Module: Machine Learning

Live Session-2

Agenda:
sklearn pipeline
Naïve Bayes Classifier
Linear Regression
Implementation

Pre-processors

- Machine learning workflows are often composed of different parts.
- A typical pipeline consists of a pre-processing step that transforms or imputes the data, and a final predictor that predicts target values.
- The transformer objects don't have a predict method but rather a transform method that outputs a newly transformed sample matrix *X*:

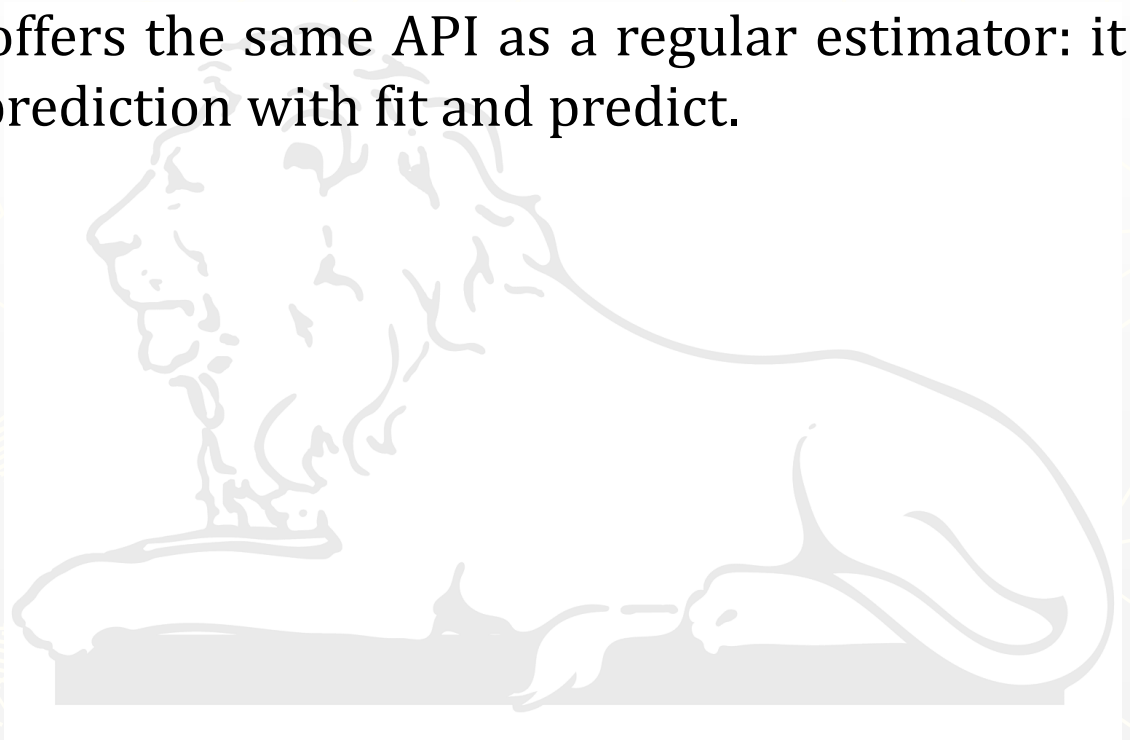
```
from sklearn.preprocessing import StandardScaler
X = [[0, 15],
...      [1, -10]]
# scale data according to computed scaling values
StandardScaler().fit(X).transform(X)
```

Processed data follows Gaussian distribution with 0 mean and unit variance

Pipelines: chaining pre-processors and estimators

Preprocessors and estimators (predictors) can be combined together into a single unifying object: a Pipeline.

The pipeline offers the same API as a regular estimator: it can be fitted and used for prediction with fit and predict.



Probability-Basics:

Random variable ≡ a numerical quantity that takes on different values depending on chance

Population ≡ the set of all possible values for a random variable

Event ≡ an outcome or set of outcomes

Probability ≡ the relative frequency of an event in the *population* … alternatively… the proportion of times an event is *expected* to occur in the long run

Conditional Probability

A **conditional probability** refers to the probability of an event *A* occurring, given that another event *B* has occurred.

Notation: P(*A* | *B*)

Read this as the "conditional probability of *A* given *B*" or the "probability of *A* given *B*."

Conditional probabilities are especially useful in economic analysis because probabilities of an event differ, depending on other events occurring.

