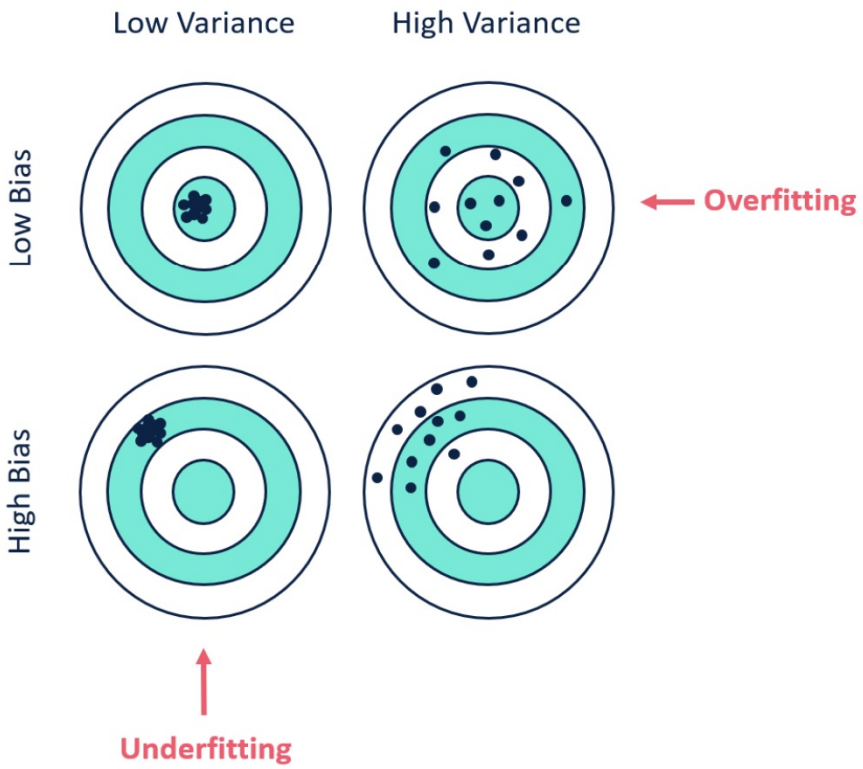
ML interviews pt 2

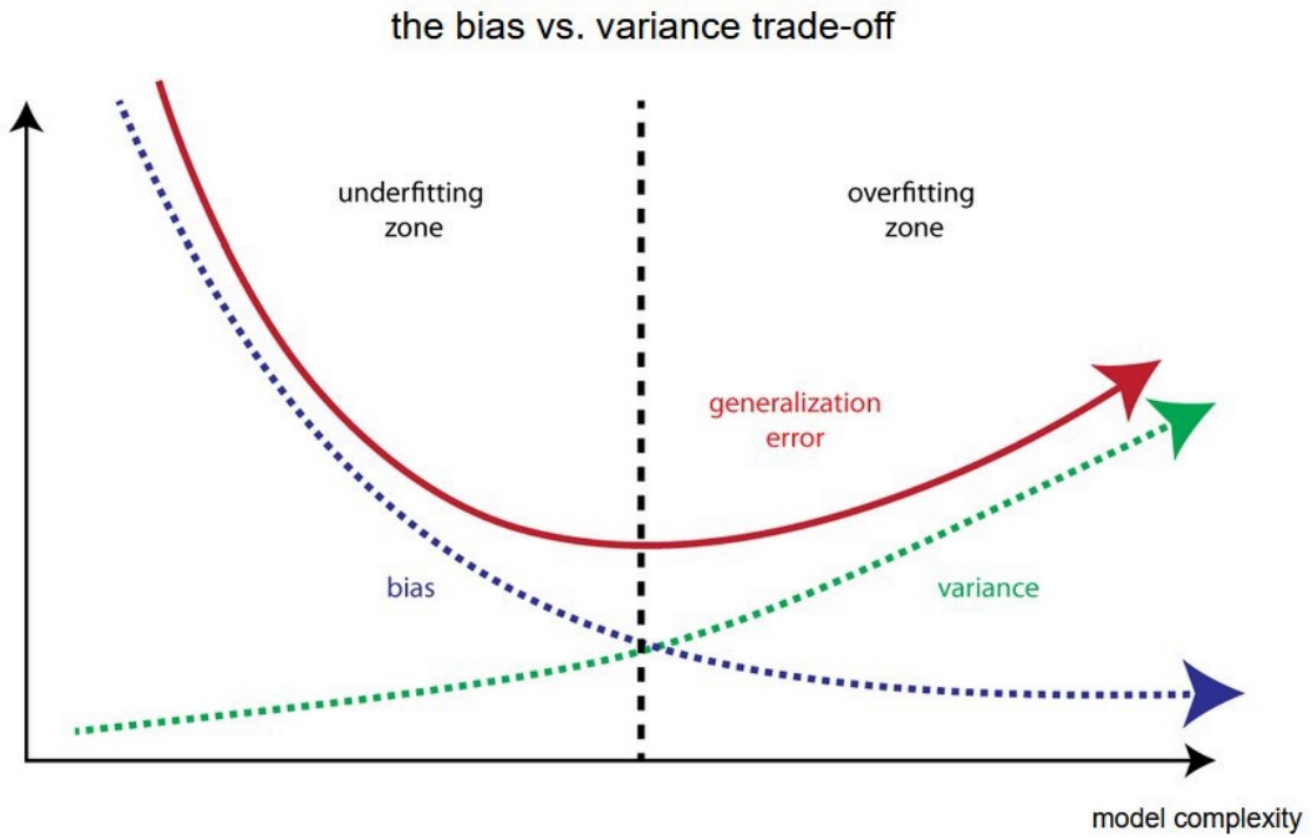
MAXIMOVSKAYA
ANASTASIA

Bias-Variance Trade-Off

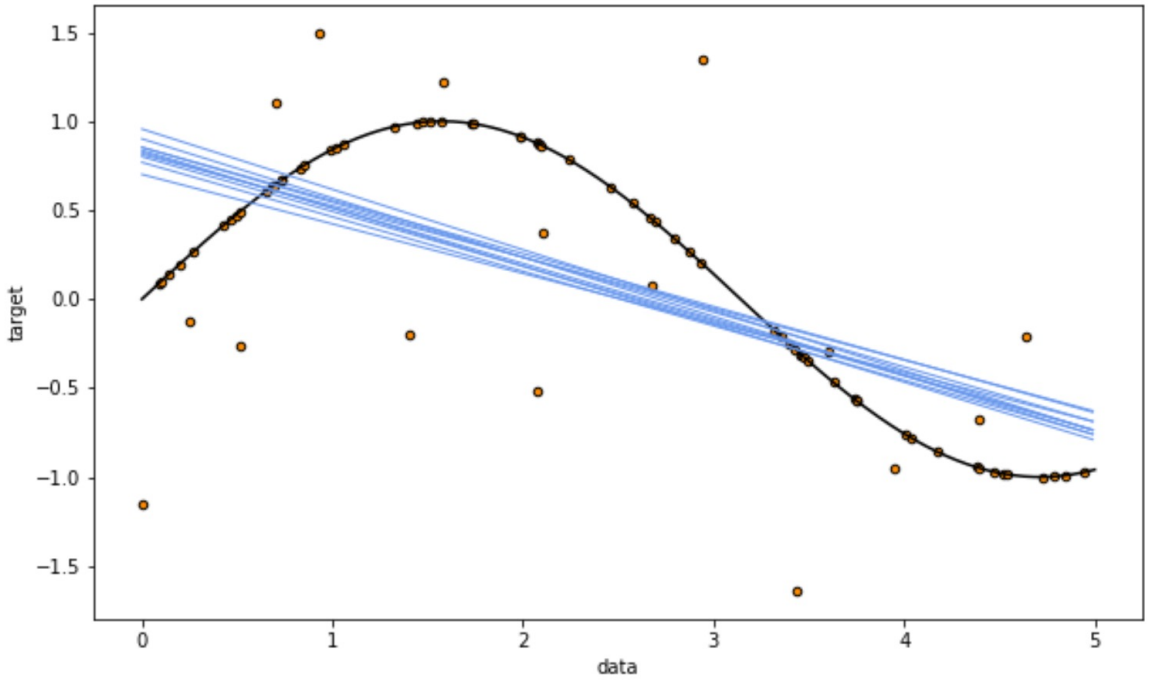
Bias-Variance Trade-Off



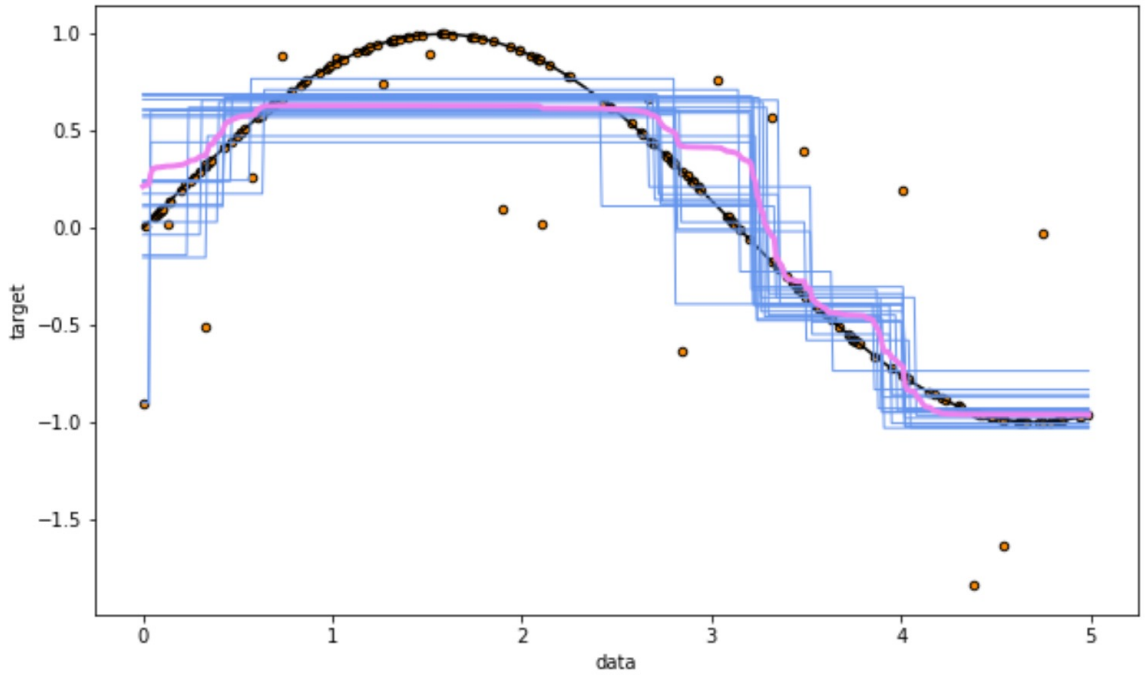
Bias-Variance Trade-Off



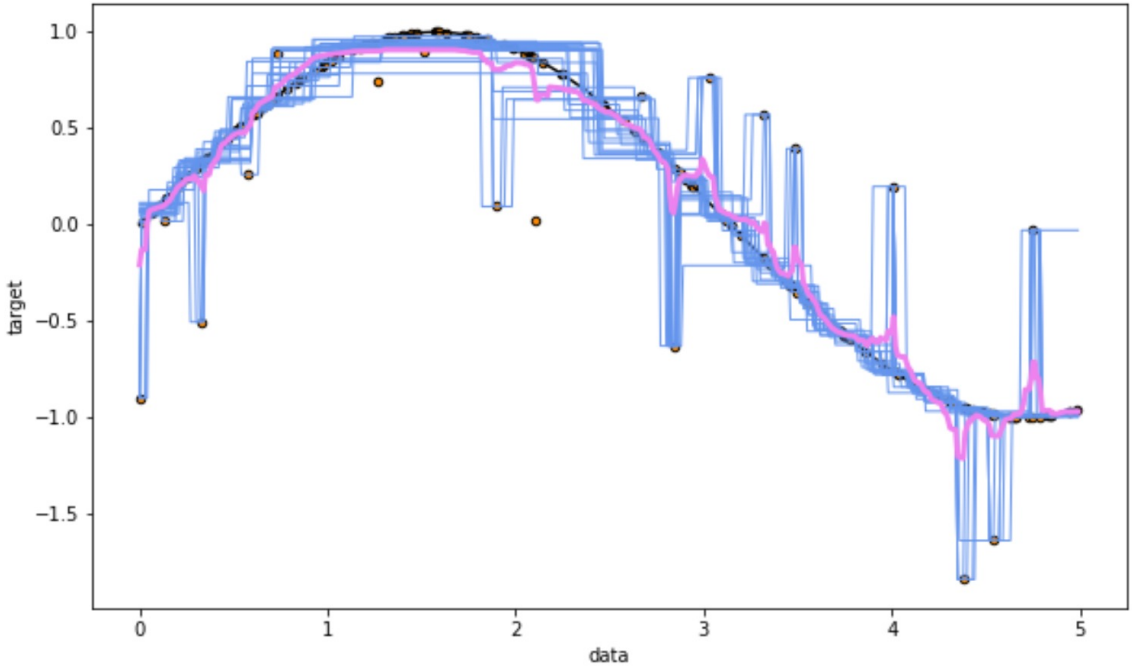
Linear Model



Trees



Trees



Bias-Variance Trade-Off

Q. Your company wants to predict whether existing automotive insurance customers are more likely to buy homeowners insurance. It created a model to better predict the best customers contact about homeowners insurance, and the model had a low variance but high bias. What does that say about the data model?

- **It was consistently wrong.**
- It was inconsistently wrong.
