Data Mining and Machine Learning

Comp 3027J

Dr Catherine Mooney
Assistant Professor

catherine.mooney@ucd.ie

Lectures and Text

- **Core Text:**
  Fundamentals of Machine Learning for Predictive Data Analytics
  By John D. Kelleher, Brian Mac Namee and Aoife D'Arcy
- Has everyone got a copy of the book from the library?
- Last week we covered Chapter 3, sections 3.5 and 3.6 (Advanced Data Exploration).
- This week we will cover Chapter 5, sections 5.2, 5.3, 5.4.1 and 5.4.3.
- Please read these sections of the book.

Similarity-based Learning: Feature Spaces and Distance Metrics

- Similarity-based prediction models attempt to mimic a very human way of reasoning by basing predictions for a target feature value on the most similar instances in memory—this makes them **easy to interpret and understand**.
- This advantage should not be underestimated as being able to understand how the model works gives people more confidence in the model and, hence, in the insight that it provides.
- The **inductive bias** underpinning similarity-based classification is that **things that are similar** (i.e. instances that have similar descriptive features) **belong to the same class**.

- The fundamentals of similarity-based learning are:
  - Feature space
  - Similarity metrics

Feature Space

College Athlete Dataset

- Descriptive features – SPEED and AGILITY of each player (both measured out of 10).
- Target feature – If the player was drafted to a professional team DRAFT (yes or no).

Table: The speed and agility ratings for 20 college athletes labelled with the decisions for whether they were drafted or not.

ID	Speed	Agility	Draft	ID	Speed	Agility	Draft
1	2.50	6.00	No	11	2.00	2.00	No
2	3.75	8.00	No	12	5.00	2.50	No
3	2.25	5.50	No	13	8.25	8.50	No
4	3.25	8.25	No	14	5.75	8.75	Yes
5	2.75	7.50	No	15	4.75	6.25	Yes
6	4.50	5.00	No	16	5.50	6.75	Yes
7	3.50	5.25	No	17	5.25	9.50	Yes
8	3.00	3.25	No	18	7.00	4.25	Yes
9	4.00	4.00	No	19	7.50	8.00	Yes
10	4.25	3.75	No	20	7.25	5.75	Yes

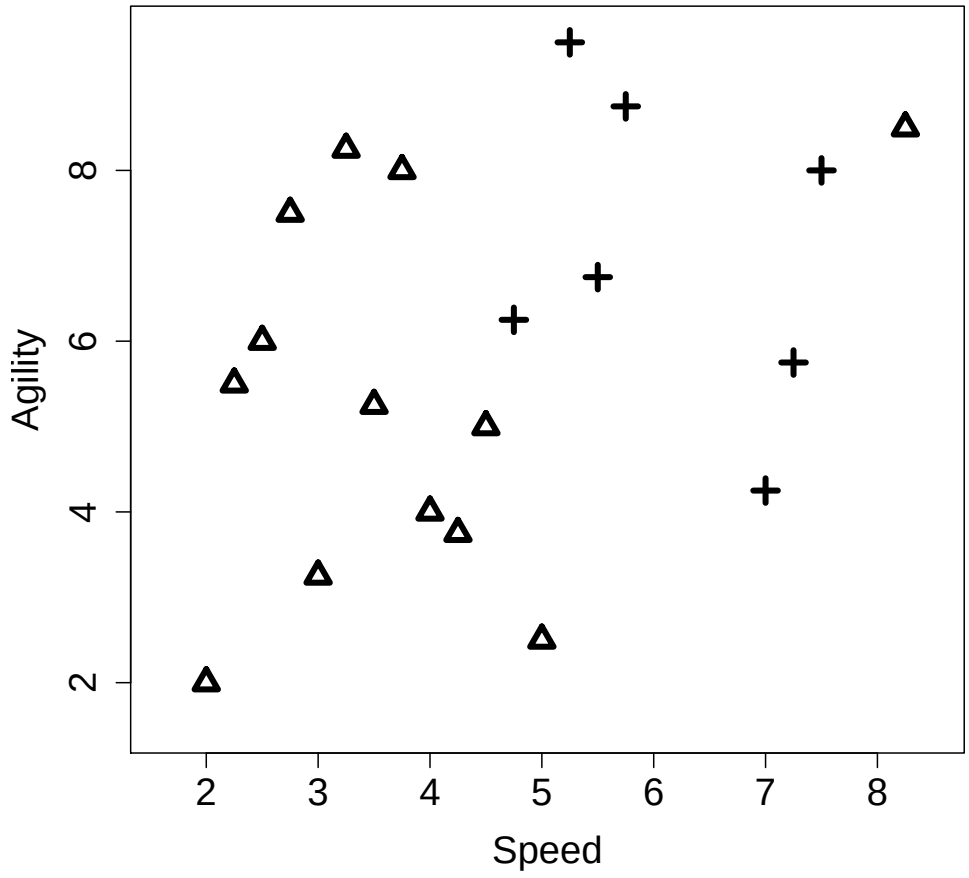


Figure: A feature space plot of the College Athlete data. In this figure SPEED has been plotted on the horizontal axis, and AGILITY has been plotted on the vertical axis. The value of the DRAFT feature is indicated by the shape representing each instance as a point in the feature space: triangles for no and crosses for yes.

- A feature space is an abstract *n*-dimensional space that is created by taking each of the descriptive features in an ABT to be the axes of a reference space and each instance in the dataset is mapped to a point in the feature space based on the values of its descriptive features.
- There is always one dimension for every descriptive feature in a dataset.

- If the values of the descriptive features of two or more instances in the dataset are the same, then these instances will be mapped to same point in the feature space.
- As the differences between the values of the descriptive features of two instances grows, so too does the distance between the points in the feature space that represent these instances.
- The distance between two points in the feature space is a useful measure of the similarity of the descriptive features of the two instances.

Measuring Similarity Using Distance Metrics

- A similarity metric measures the similarity between two instances according to a feature space
- Mathematically, a metric must conform to the following four criteria:
  1. Non-negativity: $\mathit{metric}(\mathbf{a}, \mathbf{b}) \geq 0$
  2. Identity: $\mathit{metric}(\mathbf{a}, \mathbf{b}) = 0 \Longleftrightarrow \mathbf{a} = \mathbf{b}$
  3. Symmetry: $\mathit{metric}(\mathbf{a}, \mathbf{b}) = \mathit{metric}(\mathbf{b}, \mathbf{a})$
  4. Triangular Inequality: $\mathit{metric}(\mathbf{a}, \mathbf{b}) \leq \mathit{metric}(\mathbf{a}, \mathbf{c}) + \mathit{metric}(\mathbf{b}, \mathbf{c})$

  where $\mathit{metric}(\mathbf{a}, \mathbf{b})$ is a function that returns the distance between two instances **a** and **b**.

- One of the best known metrics is Euclidean distance which computes the length of the straight line between two points. Euclidean distance between two instances **a** and **b** in a *m*-dimensional feature space is defined as:

$$Euclidean(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{m} (\mathbf{a}[i] - \mathbf{b}[i])^2} \quad (1)$$

Example

The Euclidean distance between instances
d_{12} (SPEED= 5.00, AGILITY= 2.5)
and
d_5 (SPEED= 2.75,AGILITY= 7.5)
in our College Athlete data is:

Example

The Euclidean distance between instances
d_{12} (SPEED= 5.00, AGILITY= 2.5)
and
d_5 (SPEED= 2.75,AGILITY= 7.5)
in our College Athlete data is:

$$\begin{aligned} \mathit{Euclidean}(\langle 5.00, 2.50\rangle, \langle 2.75, 7.50\rangle) &= \sqrt{(5.00-2.75)^2 + (2.50-7.50)^2} \\ &= \sqrt{30.0625} = 5.4829 \end{aligned}$$

- Another, less well known, distance measure is the Manhattan distance or taxi-cab distance.
- The Manhattan distance between two instances **a** and **b** in a feature space with m dimensions is:[1]

$$Manhattan(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{m} abs(\mathbf{a}[i] - \mathbf{b}[i]) \quad (2)$$

[1] The *abs*() function surrounding the subtraction term indicates that we use the absolute value, i.e. non-negative value, when we are summing the differences; this makes sense because distances can't be negative.

