Polynomial Regression and Overfitting

Machine Learning 2022-2023

Lab 4

Agenda

- Polynomial Regression
- Overfitting Vs Generalization
- Train, Test, Validation and Cross Validation
- Feature Selection
- Regularization for Linear Regression
- Assignment

Polynomial Regression

The linear features in : $Y = \theta_0 + \theta_1 x$

Can be transformed to Polynomial : $Y = \theta_0 + \theta_1 x + \theta_2 x^2$

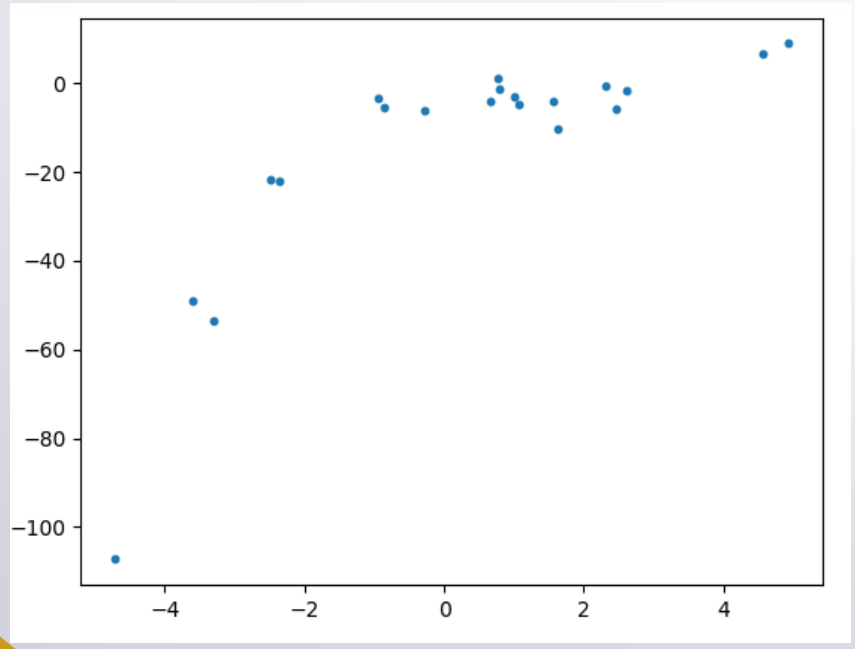
This is still considered to be **linear model** as the coefficients/weights associated with the features are still linear. x^2 is only a feature. However the curve that we are fitting is **quadratic** in nature.

Polynomial Regression

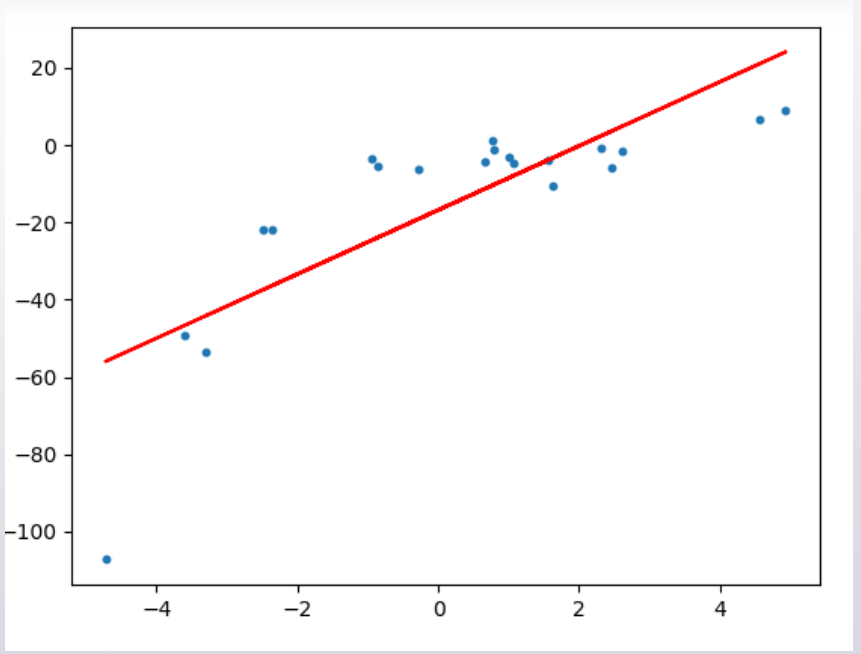
Why Polynomial Regression?

