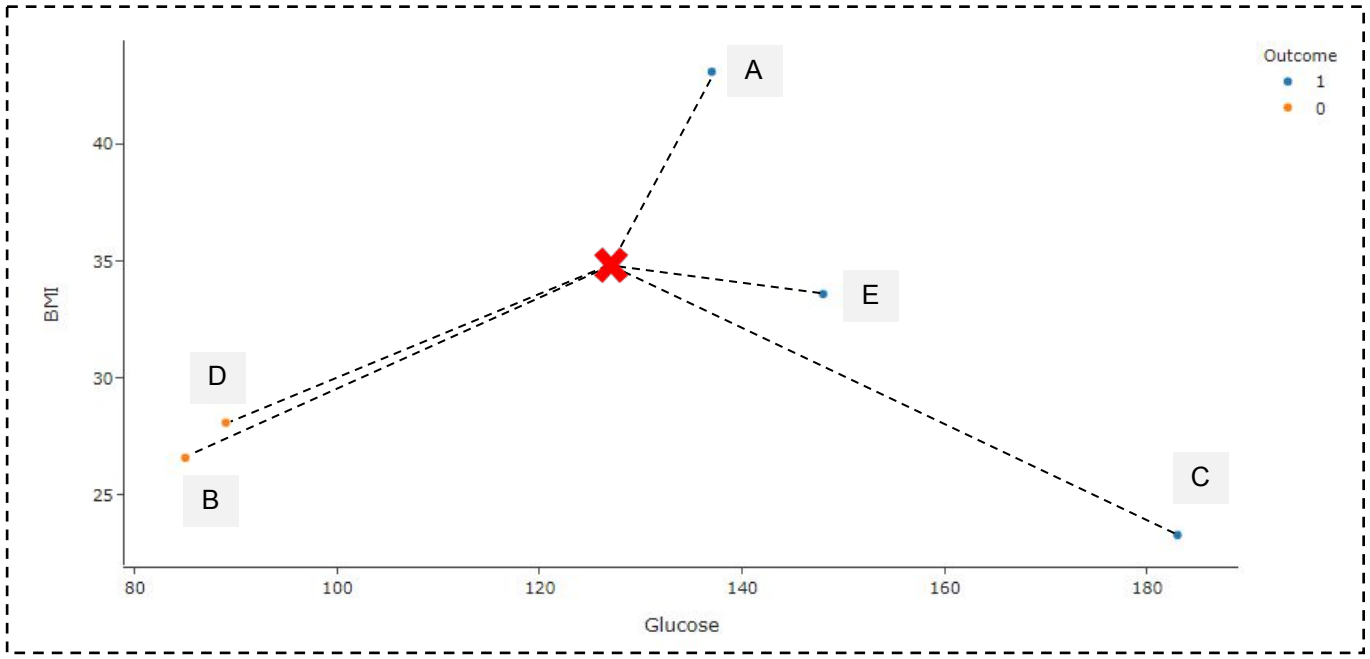
$$d(i,j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q}$$

1st dimension 2nd dimension pth dimension

Do not be frightened by this formula!
The principle behind it is the same as the Euclidean distance, except extended to more dimensions.