- It was consistently right.
- It was equally right and wrong.

Bias-Variance Trade-Off

Q. Your data science team wants to use the K-nearest neighbor classification algorithm. Someone on your team wants to use a K of 25. What are the challenges of this approach?

- Higher K values will produce noisy data.
- Higher K values lower the bias but increase the variance.
- Higher K values need a larger training set.
- **Higher K values lower the variance but increase the bias.**

Bias-Variance Trade-Off

Q. You are using K-nearest neighbor and you have a K of 1. What are you likely to see when you train the model?

- **high variance and low bias**
- low bias and low variance
- low variance and high bias
- high bias and high variance

Bias-Variance Trade-Off

Q. What does it mean to underfit your data model?

- There is too little data in your training set.
- There is too much data in your training set.
- **There is not a lot of variance but there is a high bias.**
- Your model has low bias but high variance.

Underfitted data models usually have high bias and low variance. Overfitted data models have low bias and high variance.

Bias-Variance Trade-Off

Q. The data in your model has low bias and low variance. How would you expect the data points to be grouped together on the diagram?

- **They would be grouped tightly together in the predicted outcome.**
- They would be grouped tightly together but far from the predicted.
- They would be scattered around the predict outcome.
- They would be scattered far away from the predicted outcome.

Bias-Variance Trade-Off

Q. In supervised machine learning, data scientist often have the challenge of balancing between underfitting or overfitting their data model. They often have to adjust the training set to make better predictions. What is this balance called?

- the under/over challenge
- balance between clustering classification
- **bias-variance trade-off**
- the multiclass training set challenge

Bias-Variance Trade-Off

Q. What is the best definition for bias in your data model?

- Bias is when your predicted values are scattered.
- **Bias is the gap between your predicted value and the outcome.**
- Bias is when your data is wrong for different reasons.
- Bias is when your values are always off by the same percentage.