To understand the need for polynomial regression, let's generate some random points first.

Original Points



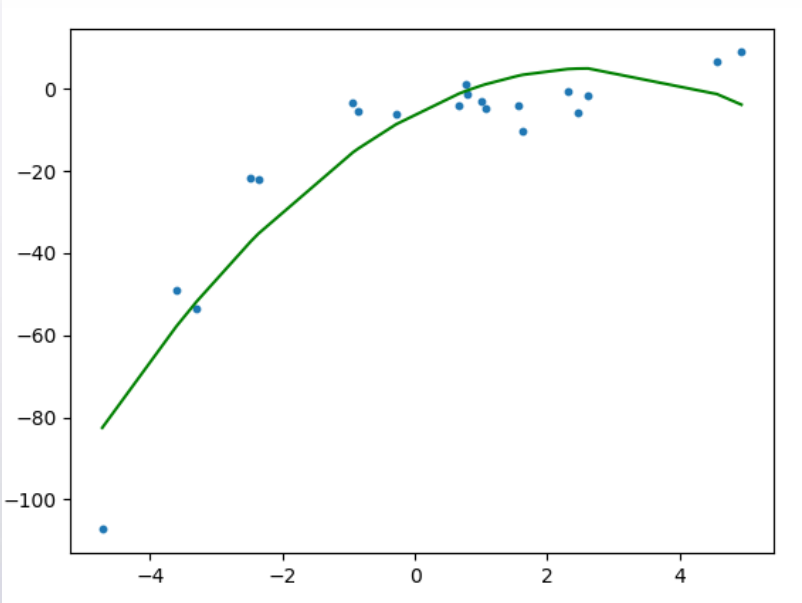
After applying a linear regression model



Polynomial Regression

To convert the original features into their higher order terms we will use the **Polynomial Features** class provided by **scikit-learn**.

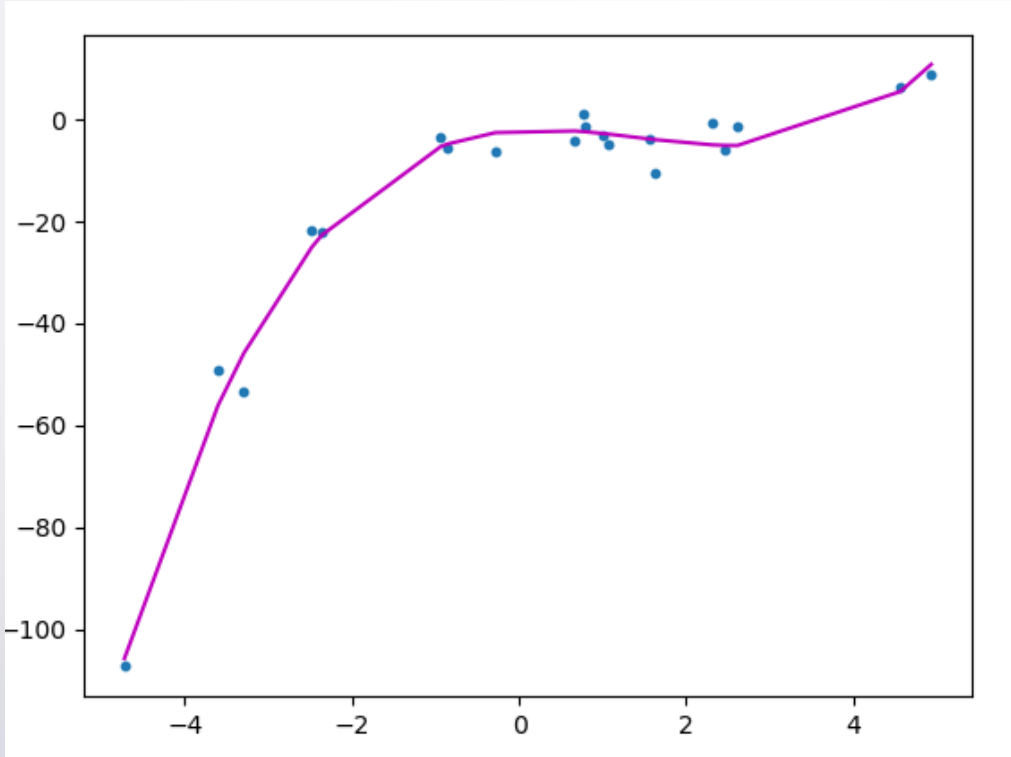
if an input sample is two dimensional and of the form [a, b], the degree-2 polynomial features will be [1, a, b, a^2, ab, b^2].