MECHANISM BEHIND MODEL: MEASURING EUCLIDEAN DISTANCE

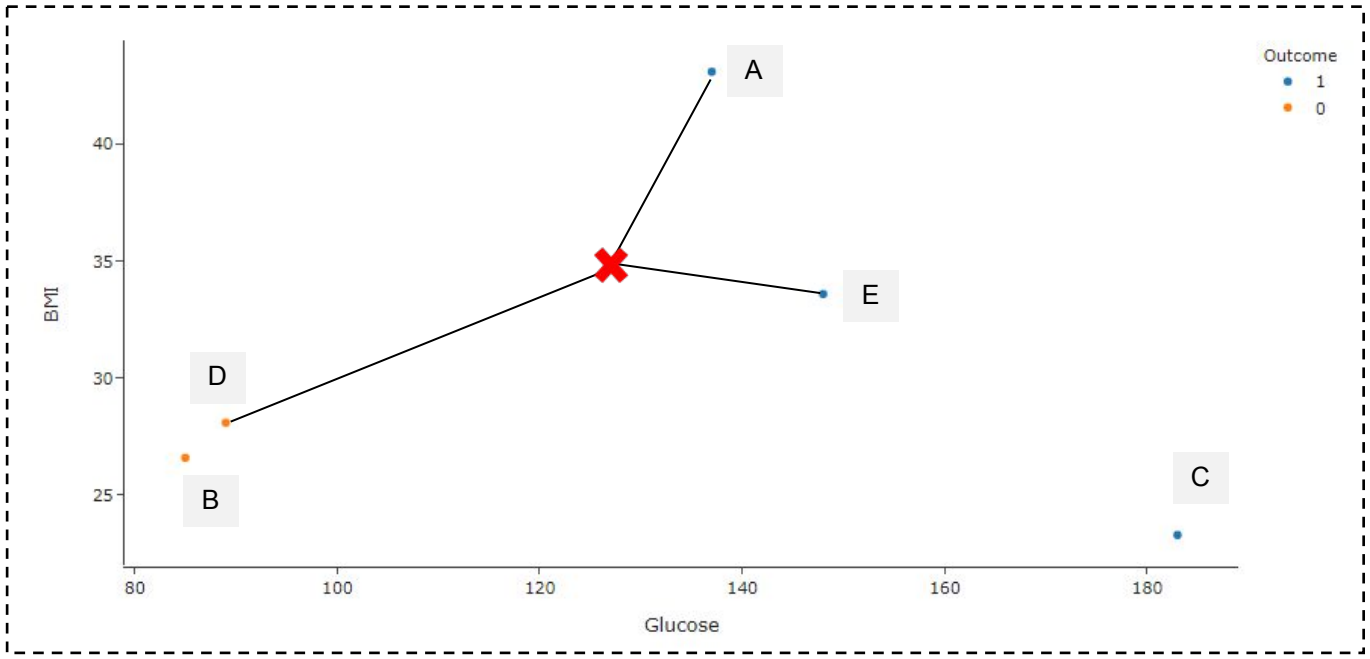
- When predicting the outcome of a new datapoint, we **calculate its distance from other labelled points** with Euclidean Distance**.**



Labelled Datapoint #	Labelled Datapoint Outcome	Distance from new datapoint ✖
A	1	12
B	0	50.7
C	1	49.4
D	0	46.5
E	1	13

MECHANISM BEHIND MODEL: MEASURING EUCLIDEAN DISTANCE

- When predicting the outcome of a new datapoint, we **calculate its distance from other labelled points** with Euclidean Distance**.**
- If we let k=3 (i.e. 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	12
B	0	50.7
C	1	49.4
D	0	46.5
E	1	13