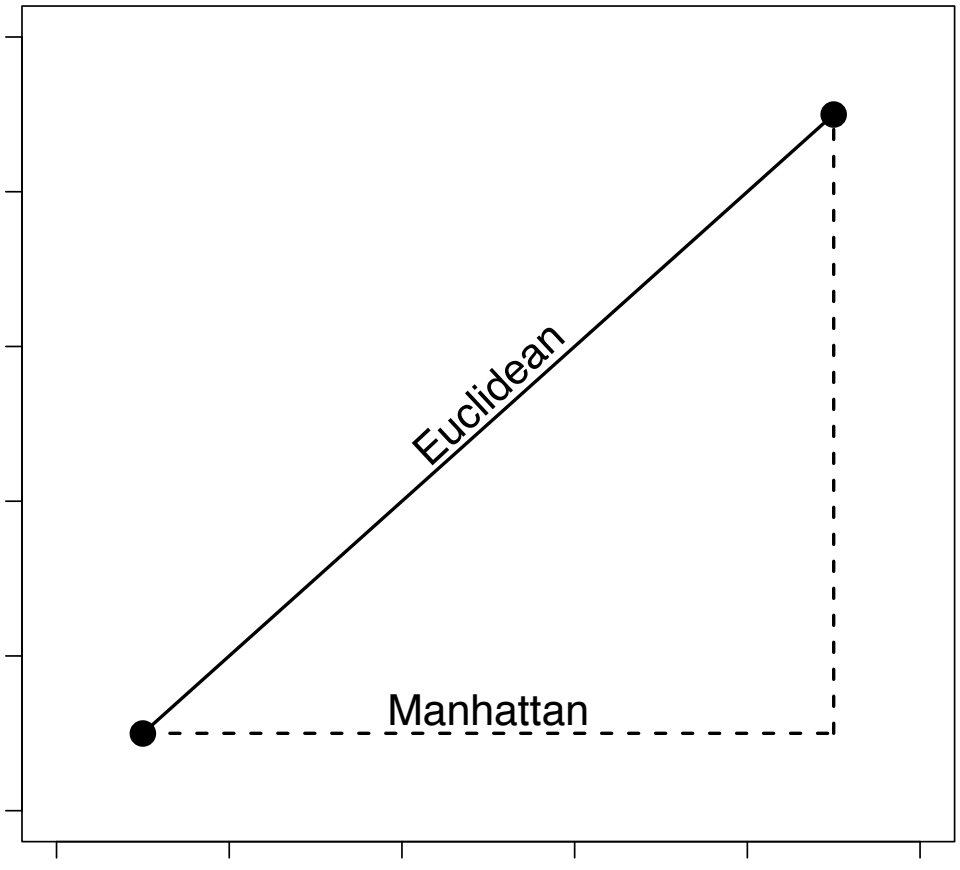


Figure: The Manhattan and Euclidean distances between two points.

Example

The Manhattan distance between instances
d_{12} (SPEED= 5.00, AGILITY= 2.5)
and
d_5 (SPEED= 2.75,AGILITY= 7.5)
in our College Athlete data is:

Example

The Manhattan distance between instances
d_{12} (SPEED= 5.00, AGILITY= 2.5)
and
d_5 (SPEED= 2.75,AGILITY= 7.5)
in our College Athlete data is:

$$\begin{aligned} \mathit{Manhattan}(\langle 5.00, 2.50\rangle, \langle 2.75, 7.50\rangle) &= \mathit{abs}(5.00 - 2.75) + \mathit{abs}(2.5 - 7.5) \\ &= 2.25 + 5 = 7.25 \end{aligned}$$

- The Euclidean and Manhattan distances are special cases of Minkowski distance
- The Minkowski distance between two instances **a** and **b** in a feature space with m descriptive features is:

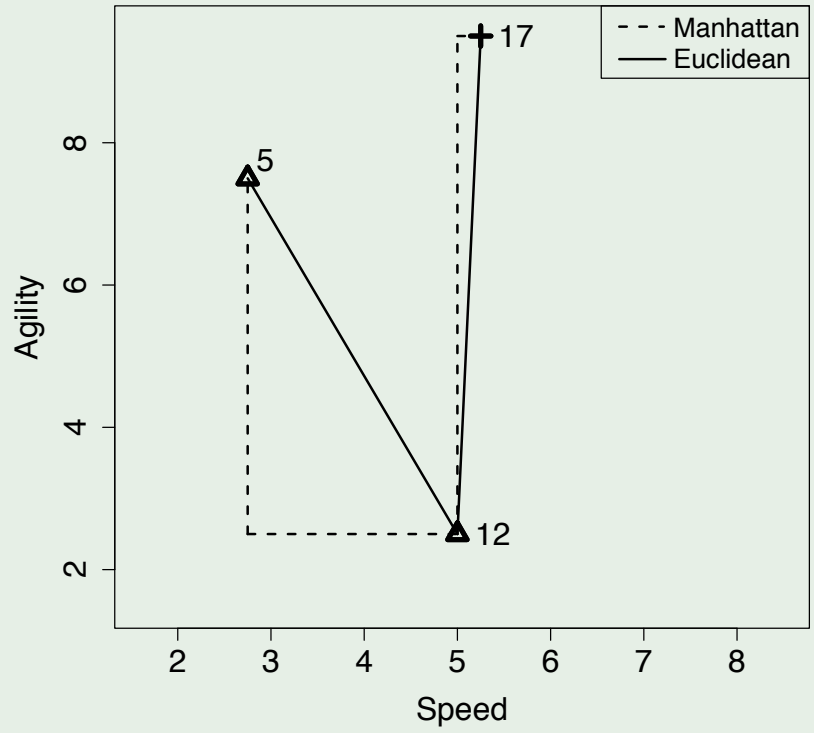
$$Minkowski(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^{m} abs(\mathbf{a}[i] - \mathbf{b}[i])^p \right)^{\frac{1}{p}} \quad (3)$$

  where different values of the parameter p result in different distance metrics
- The Minkowski distance with $p = 1$ is the Manhattan distance and with $p = 2$ is the Euclidean distance.

- The larger the value of p the more emphasis is placed on the features with large differences in values because these differences are raised to the power of p.
- The Euclidean distance (with p = 2) is more strongly influenced by a single large difference in one feature than the Manhattan distance (with p = 1).

Example

Instance ID	Instance ID	Manhattan (Minkowski p=1)	Euclidean (Minkowski p=2)
12	5	7.25	5.4829
12	17	7.25	7



The Manhattan and Euclidean distances between instances $\mathbf{d}_{12}$ (SPEED= 5.00, AGILITY= 2.5) and $\mathbf{d}_5$ (SPEED= 2.75, AGILITY= 7.5) and between instances $\mathbf{d}_{12}$ and $\mathbf{d}_{17}$ (SPEED= 5.25, AGILITY= 9.5).

Standard Approach: The Nearest Neighbour Algorithm

The Nearest Neighbour Algorithm

Require: set of training instances
Require: a query to be classified

1: Iterate across the instances in memory and find the instance that is shortest distance from the query position in the feature space.
2: Make a prediction for the query equal to the value of the target feature of the nearest neighbour.

Table: The speed and agility ratings for 20 college athletes labelled with the decisions for whether they were drafted or not.