It is quite clear from the plot that the quadratic curve is able to fit the data better than the linear line

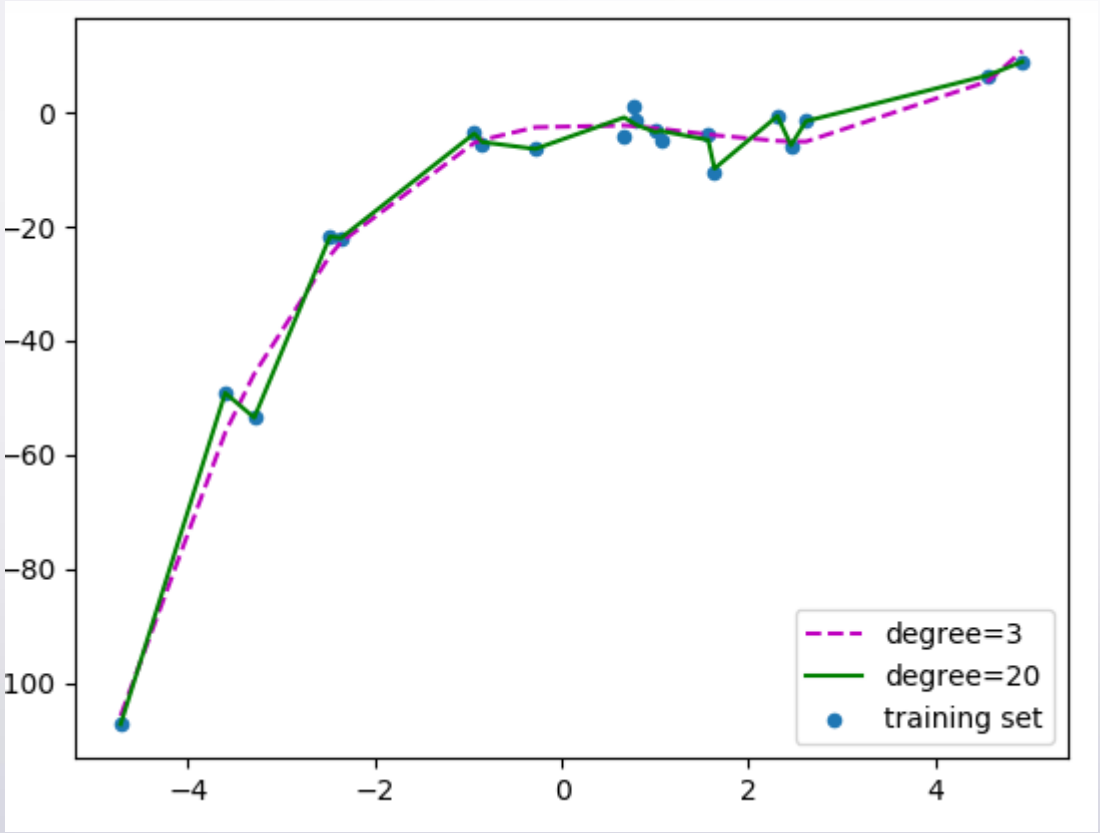
Polynomial Regression

If we try to fit a cubic curve (degree=3) to the random points, we can see that it passes through more data points than the quadratic and the linear plots.