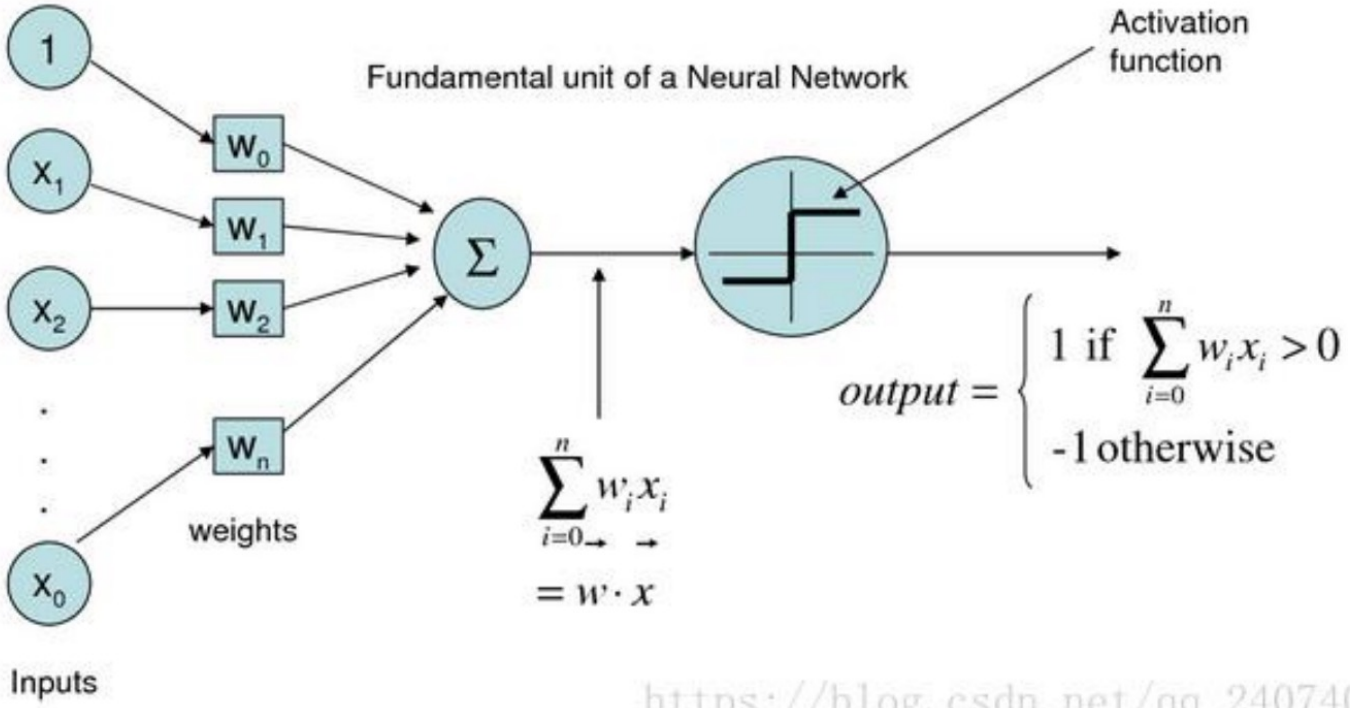
Bias-Variance Trade-Off

Q. Averaging the output of multiple decision trees helps to:

- Increase variance
- Increase bias
- **Decrease variance**
- Decrease bias

Classification Questions

What is a perceptron?



Naive Bayes

X C

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Naïve Bayes

- What is the probability of the game taking place while it is sunny?
- $P(x|c)$ = P (x=Sunny|c=Yes) = 3 / 9 = 0.33
- P (c) = P(c=Yes) = 9 / 14 = 0.64
- P (x) = P(x=Sunny) = 5 / 14 = 0.36
- P(c|x) = P(c=Yes|x=Sunny) = 0.33 * 0.64 / 0.36 = 0.6

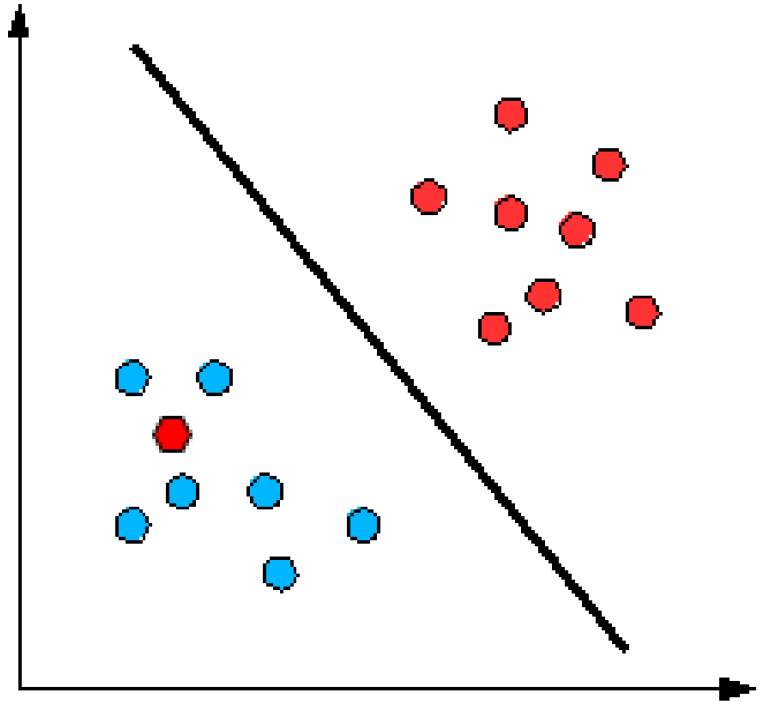
Why Naive Bayes is called Naive?