ID	Speed	Agility	Draft	ID	Speed	Agility	Draft
1	2.50	6.00	No	11	2.00	2.00	No
2	3.75	8.00	No	12	5.00	2.50	No
3	2.25	5.50	No	13	8.25	8.50	No
4	3.25	8.25	No	14	5.75	8.75	Yes
5	2.75	7.50	No	15	4.75	6.25	Yes
6	4.50	5.00	No	16	5.50	6.75	Yes
7	3.50	5.25	No	17	5.25	9.50	Yes
8	3.00	3.25	No	18	7.00	4.25	Yes
9	4.00	4.00	No	19	7.50	8.00	Yes
10	4.25	3.75	No	20	7.25	5.75	Yes

Example

- Should we draft an athlete with the following profile:

$$\text{SPEED} = 6.75,\ \text{AGILITY} = 3$$

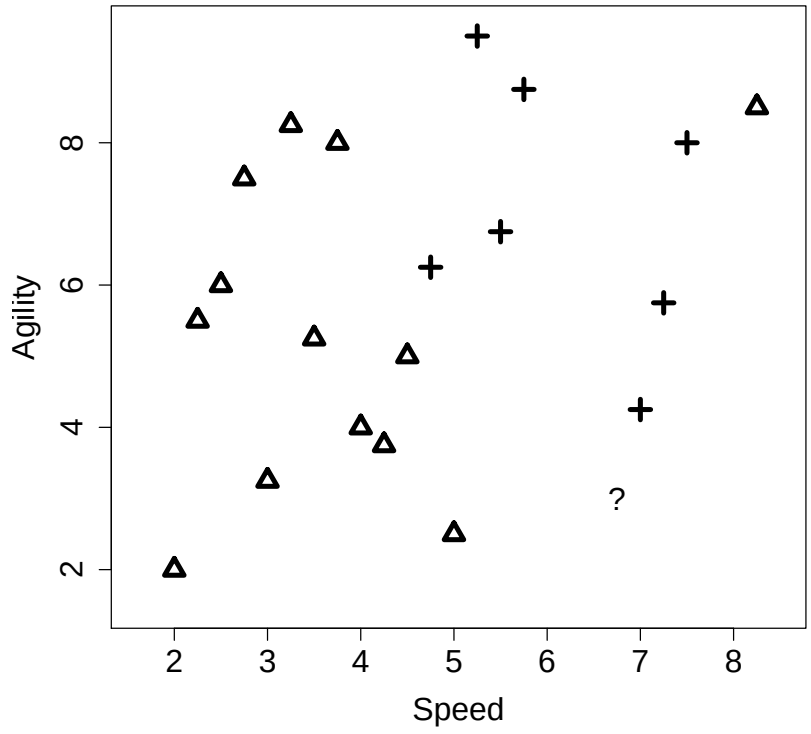


Figure: A feature space plot of the College Athlete data with the position in the feature space of the query represented by the ? marker. The value of the DRAFT feature is indicated by the shape representing each instance as a point in the feature space: **triangles for no and crosses for yes**.

- During the prediction stage, the nearest neighbour algorithm iterates across all the instances in the training dataset and computes the distance between each instance and the query.
- These distances are then ranked from lowest to highest to find the nearest neighbour.

Table: The distances (Dist.) between the query instance with SPEED = 6.75 and AGILITY = 3.00 and each instance in the College Athlete data.

ID	SPEED	AGILITY	DRAFT	Dist.
18	7.00	4.25	yes	1.27
12	5.00	2.50	no	1.82
10	4.25	3.75	no	2.61
20	7.25	5.75	yes	2.80
9	4.00	4.00	no	2.93
6	4.50	5.00	no	3.01
8	3.00	3.25	no	3.76
15	4.75	6.25	yes	3.82
7	3.50	5.25	no	3.95
16	5.50	6.75	yes	3.95
11	2.00	2.00	no	4.85
19	7.50	8.00	yes	5.06
3	2.25	5.50	no	5.15
1	2.50	6.00	no	5.20
13	8.25	8.50	no	5.70
2	3.75	8.00	no	5.83
14	5.75	8.75	yes	5.84
5	2.75	7.50	no	6.02
4	3.25	8.25	no	6.31
17	5.25	9.50	yes	6.67

- When the algorithm is searching for the nearest neighbour using Euclidean distance, it is partitioning the feature space into what is known as a **Voronoi tessellation**.
- It is trying to decide which Voronoi region the query belongs to.
- The Voronoi region belonging to a training instance defines the set of queries for which the prediction will be determined by that training instance.

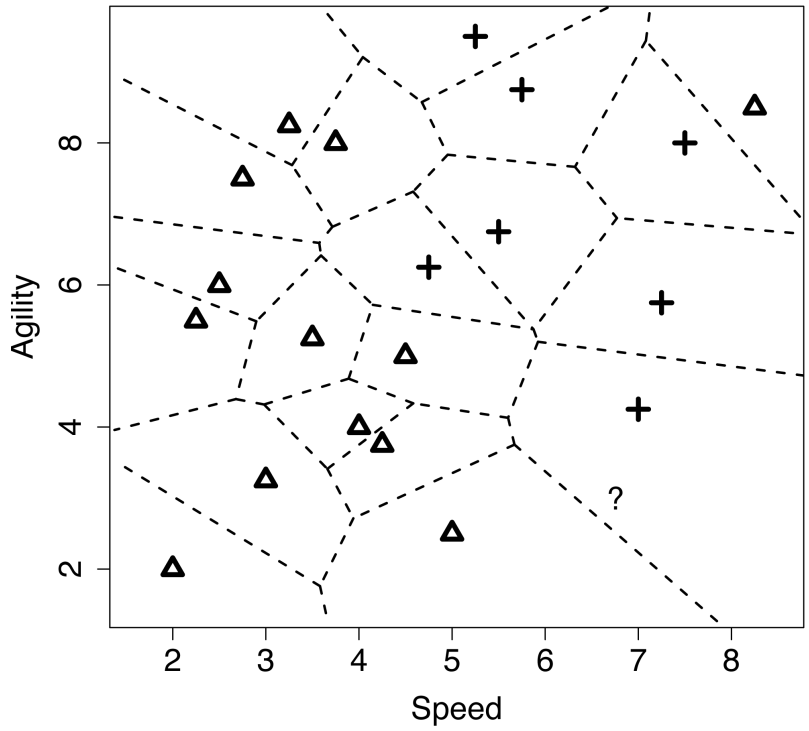


Figure: The Voronoi tessellation of the feature space for the College Athlete data with the position of the query represented by the ? marker

- The nearest neighbour prediction algorithm creates a set of local models, or neighbourhoods, across the feature space
- Each model is defined by a subset of the training dataset (in this case, one instance)
- The algorithm is also creating a global prediction model based on the full dataset
- We can see this if we highlight the **decision boundary** within the feature space