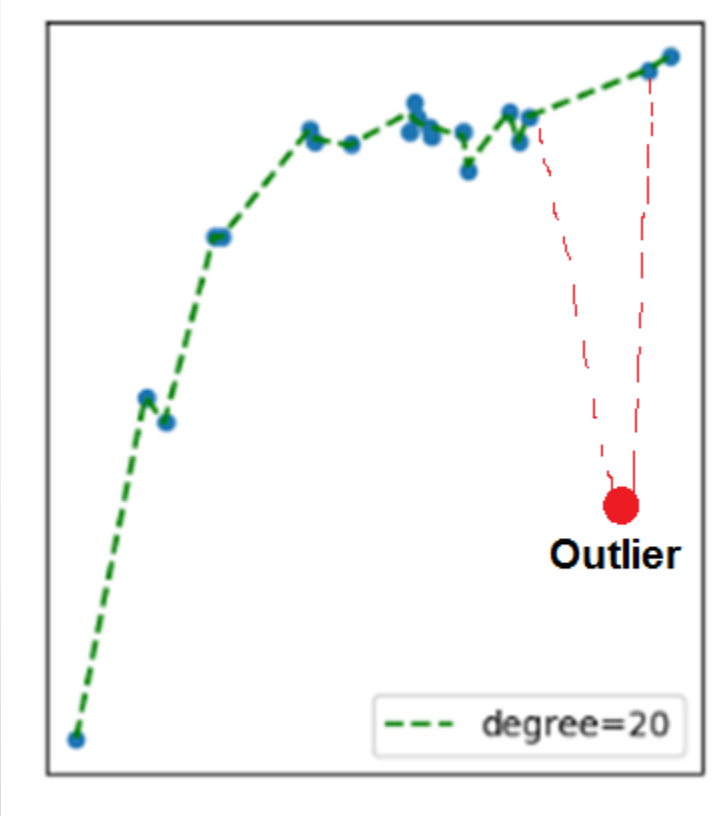
Polynomial Regression

If we further increase the degree to 20, we can see that the curve passes through more data points.



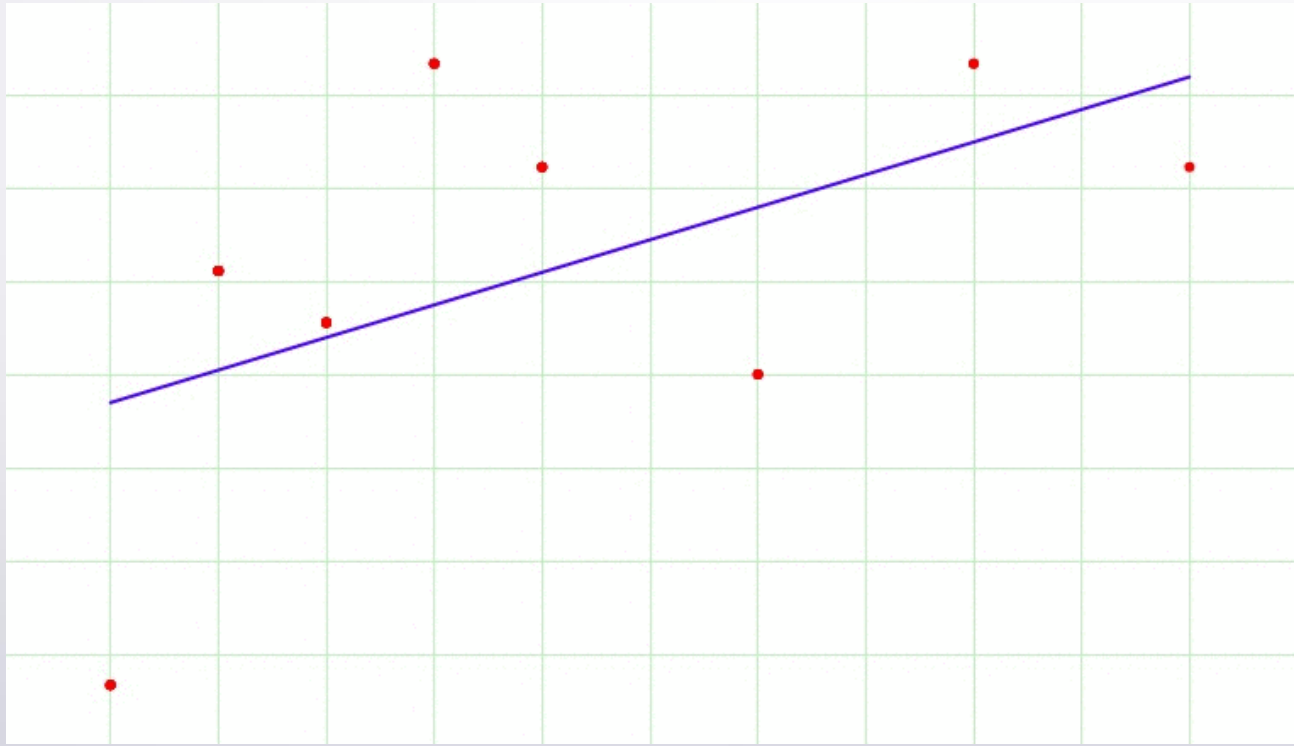
Polynomial Regression

For degree=20, the model is also capturing the noise in the data. This is an example of **over-fitting**. Even though this model passes through most of the data, it will fail to generalize on unseen data.