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

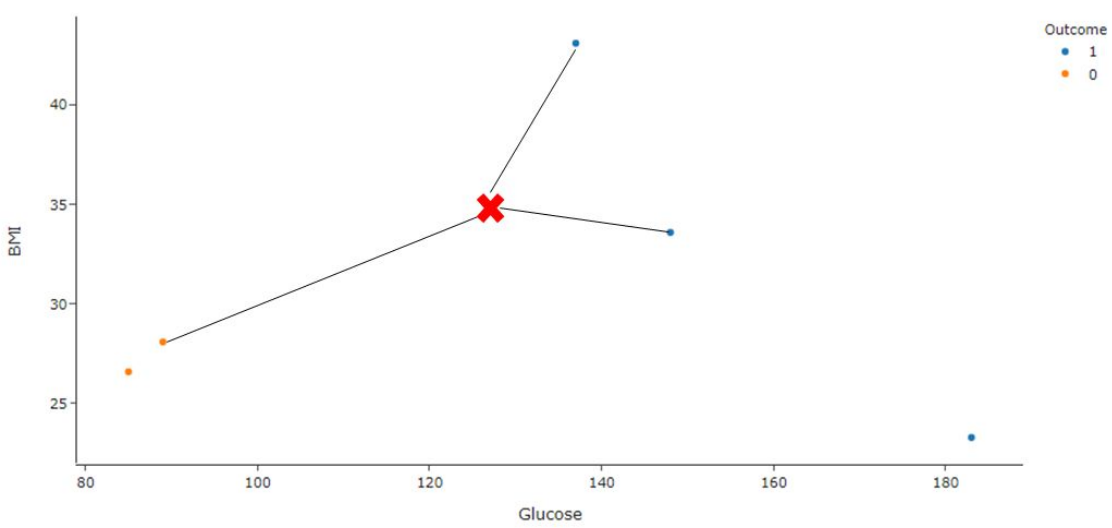
MECHANISM BEHIND MODEL: SCALING

- One thing to note: K-NN model is a distance-based model, so it is highly sensitive to **feature scaling**.
  Therefore, **make sure all your feature scales are standardised**! (more on this in Lesson 6)

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

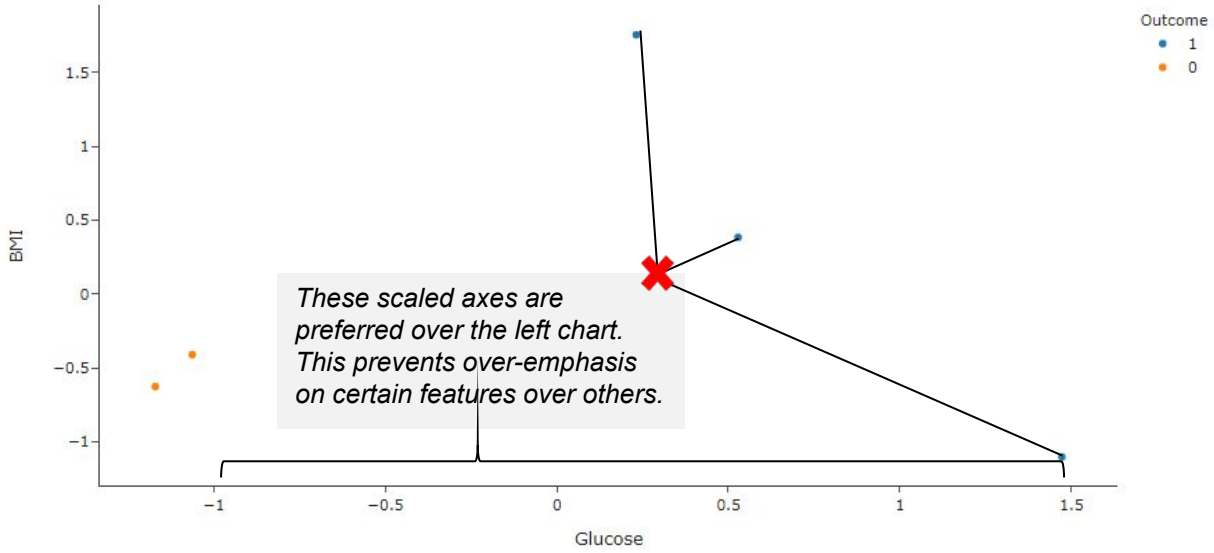
Notice that after scaling your features differently, the nearest 3 neighbours now share the same outcome of 1.

BEFORE SCALING



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

AFTER SCALING



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

MECHANISM BEHIND MODEL: SCALING *(sneak preview for Lesson 6)*

- Why is this so? You will realize the range of values of BMI (23 – 43) and Glucose (89 – 183) is quite different.
- **Differences in range will heavily skew distance-based models** such as K-NN, which would then favor a feature over the other.
- To resolve this, for all K-NN models we must scale all features before calculating the Euclidean distance.
  There are various methods to perform scaling, and one of them is **Standardization / Z-score Normalization**