Conditional Probability

- The conditional probability of A given B is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- The conditional probability of B given A is

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Bayes Theorem

- Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Bayes Theorem:

- Given:
  - A doctor knows that Cold causes fever 50% of the time
  - Prior probability of any patient having cold is 1/50,000
  - Prior probability of any patient having fever is 1/20

- If a patient has fever, what's the probability he/she has cold?

$$P(C \mid F) = \frac{P(F \mid C)P(C)}{P(F)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifier

Problem statement: Given features $X1, X2, \ldots, Xn$, Predict a label C

Working Procedure:
Consider each attribute and class label as random variables

Given a record with attributes $(A1, A2, \ldots, An)$

–Goal is to predict class C

–Specifically, we want to find the value of C that maximizes $P(C|\ A1, A2, \ldots, An\ )$

Naïve Bayes Classifier

- Let's say that we are interested in knowing whether an e-mail that contains the word *freeloan* (event) is spam (hypothesis). If we use the Bayes theorem description, this problem can be formulated as:

$$P(class = Spam | Contains = freeloan)$$

$$= \frac{P(Contains = freeloan | Class = Spam) * P(class = spam)}{P(contains = freeloan)}$$

Naïve Bayes Classifier

P(class=SPAM|contains="freeloan") is the probability of an e-mail being SPAM given that this e-mail contains the word freeloan. This is what we are interested in predicting.

P(contains="freeloan"|class=SPAM) is the probability of an e-mail containing the word freeloan given that this e-mail has been recognized as SPAM. This is our training data, which represents the correlation between an e-mail being considered SPAM and such e-mail containing the word freeloan.

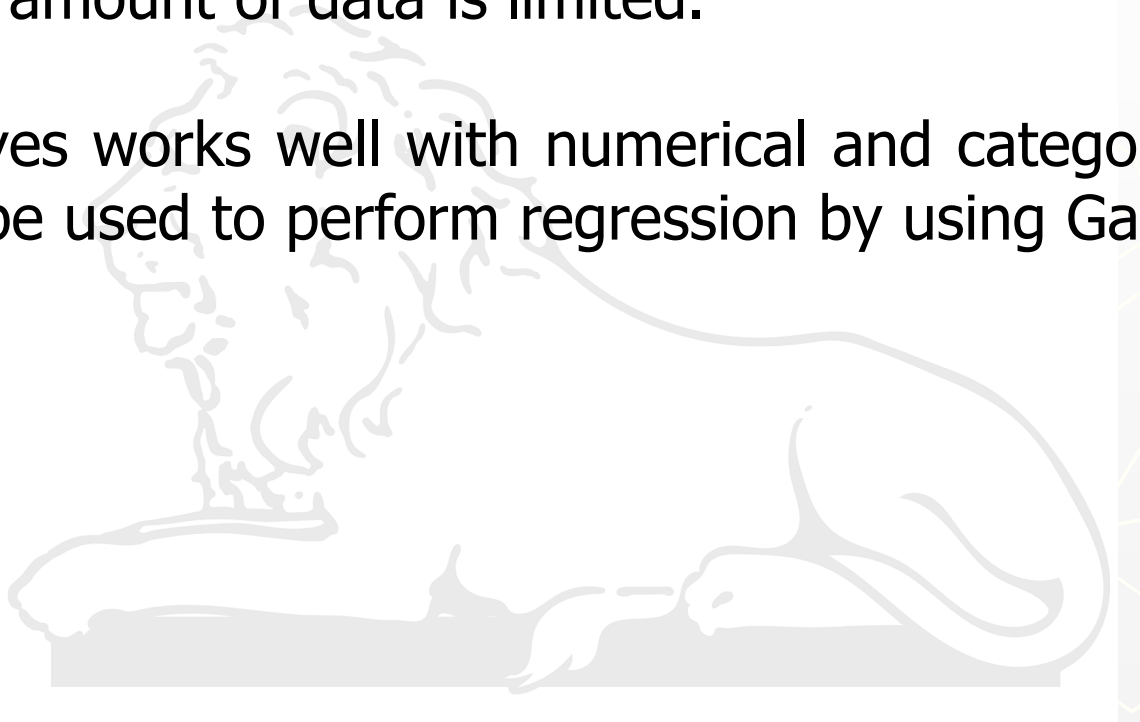
P(class=SPAM) is the probability of an e-mail being SPAM (without any prior knowledge of the words it contains). This is simply the proportion of e-mails being SPAM in our entire training set. We multiply by this value because we are interested in knowing how significant is information concerning SPAM e-mails. If this value is low, the significance of any events related to SPAM e-mails will also be low.

P(contains="freeloan") is the probability of an e-mail containing the word freeloan. This is simply the proportion of e-mails containing the word freeloan in the entire training set. We divide by this value because the more exclusive the word freeloan is, the more important is the context in which it appears.

Naïve Bayes: Pros

- Naive Bayes is a simple and easy to implement algorithm. Because of this, it might outperform more complex models when the amount of data is limited.

- Naive Bayes works well with numerical and categorical data. It can also be used to perform regression by using Gaussian Naive Bayes.