Overfitting

Overfitting happens when model learns signal as well as noise in the training data and wouldn't perform well on new data on which model wasn't trained on. In the example below, you can see underfitting in first few steps and overfitting in last few.



Our model doesn't *generalize* well from our training data to unseen data.
This is known as overfitting, and it's a common problem in machine learning and data science.

Overfitting

How to Prevent Overfitting

Detecting overfitting is useful, but it doesn’t solve the problem. Fortunately, you have several options to try.

Here are a few of the most popular solutions for overfitting:

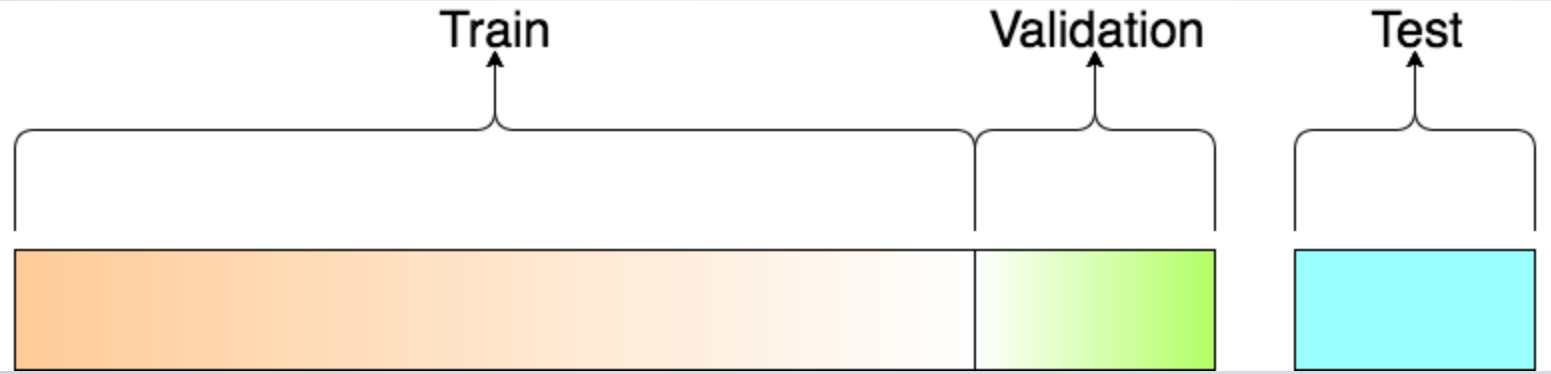
- **Cross-validation**
- **Train with more data**
- **Remove features**
- **Regularization**
- **Early stopping**

Train, Test and Validation Sets

Training Dataset: The sample of data used to fit the model.