We call it naive because its assumptions (it assumes that all of the features in the dataset are equally important and independent) are really optimistic and rarely true in most real-world applications:

- We consider that these predictors are independent;
- We consider that all the predictors have an equal effect on the outcome (like the day being windy does not have more importance in deciding to play golf or not)

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

What is a Decision Boundary?



What types of Classification Algorithms do you know?

- **Logistic regression**: ideally used for classification of *binary* variables. Implements the *sigmoid function* to calculate the probability that a data point belongs to a certain class.
- **K-Nearest Neighbours (kNN)**: calculate the distance of one data point from every other data point and then takes a majority vote from *k-nearest neighbors* of each data points to classify the output.
- **Decision trees**: use multiple *if-else statements* in the form of a tree structure that includes *nodes* and *leaves*. The nodes breaking down the one major structure into smaller structures and eventually providing the final outcome.

What types of Classification Algorithms do you know?

- **Random Forest**: uses multiple *decision trees* to predict the outcome of the target variable. Each decision tree provides its own outcome and then it takes the majority vote to classify the final outcome.
- **Support Vector Machines**: it creates an *n-dimensional space* for the *n number of features* in the dataset and then tries to create the hyperplanes such that it divides and classifies the data points with the maximum margin possible.

What is the difference between KNN classification and K-means Clustering?

- ***K-nearest neighbors*** or *KNN* is a *supervised classification (or regression) algorithm*. This means that we need labeled data to classify an unlabeled data point. It attempts to classify a data point based on its proximity to other K-data points in the feature space.
- ***K-means Clustering*** is an *unsupervised clustering algorithm*. It requires only a set of unlabeled points and a threshold K, so it gathers and groups data into K number of clusters.

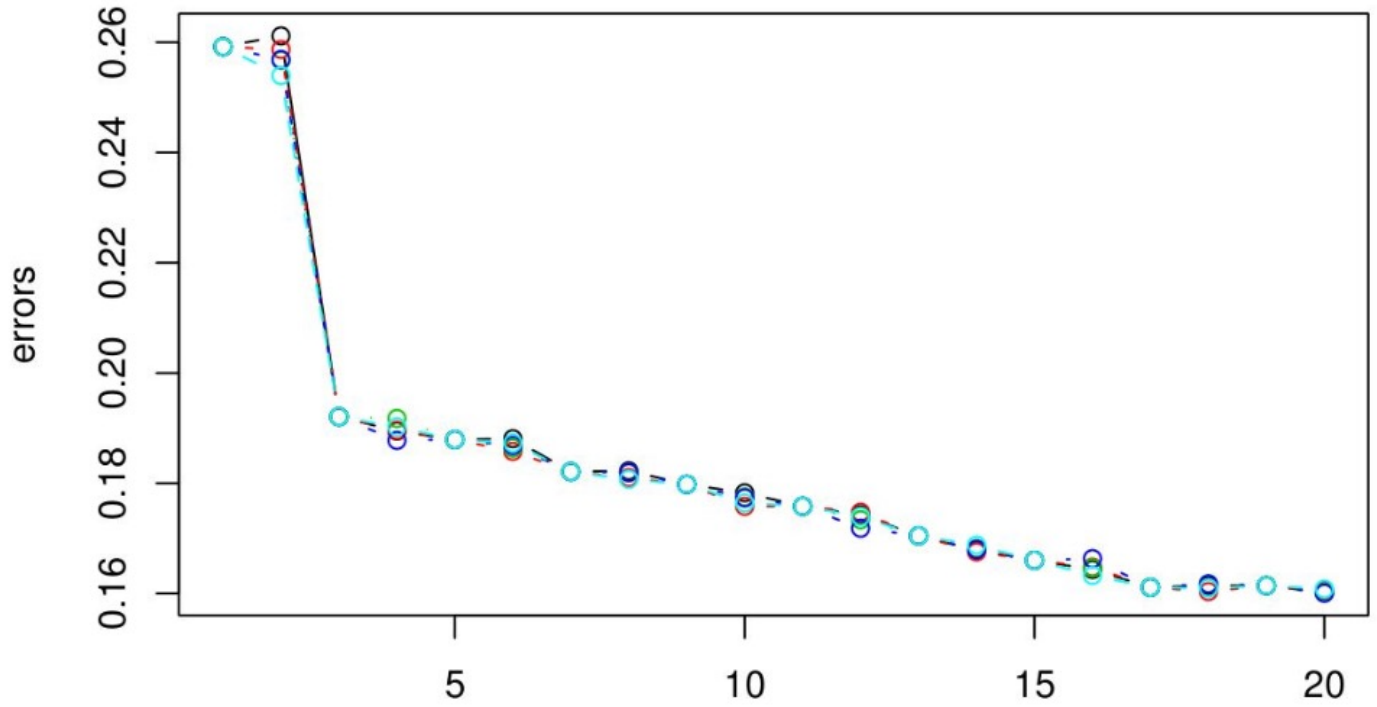
How do you choose the optimal k in k-NN?