Naïve Bayes: Cons

Given the construction of the theorem, it does not work well when you are missing certain combination of values in your training data.

In other words, if you have no occurrences of a class label and a certain attribute value together (e.g. class="spam", contains="$$$") then the frequency-based probability estimate will be zero. Given Naive-Bayes' conditional independence assumption, when all the probabilities are multiplied you will get zero.

Basic Regression Model

The Regression Problem

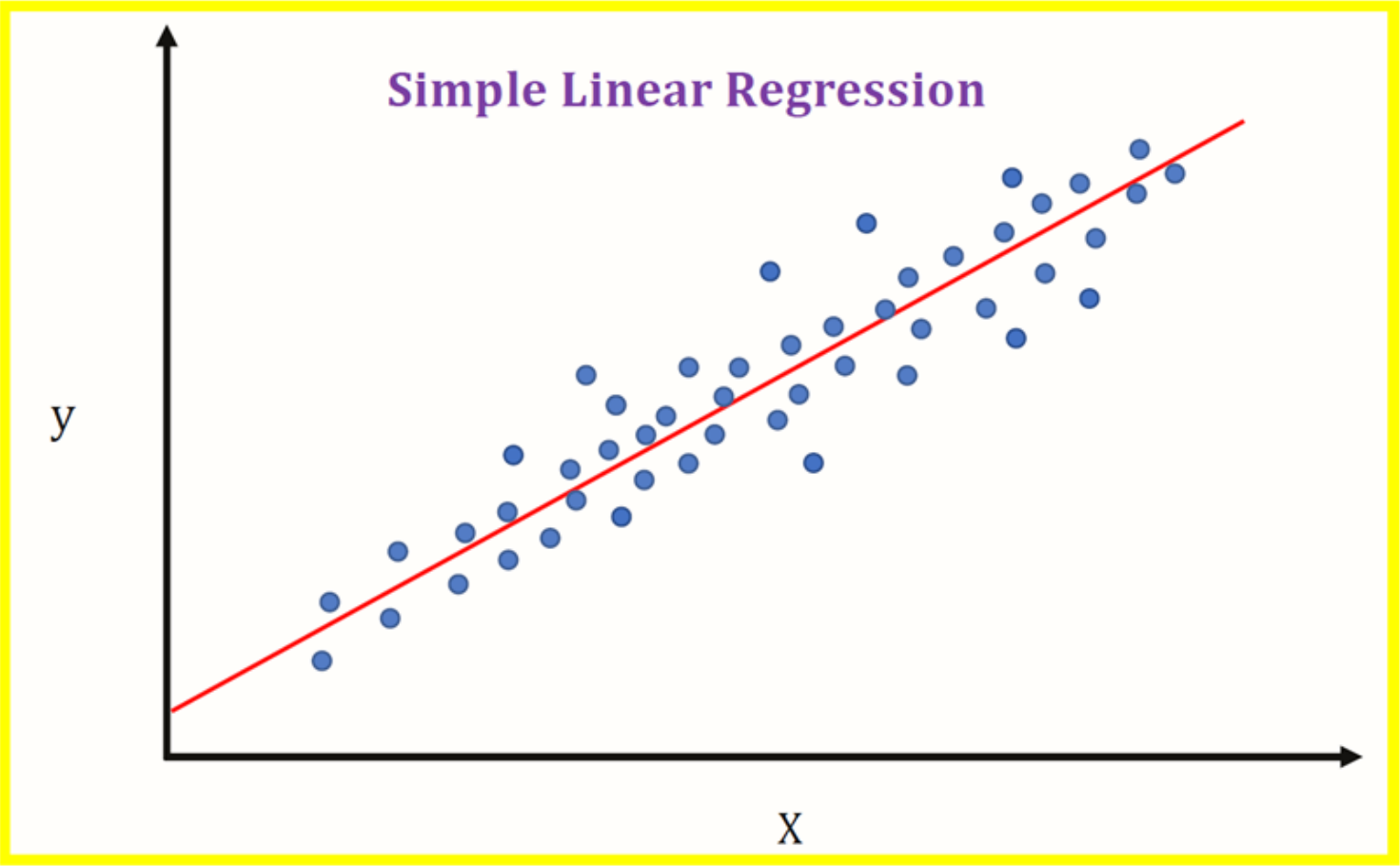
Regression refers to the problem of learning the relationships between some (qualitative or quantitative) input variables

$$X = [x_1, x_2, \dots, x_n]^T$$

and a quantitative output variable y. Means, regression is about learning a model f such that

$$y = f(X) + \epsilon$$

Simple Linear Regression



Multiple/Linear Regression Model

The linear regression model describes the output variable y (a scalar) as an affine combination of the input variables $x_1;\ x_2; \ldots; x_p$ (each a scalar) plus a noise term ϵ, i.e.