Validation Dataset: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

Test Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

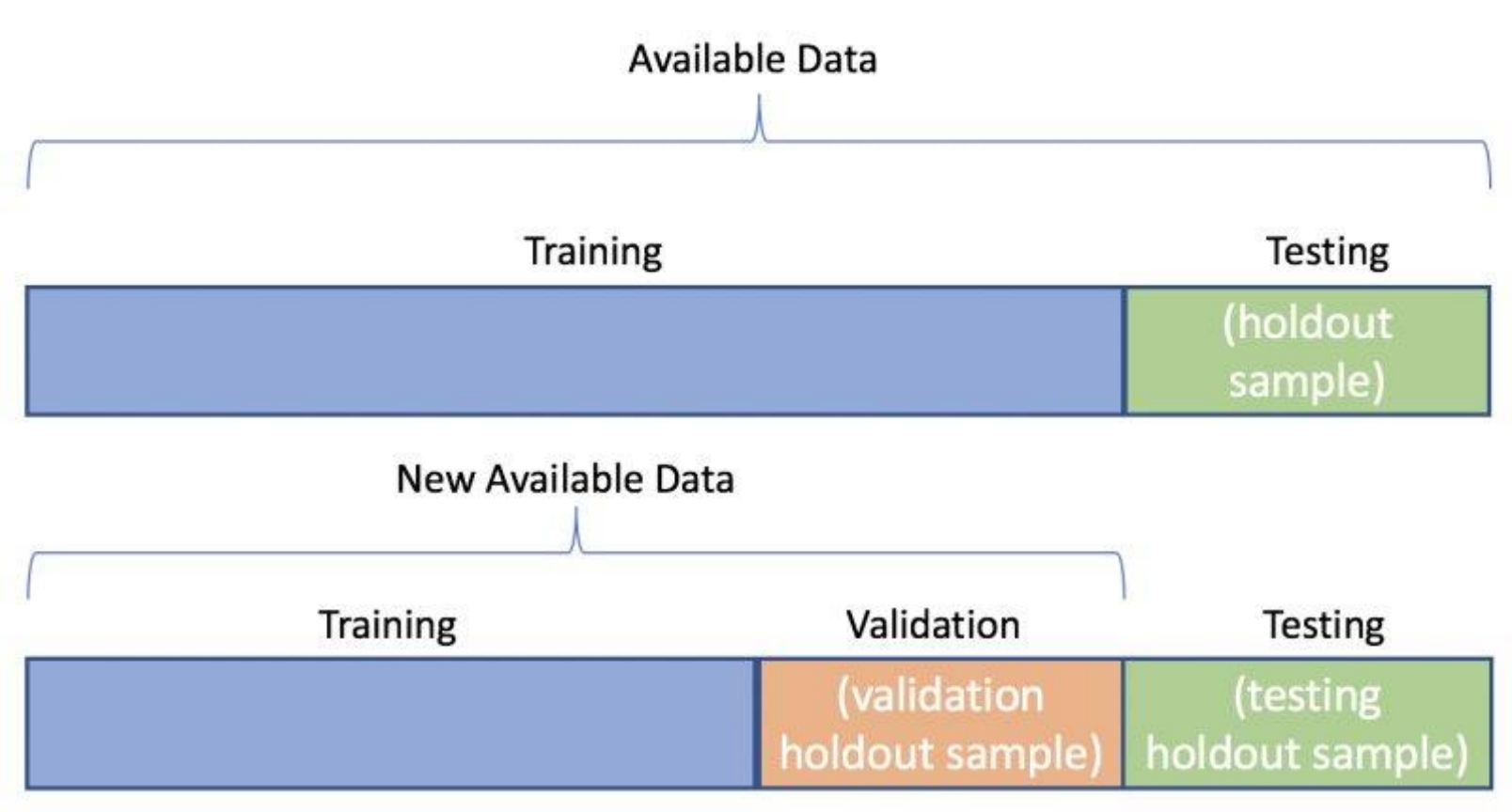


Dataset Splits

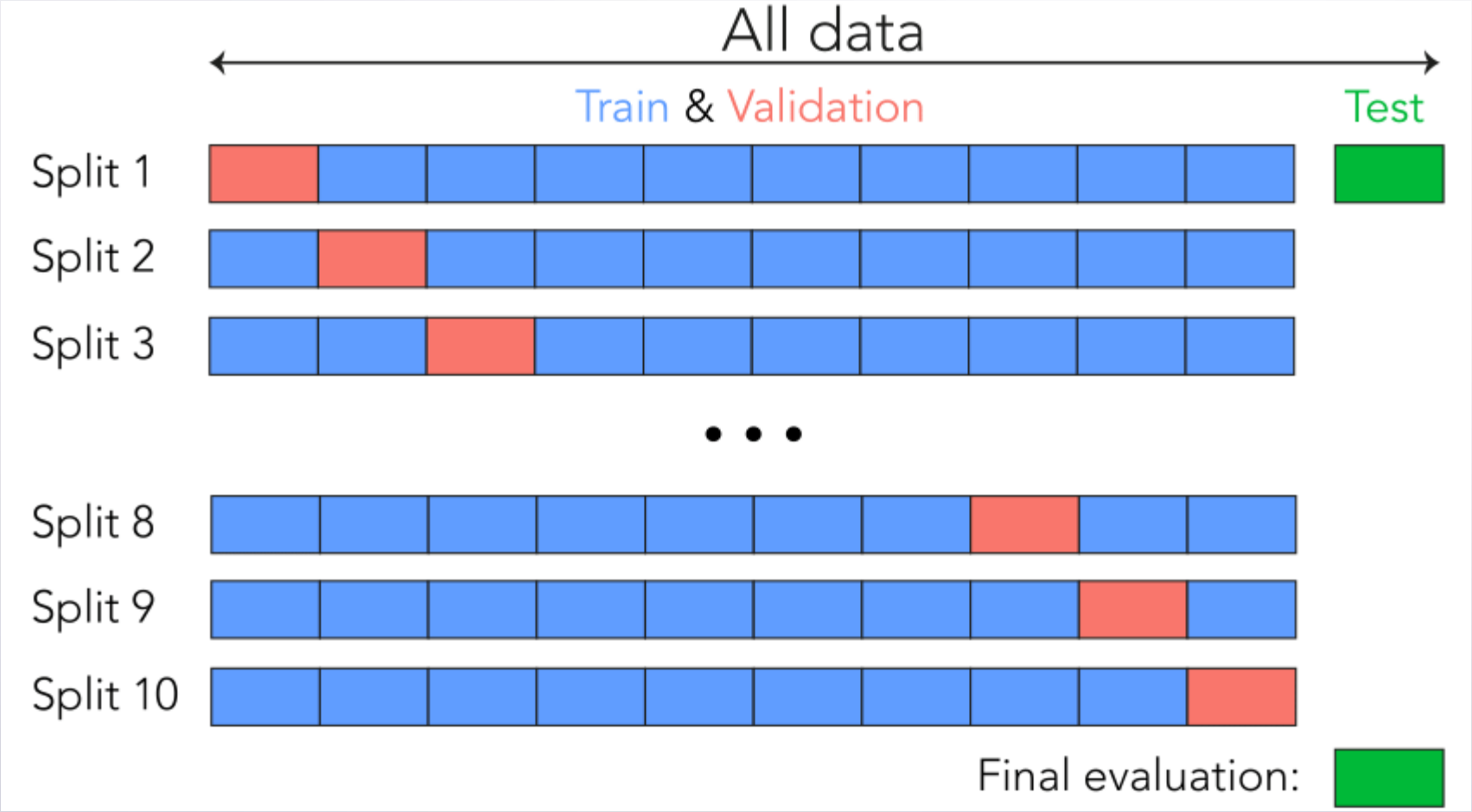
In splitting the data, we have 3 options:

1. Simply split into train and test: But that way tuning a hyperparameter makes the model 'see' the test data (i.e. knowledge of test data leaks into the model)
2. Split into train, validation, test sets: Then the validation data would eat into the training set
3. Cross-validation: Split into train and test, and train multiple models by sampling the train set. Finally, just test once on the test set.

Train, Test and Validation Sets



K-Fold Cross-Validation



Feature Selection

Data Correlation: Is a way to understand the relationship between multiple variables and attributes in your dataset. Using Correlation, you can get some insights such as:

- One or multiple attributes depend on another attribute or a cause for another attribute.
- One or multiple attributes are associated with other attributes.

Why is correlation useful?

•Correlation can help in predicting one attribute from another (Great way to impute missing values).
•Correlation can (sometimes) indicate the presence of a causal relationship.
•Correlation is used as a basic quantity for many modelling techniques

Feature Selection

Positive Correlation: means that if feature **A** increases then feature **B** also increases or if feature **A** decreases then feature **B** also decreases.

Negative Correlation: means that if feature **A** increases then feature **B** decreases and vice versa.

No Correlation: No relationship between those two attributes.

In python, **dataframe.corr()** computes the correlation of the columns and returns the correlation matrix.