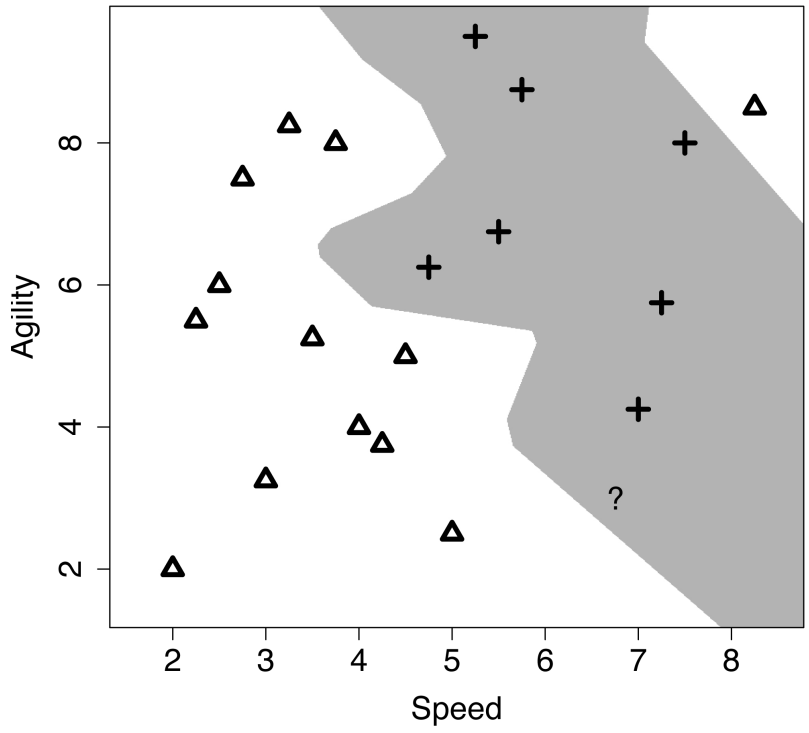


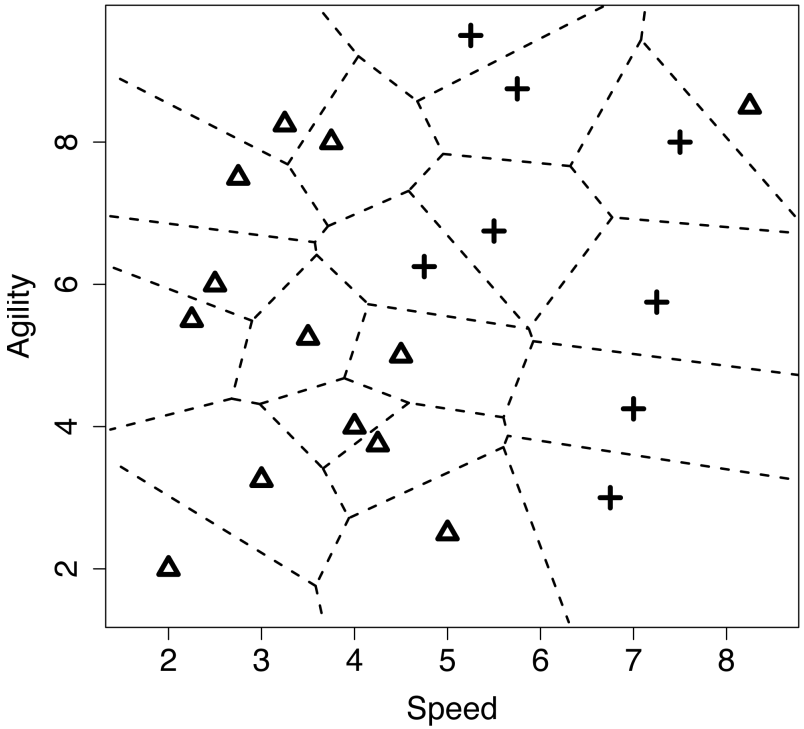
Figure: The decision boundary created by aggregating the neighbouring Voronoi regions that belong to the same target level.

- The **decision boundary** is the boundary between regions of the feature space in which different target levels will be predicted.
- We can generate the decision boundary by aggregating the neighbouring local models (in this case **Voronoi regions**) that make the same prediction.

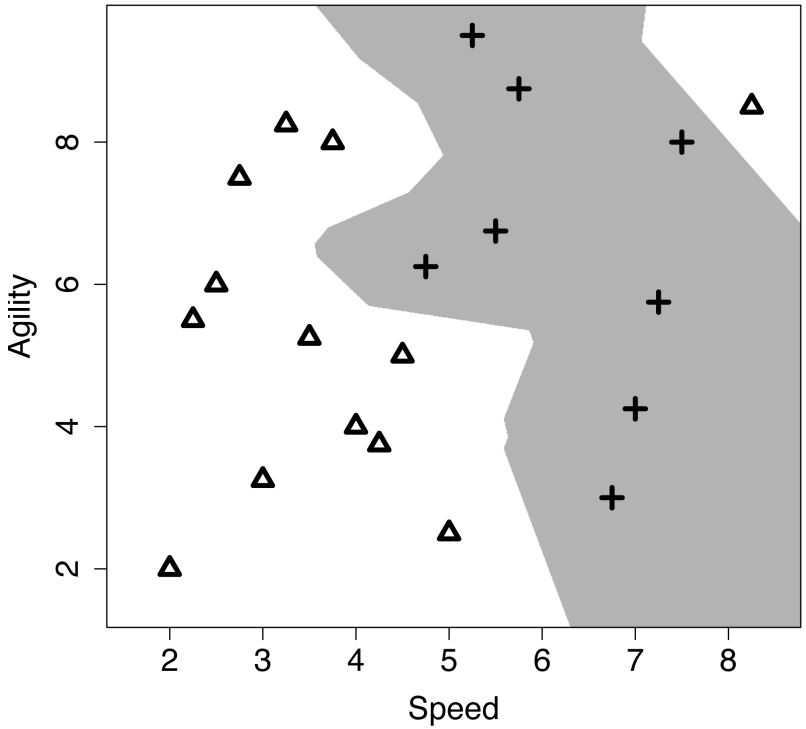
- One of the great things about nearest neighbour algorithms is that we can add in new data to update the model very easily.

Table: The extended version of the college athletes dataset.

ID	SPEED	AGILITY	DRAFT	ID	SPEED	AGILITY	DRAFT
1	2.50	6.00	no	12	5.00	2.50	no
2	3.75	8.00	no	13	8.25	8.50	no
3	2.25	5.50	no	14	5.75	8.75	yes
4	3.25	8.25	no	15	4.75	6.25	yes
5	2.75	7.50	no	16	5.50	6.75	yes
6	4.50	5.00	no	17	5.25	9.50	yes
7	3.50	5.25	no	18	7.00	4.25	yes
8	3.00	3.25	no	19	7.50	8.00	yes
9	4.00	4.00	no	20	7.25	5.75	yes
10	4.25	3.75	no	21	6.75	3.00	yes
11	2.00	2.00	no				



(a) Voronoi tessellation



(b) Decision boundary ($k = 1$)

Figure: (a) The Voronoi tessellation of the feature space when the dataset has been updated to include the query instance; (b) the updated decision boundary reflecting the addition of the query instance in the training set.

Summary

- The inductive bias underpinning similarity-based machine learning algorithms is that things that are similar (i.e. instances that have similar descriptive features) also have the same target feature values.
- The nearest neighbour algorithm creates an implicit global predictive model by aggregating local models, or neighbourhoods.
- The definition of these neighbourhoods is based on similarity within the feature space to the labelled training instances.
- Predictions are made for a query instance using the target level of the training instance defining the neighbourhood in the feature space that contains the query.

Any questions so far?
5 min break...