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_p x_p + \epsilon$$

where, a_i are the parameters of the regression model which we needs to estimate based on the given data.

How to learn the parameters $a_0, a_1, \ldots, a_p$ from some training dataset $T = \{X_i, y_i\}$ for $i = 1,2,..,n$. Once, we estimate these parameters, future outputs for inputs that we have not yet seen can be predicted.

Mean Squared Error (MSE):

The residual for the i^{th} observation is given by:

$$r_i = y_i - \hat{y}_i = y_i - (\hat{a}_0 + \hat{a}_1 x_{i1} + \cdots + \hat{a}_p x_{ip})$$

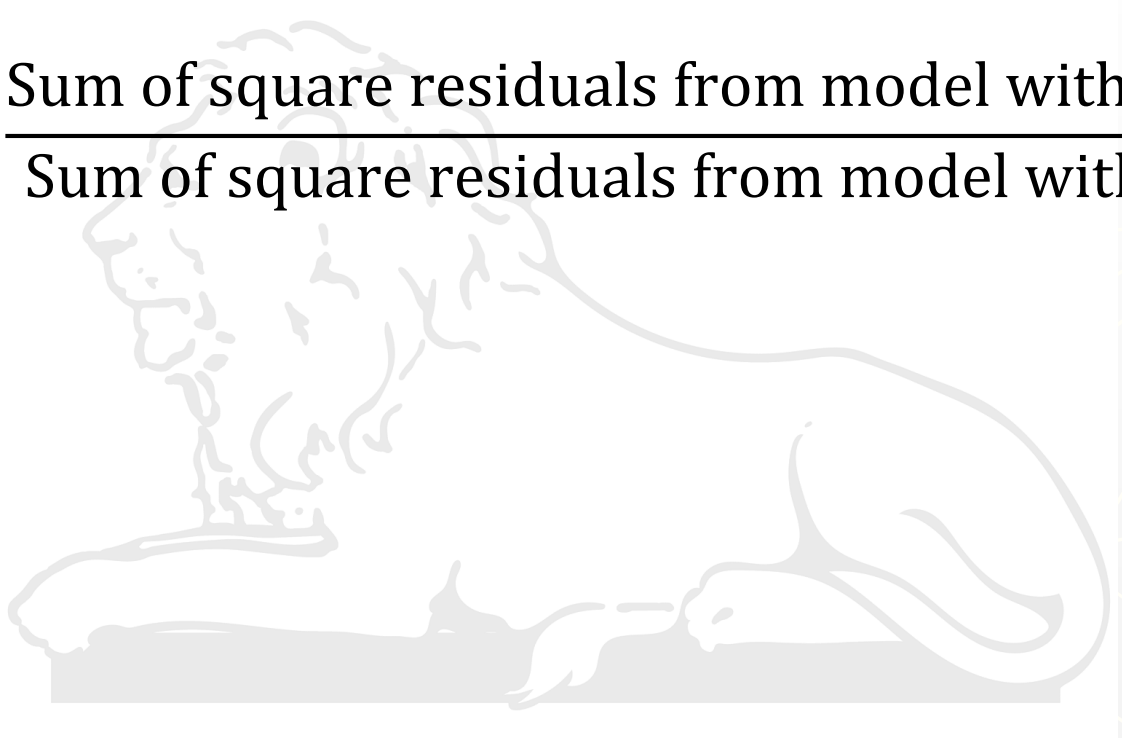
The mean squared error (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the "errors") and squaring them. The squaring is necessary to remove any negative signs.

$$MSE = \frac{(y_i - \hat{y}_i)^2}{n}$$

Goodness of fitting: Residual

Consider the linear regression models as $Y = \alpha + \beta X$ and $Y = \alpha$, then R^2 is defined as

$$R^2 = 1 - \frac{\text{Sum of square residuals from model with } \alpha \text{ and } \beta}{\text{Sum of square residuals from model with } \alpha \text{ only}}$$



Goodness of fitting

- We have $0 \leq R^2 \leq 1$
- If $SS(Res) = SS(total)$, then $R^2 = 0 \rightarrow$ model is not useful.
- If $SS(Res) = 0$, then $R^2 = 1 \rightarrow$ model fits all the points perfectly.

Essentially the same thing happens when there is more than one independent variables

How large does R^2 need to be to be considered as "good"?

This depends on the context, there is no absolute answer here. For hard to predict Y variables, smaller values may be "good".

Model Validation

One often attempts to “validate” a model either by:

- Randomly splitting an existing data set into two parts, and using part of the data for “model fitting”, and part of the data for “model validation”.
- Using one full data set for “model fitting”, and finding a second independent data set for “model validation”.