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

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

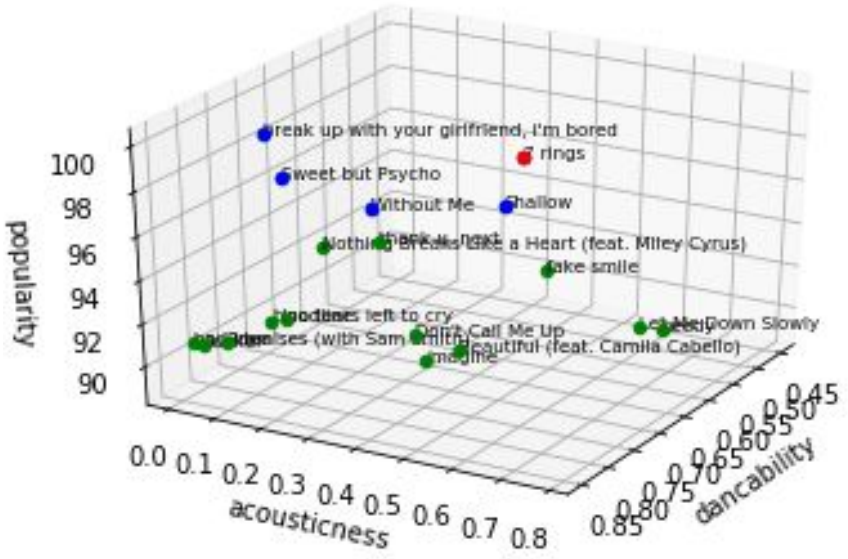
K-NEAREST NEIGHBOURS (K-NN)

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

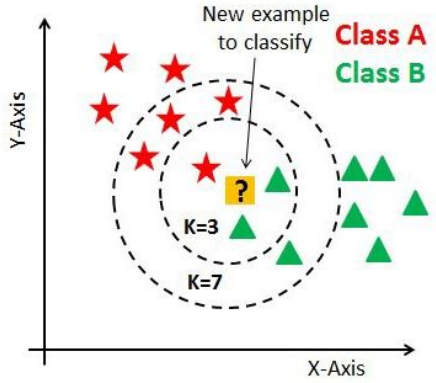


K-NEAREST NEIGHBOURS (K-NN)

STRATEGY FOR SELECTING THE K VALUE

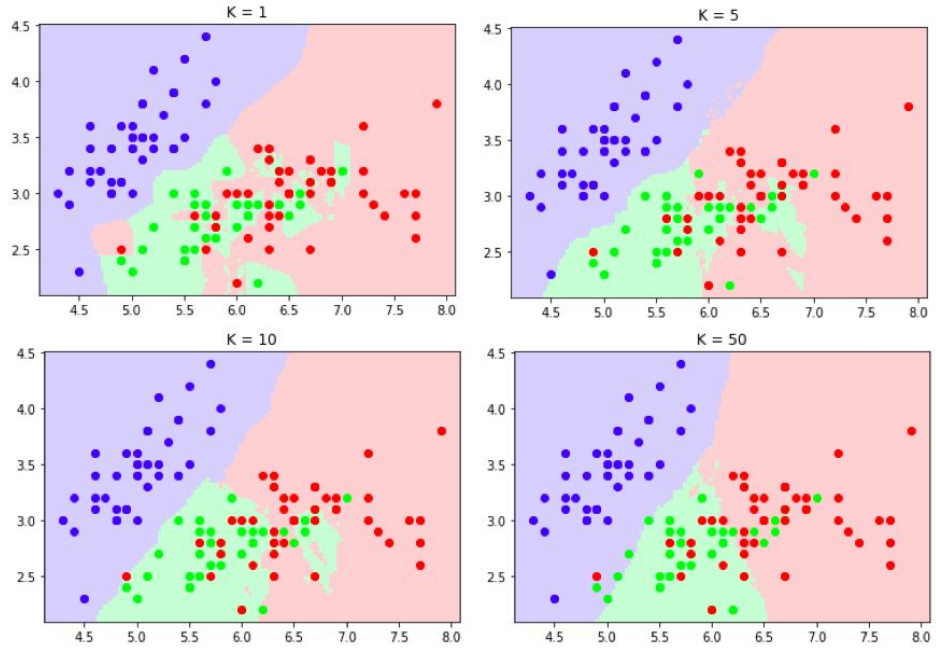
STRATEGY FOR SELECTING THE K VALUE: WHY IS THERE A NEED FOR THIS?