There is not a rule of thumb to choose a standard optimal ***k***. This value depends and varies from dataset to dataset, but as a general rule, the main goal is to keep it:

- small enough to exclude the samples of the other classes but
- large enough to minimize any noise in the data.

A way to looking for this optimal parameter, commonly called the *Elbow method*, consist in creating a *for loop* that trains various ***KNN*** models with different ***k values***, keeping track of the error for each of these models, and use the model with the ***k value*** that achieves the best accuracy.

How do you choose the optimal k in k-NN?

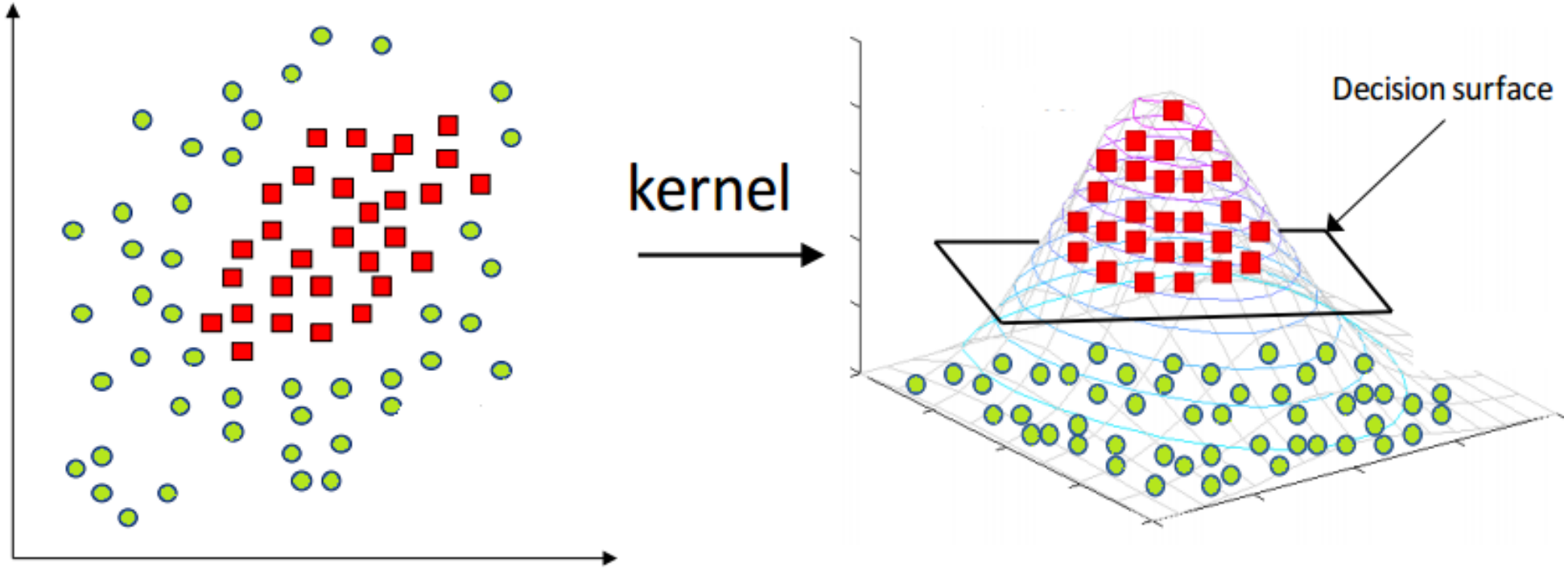


Why would you use the Kernel Trick?