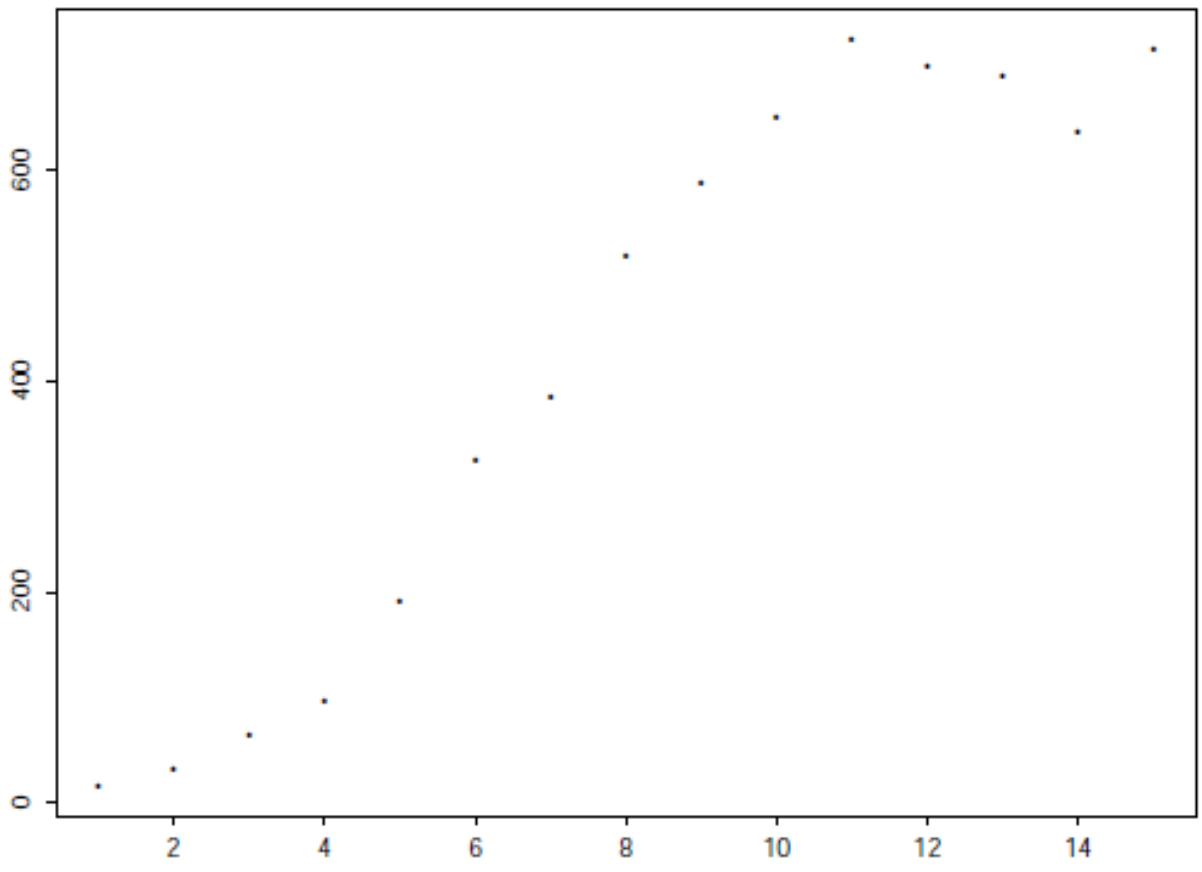
Polynomial Regression

Consider that the following table display data on the dry weight (Y) of 15 onion bulbs randomly assigned to 15 growing times (X) until measurement.

Growing Time	Dry Weight	Growing Time	Dry Weight
1	16.08	9	590.03
2	33.83	10	651.92
3	65.8	11	724.93
4	97.2	12	699.56
5	191.55	13	689.96
6	326.20	14	637.56
7	386.87	15	717.41
8	520.53		