Handling Noisy Data

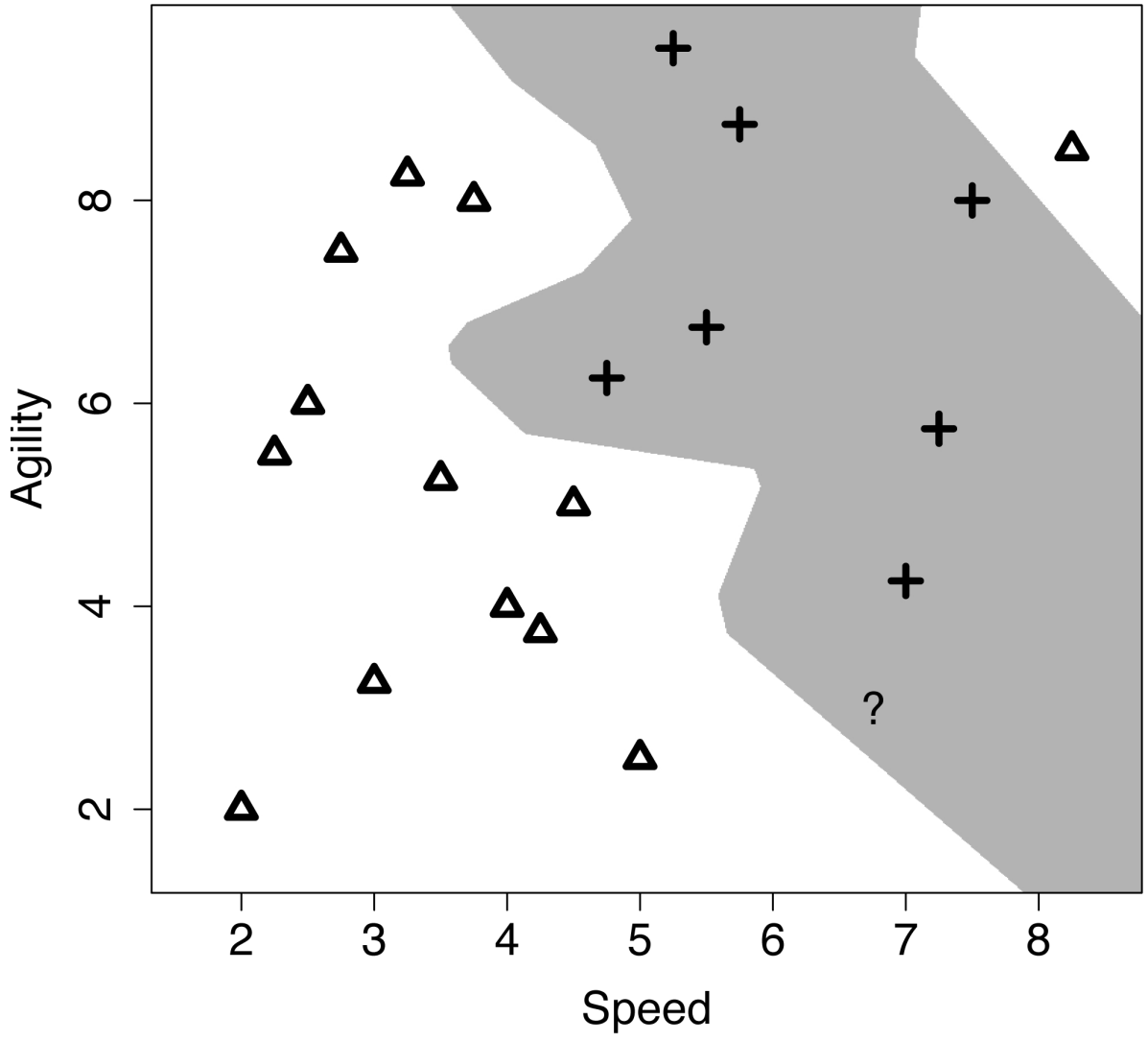


Figure: Is the instance at the top right of the diagram really *noise*?

- The top right corner of the feature space contained a no region.
- Considering that all the immediate neighbours of this instance are associated with the yes target level, it is likely that either:
  - This instance has been incorrectly labelled and should have a target feature value of yes, or
  - One of the descriptive features for this instance has an incorrect value and hence it is in the wrong location in the feature space.
- This instance is likely to be an example of **noise** in the dataset.

- The nearest neighbour algorithm is a set of local models, each defined using a single instance.
- The algorithm is sensitive to noise because any errors in the description or labelling of training data results in errors in the local models and incorrect predictions.
- We need to reduce the dependency of the algorithm on individual (possibly noisy) instances.
- To do this we simply modify the algorithm to return the majority target level within the set of k nearest neighbours to the query.

- The k nearest neighbours model predicts the target level with the majority vote from the set of k nearest neightbors to the query **q**:

$$\mathbb{M}_k(\mathbf{q}) = \underset{l \in levels(t)}{\text{argmax}} \sum_{i=1}^{k} \delta(t_i, l) \tag{4}$$

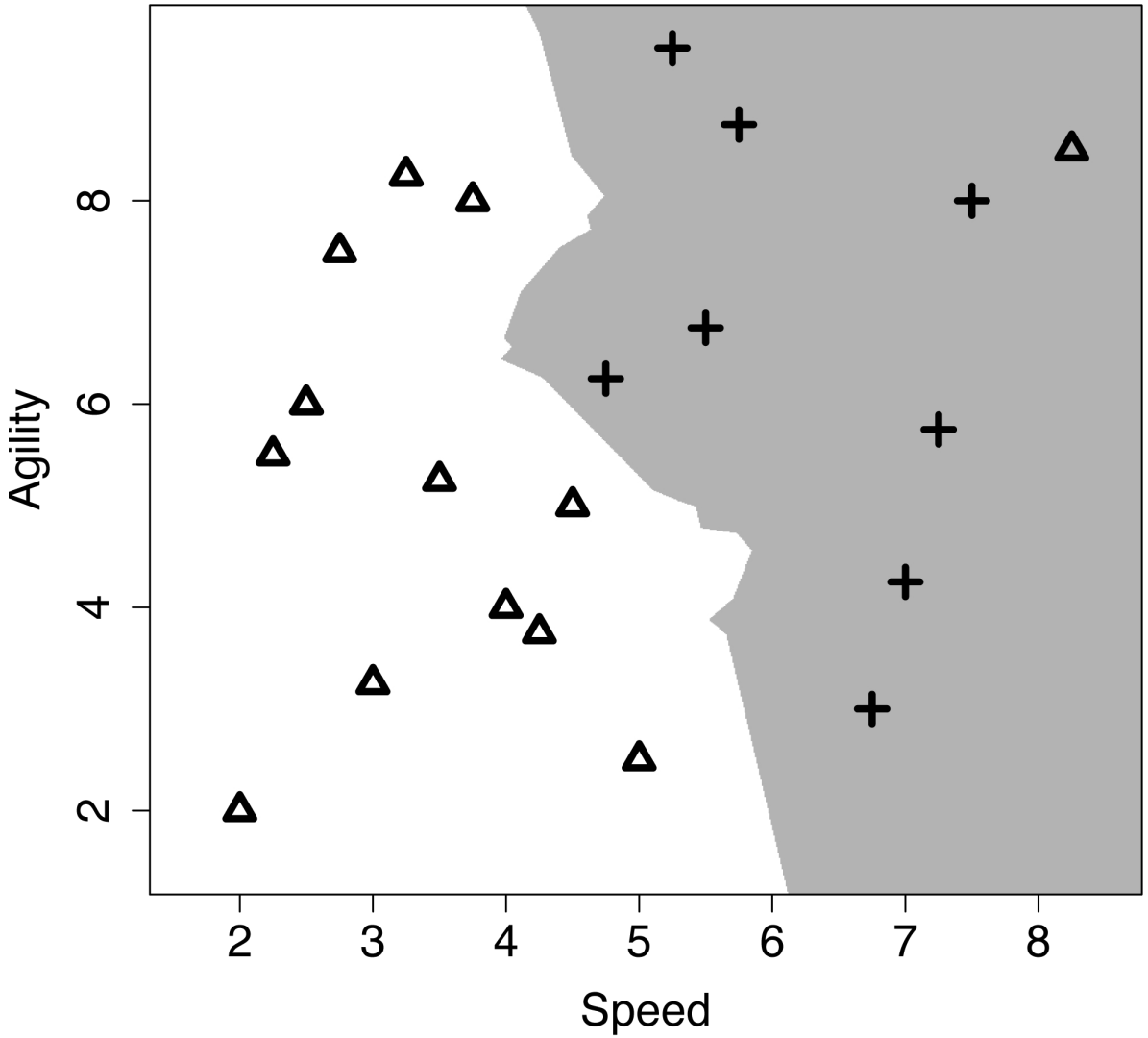


Figure: The decision boundary using majority classification of the nearest 3 neighbours.