When it comes to **classification** problems, the goal is to establish a decision boundary that maximizes the margin between the classes. However, in the real world, this task can become difficult when we have to treat with **non-linearly separable data**. One approach to solve this problem is to perform a data transformation process, in which we map all the data points to a **higher dimension** find the boundary and make the classification.

That sounds alright, however, when there are more and more dimensions, computations within that space become more and more expensive. In such cases, the **kernel trick allows us to operate in the original feature space without computing the coordinates of the data** in a higher-dimensional space and therefore offers a more efficient and less expensive way to transform data into higher dimensions.

Why would you use the Kernel Trick?



Why would you use the Kernel Trick?

There exist different kernel functions, such as:

- *linear*,
- *nonlinear*,
- *polynomial*,
- *radial basis function (RBF)*, and
- *sigmoid*.

Each one of them can be suitable for a particular problem depending on the data.

What's the difference between Multiclass Classification models and Multi Label model?

Multiclass classification problems:

- A *single example* is assigned *exactly one label* from a group of many possible classes.
- For example, if our model is classifying images as *cats*, *dogs*, or *rabbits*, the *softmax output* might look like this for a given image: [.89, .02, .09]. This means our model is predicting an 89% chance the image is a *cat*, 2% chance it's a *dog*, and 9% chance it's a *rabbit*. Because each image can have only one possible label in this scenario, we can take the **argmax** (index of the highest probability) to determine our model's predicted class.

What's the difference between Multiclass Classification models and Multi Label model?

Multilabel models:

- Refers to problems where we can assign *more than one label* to a *given training example*.
- For example, in text models, we can imagine a few scenarios where text can be labeled with multiple tags. Suppose that we have a dataset of Stack Overflow questions, we could build a model to predict the tags associated with a particular question. As an example, the question "*How do I plot a pandas DataFrame?*" could be tagged as "*Python*", "*pandas*", and "*visualization*".

When Logistic Regression can be used?

Logistic regression can be used in *classification* problems where the output or dependent variable is *categorical* or *binary*. However, in order to implement logistic regression correctly, the dataset must also satisfy the following properties:

1. There should not be a high correlation between the independent variables. In other words, the predictor variables should be independent of each other.
2. There should be a linear relationship between the logit of the outcome and each predictor variable. The logit function is given as logit(p) = log(p/(1-p)), where p is the probability of the outcome.

When all the requirements above are satisfied, logistic regression can be used.

How would you make a prediction using a Logistic Regression model?

In **Logistic regression** models, we are modeling the *probability* that an input (X) belongs to the default class (Y=1), that is to say:

$$P(X) = P(Y = 1|X)$$

where the P(X) values are given by the ***logistic function***,

$$P(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

The β0 and β1 values are estimated during the training stage using *maximum-likelihood* estimation or *gradient descent*. Once we have it, we can make predictions by simply putting numbers into the *logistic regression equation* and calculating a result.

How would you make a prediction using a Logistic Regression model?

For example, let's consider that we have a model that can predict whether a person is male or female based on their height, such as if $P(X) \geq 0.5$ the person is male, and if $P(X) < 0.5$ then is female.

During the training stage we obtain $\beta 0 = -100$ and $\beta 1 = 0.6$, and we want to evaluate what's the probability that a person with a height of 150cm is male, so with that intention we compute:

$$y = \frac{e^{-100+0.6\cdot 150}}{1 + e^{-100+0.6\cdot 150}} = 0.00004539\cdots$$

Given that logistic regression solves a *classification* task, we can use directly this value to predict that the person is a female.

Case study

ML system design intro

1. Problem statement
2. KPI and limitations
3. Baseline (heuristics or a simple model)
4. Data (what features, EDA, preproccessing)
5. Metrics and validation
6. Modelling
7. Problems
8. Deployment

Questions

You work for a large telecom company that provides cellular services. You need to measure customer churn.

- How to define churn?
- What data will be useful to you?
- How to evaluate the quality of models?
- What models can be used?
- How to present the results of your research to the management (what graphs will you draw, how will you summarize)?