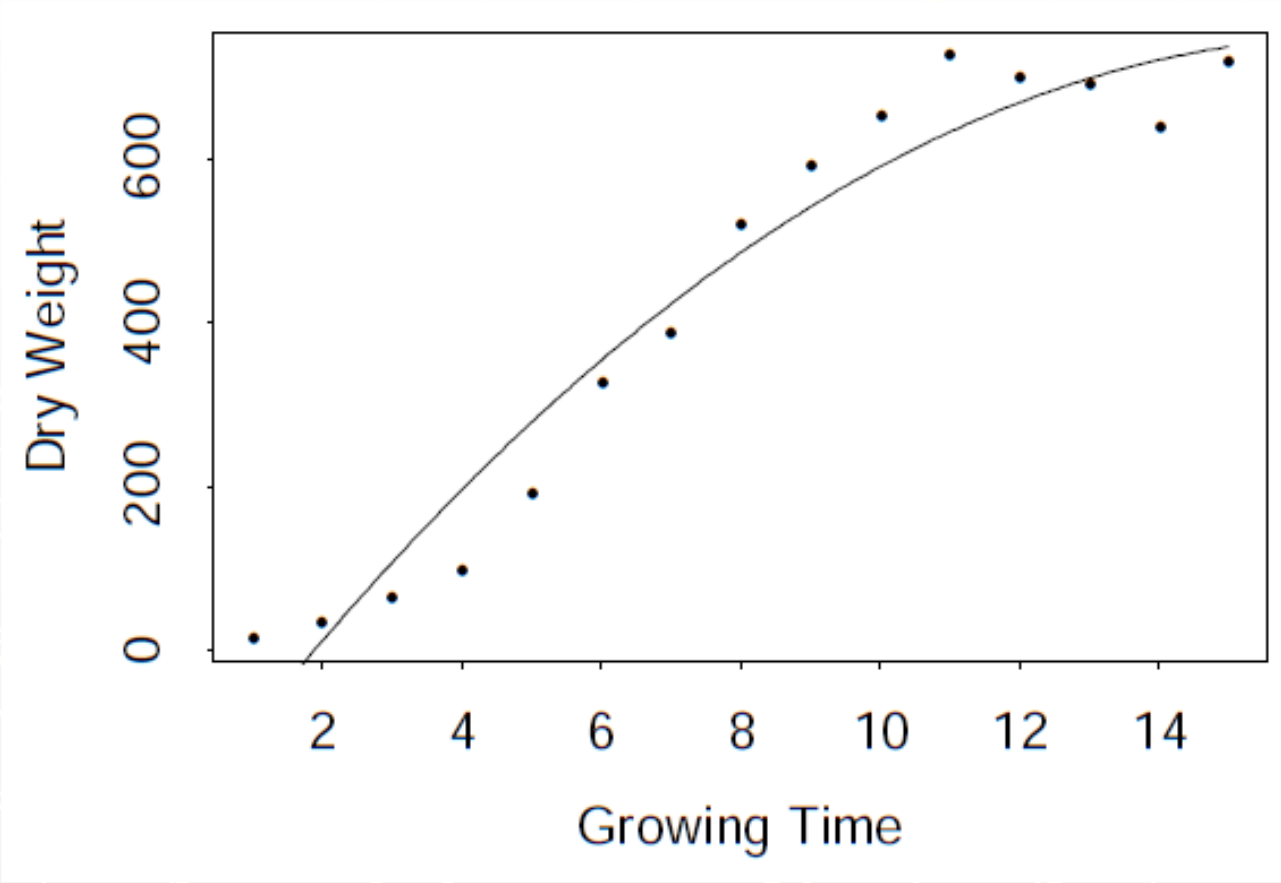
Polynomial Regression

Consider that the following scatterplot display data on the dry weight (Y) of 15 onion bulbs randomly assigned to 15 growing times (X) until measurement.