Feature Selection

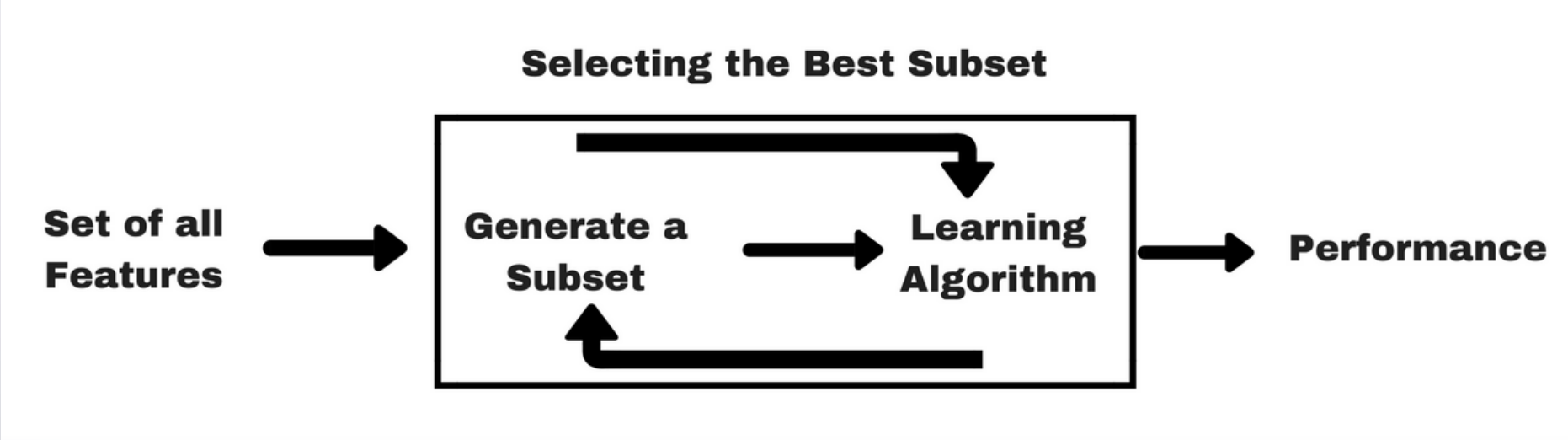
- There are four main types of feature selection methods:
  A. Filter methods
  B. Wrapper methods
  C. Embedded methods such as Lasso Regularization
  D. Hybrid methods

- Filter Methods:

- Filter Methods Examples:
  Chi-squared test, information gain, and correlation coefficient scores.

Feature Selection

- Wrapper Methods:



- Wrapper Methods Examples:
  forward feature selection, backward feature elimination, recursive feature elimination, etc

Regularization

Regularization refers to a broad range of techniques for artificially forcing your model to be simpler.

The method will depend on the type of learner you're using. For example, you could use dropout on a neural network, or add a penalty parameter to the cost function in regression.

A regression model that uses L1 regularization technique is called ***Lasso Regression*** and model which uses L2 is called ***Ridge Regression***.

The key difference between these two is the penalty term. **Ridge regression** adds "*squared magnitude*" of coefficient as penalty term to the loss function. **Lasso Regression** adds "*absolute value of magnitude*" of coefficient as penalty term to the loss function.

Regularization

$$y_i' = B0 + B1X1 + B2X2 + \cdots BpXp$$

$$y_i' = B0 + \sum_{j=1}^{p} B_j X_j$$

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - y_i')^2 + \alpha \sum_{j=1}^{p} B_j^2$$

Here, if *alpha* is **zero** then you can imagine that we didn't make any regularization. However, if *alpha* is **very large** then it will add too much weight and it will lead to under-fitting. Having said that it's important how *lambda* is chosen. This technique works very well to avoid over-fitting issue.

Linear Regression (Vectorized Form)

In multiple Linear Regression :

$$Y = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \cdots + \Theta_n x_n$$

$$\Theta = [\, \Theta_1 \; \Theta_2 \; \ldots \; \Theta_n]$$

$$X = [\, x_1 \; x_2 \; \ldots \; x_n \,]$$

$$Y = \Theta^T . X + \Theta_0 \Rightarrow (W^T . X + b)$$

Assignment

- Apply Polynomial regression from scratch to reduce the mean square error acquired on the dataset provided.
- Your code should be dynamic (allow degree=2, degree=3 and so on).
- Built-in Polynomial Regression is NOT ALLOWED
- You should apply preprocessing, feature selection and use at least a train-test split (80% train-20% test)
- Show your process in the code.
- Assignment Deadline: Friday 31/3/2023 11:59 PM
- A form will be announced for submission

Thank You