- 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

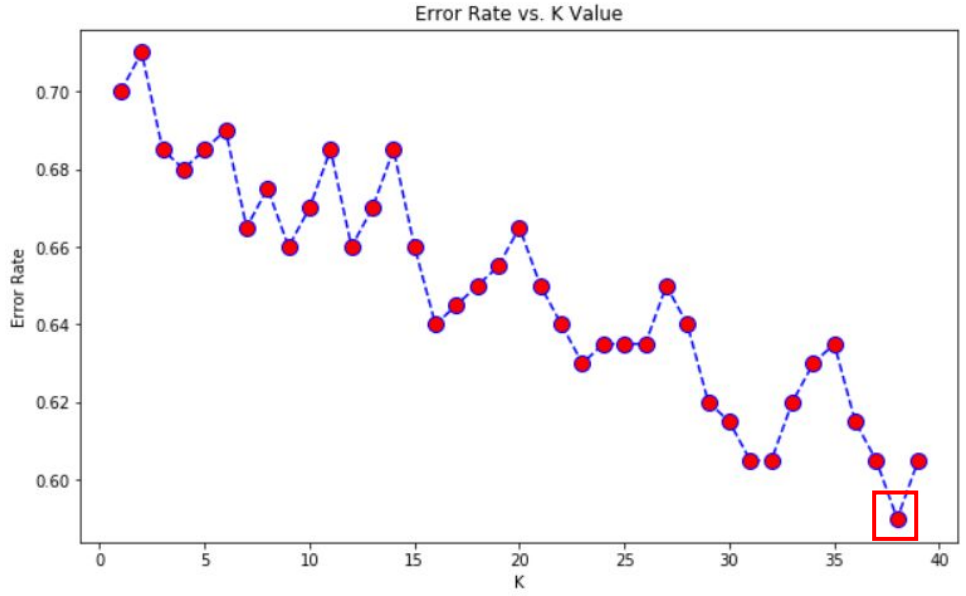
Example where K-NN is used to classify 3 types of outcomes



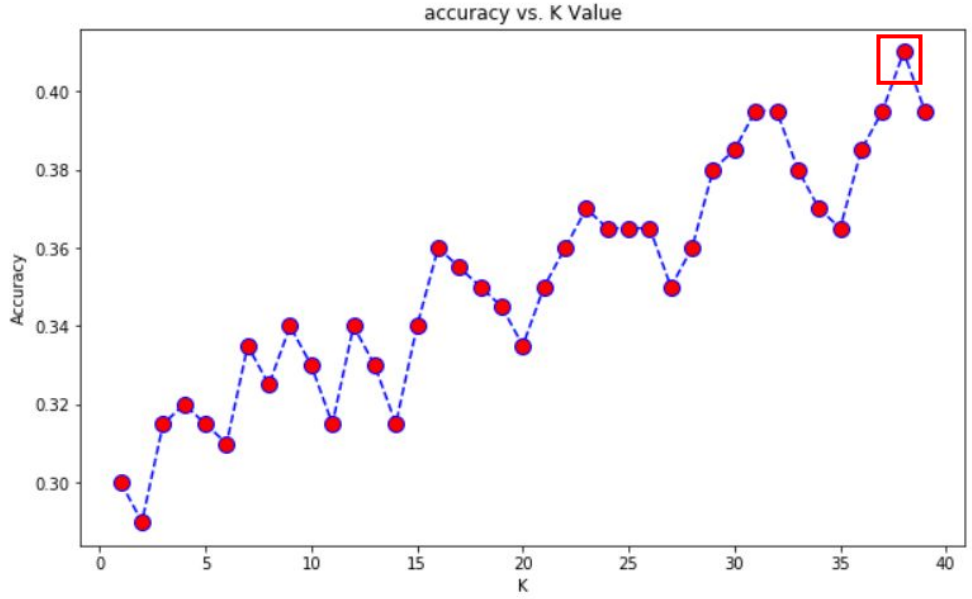
STRATEGY FOR SELECTING THE K VALUE: WHY IS THERE A NEED FOR THIS?

- 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