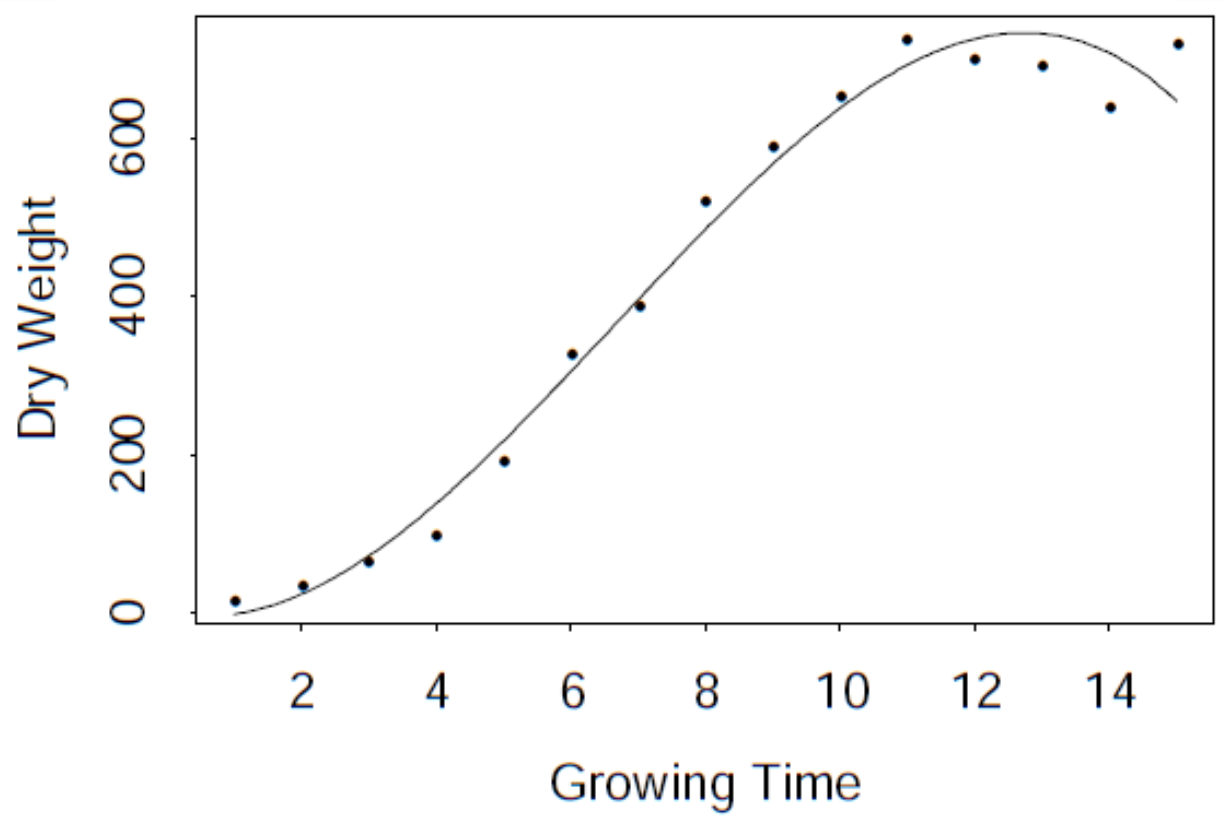
Polynomial Regression

$$Y = \alpha_0 + \alpha_1 x + \alpha_2 x^2$$



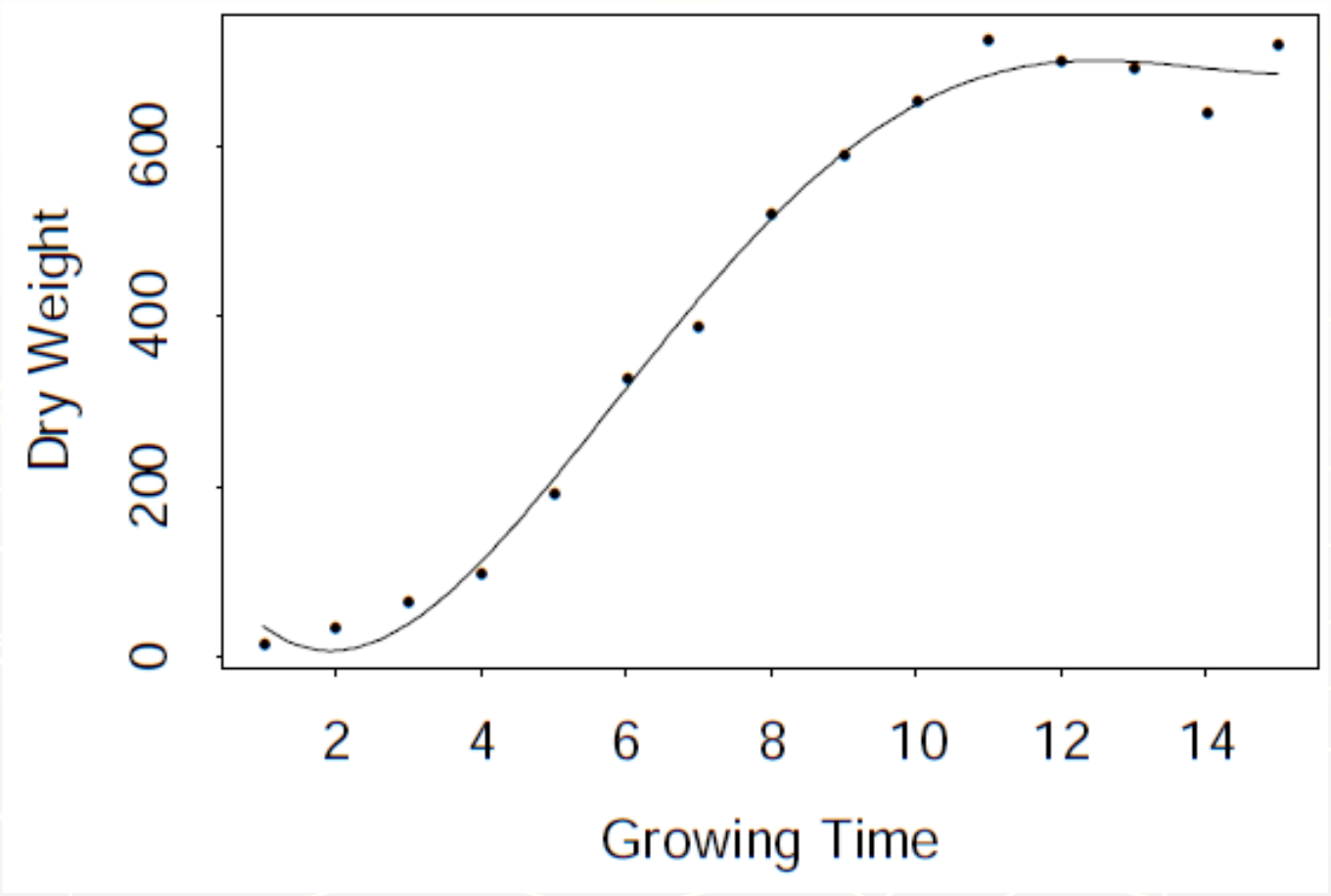
Polynomial Regression