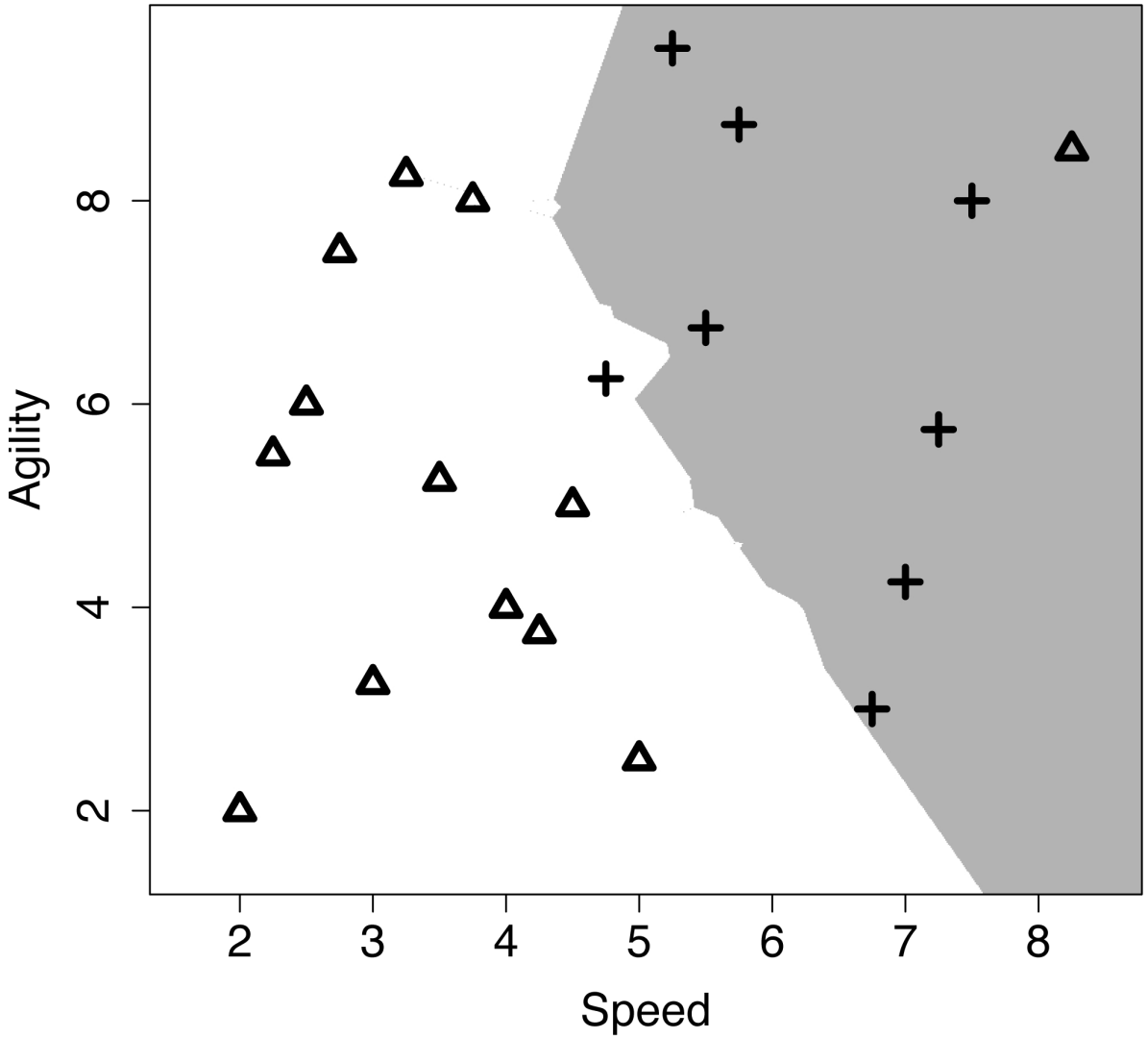


Figure: The decision boundary using majority classification of the nearest 5 neighbours.

- There is always a trade-off in setting the value of k.
- If we set k too, low we run the risk of the algorithm being sensitive to noise in the data and **overfitting**.
- If we set k too high, we run the risk of losing the true pattern of the data and **underfitting**.

- Need to be careful not to set k too high when we are dealing with an imbalanced dataset.
- An **imbalanced dataset** is a dataset that contains significantly more instances of one target level than another.
- As k increases, the majority target level will dominate the feature space.
- The dataset in the college athlete example is imbalanced—there are 13 no instances and only 7 yes instances.

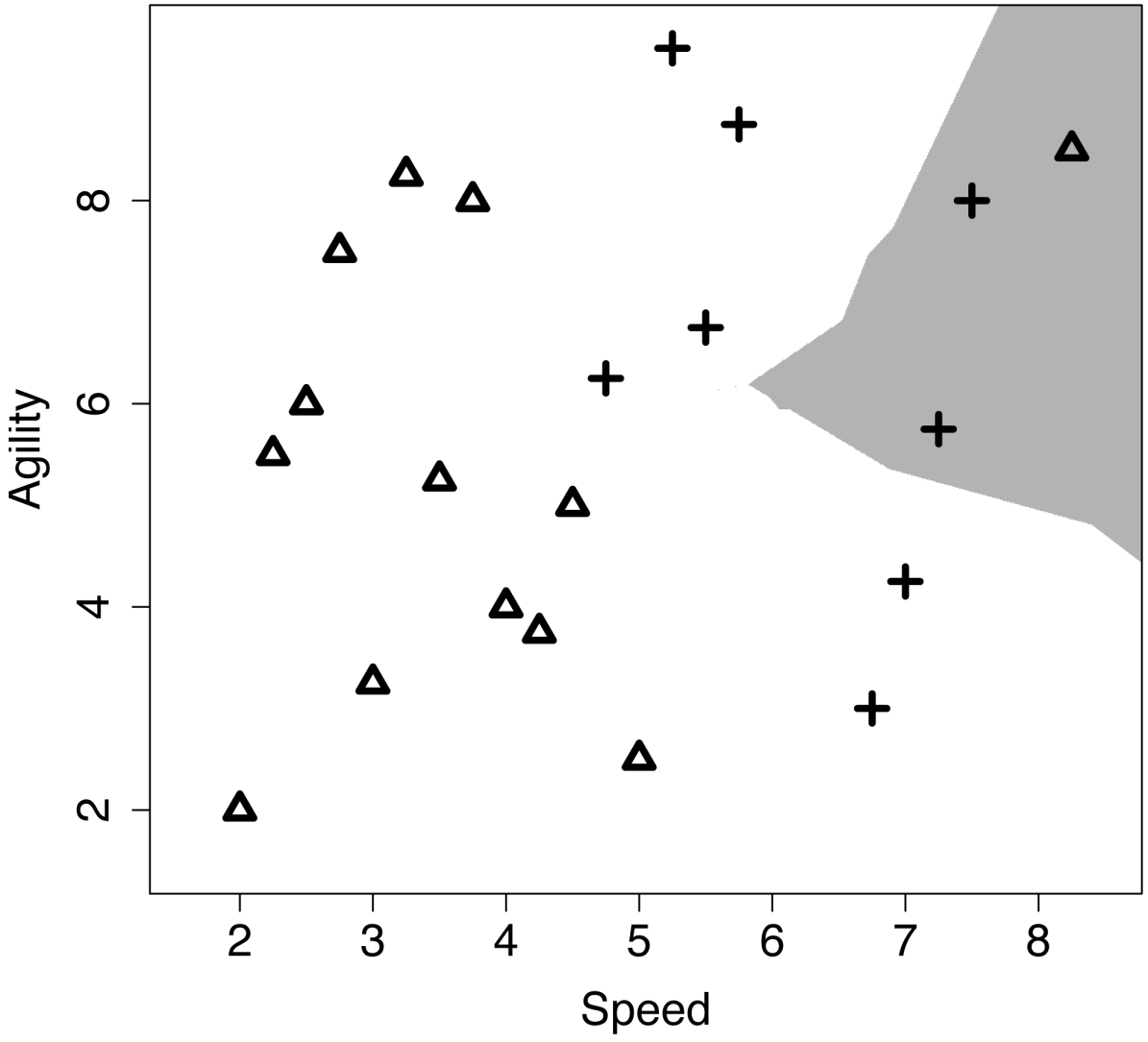


Figure: The decision boundary when k is set to 15.

- One way to address the problem of how to set k is to use a **distance weighted k nearest neighbour** approach.
- The contribution of each neighbour to the prediction is a function of the inverse distance between the neighbour and the query.
- The votes of the neighbors that are close to the query get a lot of weight, and the votes of the neighbors that are further away from the query get less weight.

- In a distance weighted k nearest neighbour algorithm the contribution of each neighbour to the classification decision is weighted by the reciprocal of the squared distance between the neighbour **d** and the query **q**:

$$\frac{1}{dist(\mathbf{q}, \mathbf{d})^2} \tag{5}$$

- The weighted k nearest neighbour model is defined as:

$$\mathbb{M}_k(\mathbf{q}) = \underset{l \in levels(t)}{\text{argmax}} \sum_{i=1}^{k} \frac{1}{dist(\mathbf{q}, \mathbf{d_i})^2} \times \delta(t_i, l) \tag{6}$$

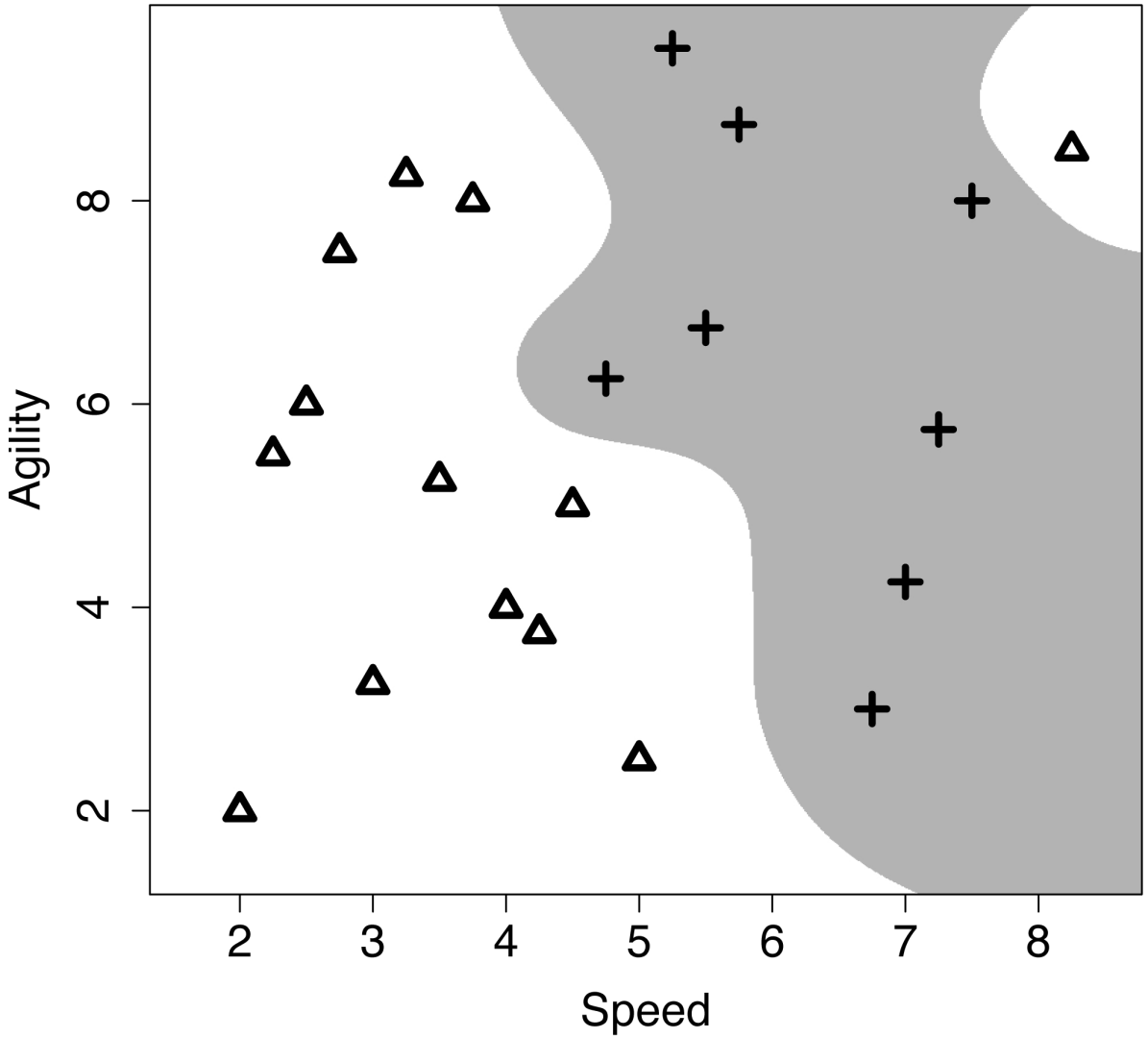


Figure: The weighted k nearest neighbour model decision boundary ($k = 21$).

Data Normalization

Table: A dataset listing the salary and age information for customers and whether or not they purchased a pension plan.

ID	Salary	Age	Purchased
1	53700	41	No
2	65300	37	No
3	48900	45	Yes
4	64800	49	Yes
5	44200	30	No
6	55900	57	Yes
7	48600	26	No
8	72800	60	Yes
9	45300	34	No
10	73200	52	Yes

- The marketing department wants to decide whether or not they should contact a customer with the following profile:

$$\langle \text{SALARY} = 56,000, \text{AGE} = 35 \rangle$$

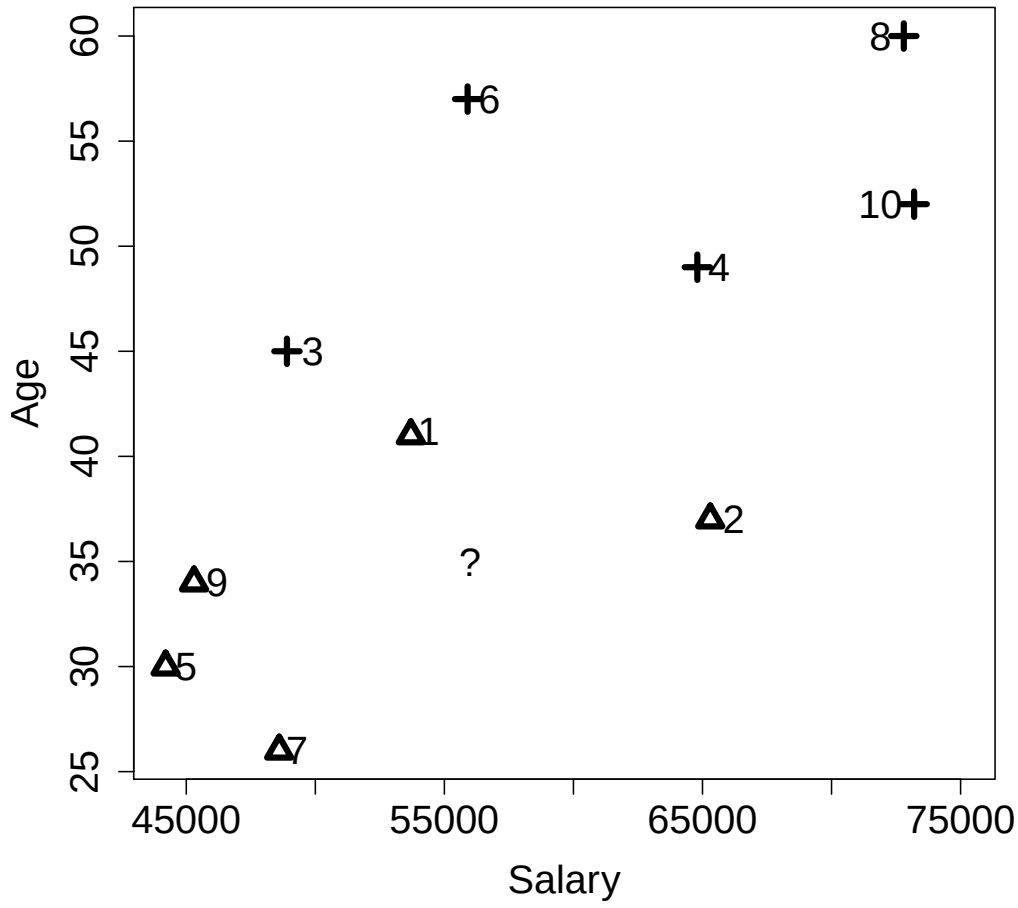


Figure: The salary and age feature space. The instances are labelled their IDs, triangles represent the negative instances and crosses represent the positive instances. The location of the query $\langle \text{SALARY} = 56,000, \text{AGE} = 35 \rangle$ is indicated by the ?.

From inspecting the plot, it would appear as if instance **d**1, which has a target level no, is the closest neighbour to the query. So we would expect that the model would predict no and that the customer would not be contacted.

However, the model actually predicts 'yes'. Why?

				Salary and Age		Salary Only		Age Only	
ID	Salary	Age	Purch.	Dist.	Rank	Dist.	Rank	Dist.	Rank
1	53700	41	No	2300.0078	2	2300	2	6	4
2	65300	37	No	9300.0002	6	9300	6	2	2
3	48900	45	Yes	7100.0070	3	7100	3	10	6
4	64800	49	Yes	8800.0111	5	8800	5	14	7
5	44200	30	No	11800.0011	8	11800	8	5	5
6	55900	57	Yes	102.3914	1	100	1	22	9
7	48600	26	No	7400.0055	4	7400	4	9	3
8	72800	60	Yes	16800.0186	9	16800	9	25	10
9	45300	34	No	10700.0000	7	10700	7	1	1
10	73200	52	Yes	17200.0084	10	17200	10	17	8

The dataset with the Euclidean distance (Dist.) between each instance and the query SALARY = 56,000, AGE = 35 when we use both the SALARY and AGE features, just the SALARY feature, and just the AGE feature. The Rank columns rank the distances of each instance to the query (1 is closest, 10 is furthest away).

- This odd prediction is caused by features taking different ranges of values, this is equivalent to features having different variances.
- We can adjust for this using normalization; the equation for range normalization is:

$$a_i' = \frac{a_i - min(a)}{max(a) - min(a)} \times (high - low) + low \tag{7}$$

	Normalized Dataset			Salary and Age		Salary Only		Age Only	
ID	Salary	Age	Purch.	Dist.	Neigh.	Dist.	Neigh.	Dist.	Neigh.
1	0.3276	0.4412	No	0.1935	1	0.0793	2	0.17647	4
2	0.7276	0.3235	No	0.3260	2	0.3207	6	0.05882	2
3	0.1621	0.5588	Yes	0.3827	5	0.2448	3	0.29412	6
4	0.7103	0.6765	Yes	0.5115	7	0.3034	5	0.41176	7
5	0.0000	0.1176	No	0.4327	6	0.4069	8	0.14706	3
6	0.4034	0.9118	Yes	0.6471	8	0.0034	1	0.64706	9
7	0.1517	0.0000	No	0.3677	3	0.2552	4	0.26471	5
8	0.9862	1.0000	Yes	0.9361	10	0.5793	9	0.73529	10
9	0.0379	0.2353	No	0.3701	4	0.3690	7	0.02941	1
10	1.0000	0.7647	Yes	0.7757	9	0.5931	10	0.50000	8

The updated version of the table once we have applied range normalization to the SALARY and AGE features in the dataset and to the query instance.

- Normalizing the data is an important thing to do for almost all machine learning algorithms, not just nearest neighbour!

Summary

- The standard approach to implementing a similarity-based prediction model is the nearest neighbour algorithm.
- This algorithm is built on two fundamental concepts:
  - a **feature space**
  - a **measure of similarity** between instances within the feature space.

- The **inductive bias** underpinning similarity-based classification is that things that are similar (i.e. instances that have similar descriptive features) belong to the same class.
- The nearest neighbour algorithm creates an implicit global predictive model by aggregating local models, or neighbourhoods.
- The definition of these neighbourhoods is based on proximity within the feature space to the labelled training instances.
- Queries are classified using the label of the training instance defining the neighbourhood in the feature space that contains the query.

- Nearest neighbour models are very sensitive to noise in the target feature the easiest way to solve this problem is to employ a ***k* nearest neighbour**.
- **Normalization** techniques should almost always be applied when nearest neighbour models are used.
- It is easy to adapt a nearest neighbour model to **continuous targets** (not covered today but you can read about how to do this in your book).
- There are many different measures of **similarity** (also covered in more detail in the book).
- **Feature selection** is a particularly important process for nearest neighbour algorithms it alleviates the **curse of dimensionality**.

- The nearest neighbour algorithm is what is known as a **lazy learner**...
- Delays abstracting from the data until it is asked to make a prediction.
- At this point the information in the query is used to define a neighbourhood in the feature space, and a prediction is made based on the instances in this neighbourhood.
- As the number of instances becomes large, the model will become slower because it has more instances to check when defining the neighborhood.

- An advantage of the lazy learning strategy, however, is that similarity-based machine learning approaches are robust to concept drift.
- **Concept drift** is a phenomenon that occurs when the relationship between the target feature and the descriptive features changes over time.

Weaknesses

- They are sensitive to the curse of dimensionality.
- They are slower than other models at making predictions (particularly with very large datasets).
- They may not be able to achieve the same levels of accuracy as other learning approaches.

Strengths

- They are easy to interpret.
- They can handle different types of descriptive features.
- They are relatively robust to noise.
- They are more robust to concept drift.

Recommended Reading

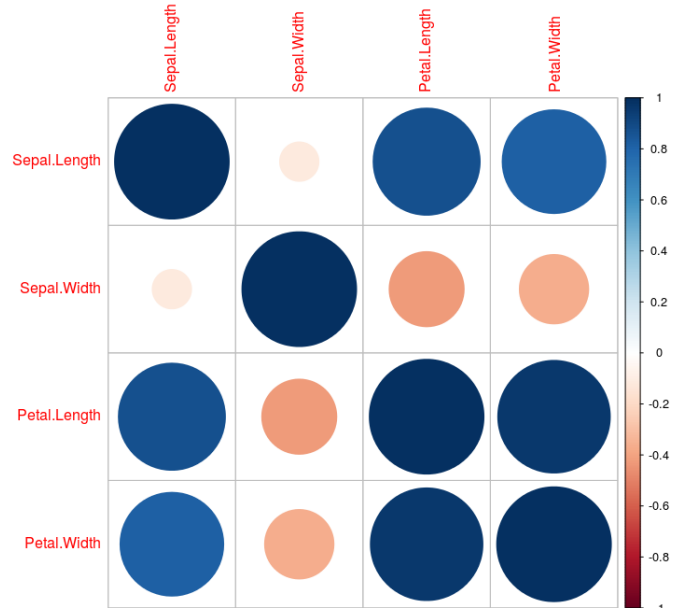
- **Core Text:**
  Fundamentals of Machine Learning for Predictive Data Analytics
  By John D. Kelleher, Brian Mac Namee and Aoife D'Arcy
- This week we covered Chapter 5, sections 5.2, 5.3, 5.4.1 and 5.4.3.
- I would suggest that you would read over these sections again
- Email me if you have any questions and I will cover them at the beginning of class next week
- Next week we will cover "Feature Selection" and "Evaluation" (Chapter 8)

Feedback from Lab 2

- Mostly issues with network and downloading package
- In future, when a package is required I will let you know in advance so you can download it before the lab

Packages to download

- install.packages('Hmisc')
- install.packages('corrplot')
- install.packages('caret')
- install.packages('class')
- install.packages('randomForest')
- install.packages('dplyr')
- install.packages('dlookr')
- install.packages('nycflights13')



- The sizes of the circles are scaled according to the absolute value of the strength of the correlation to draw attention to those pairs of features with the strongest relationships.
- Blue represents positive correlations and red represents negative or inverse correlations.

Preview of Lab 3

- In Lab 3 I will be explaining the first part of your assignment to you
- Your assignment is worth 50% of your final grade!!
- Make sure that you attend the lab this Thursday