$$Y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_4 x^4$$



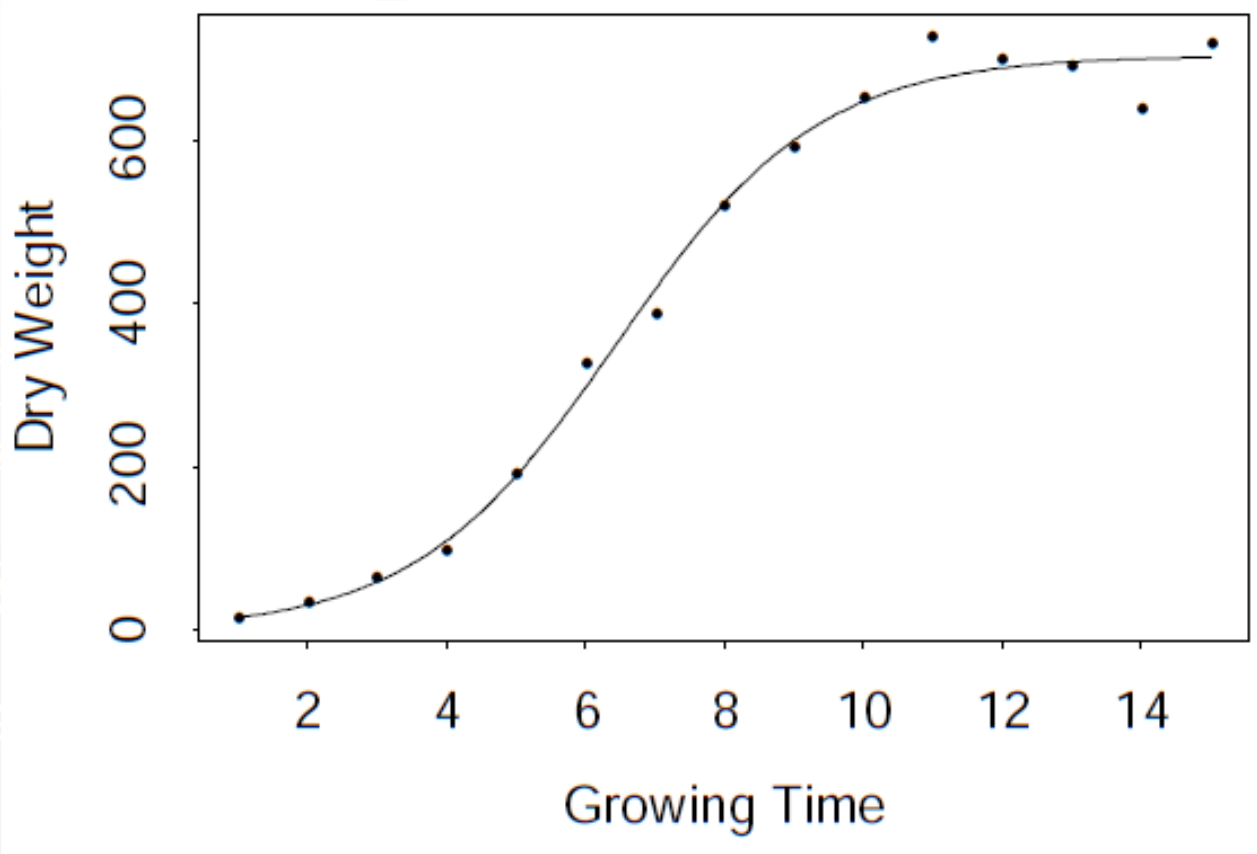
Polynomial Regression

$$Y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_4 x^4 + \alpha_5 x^5$$

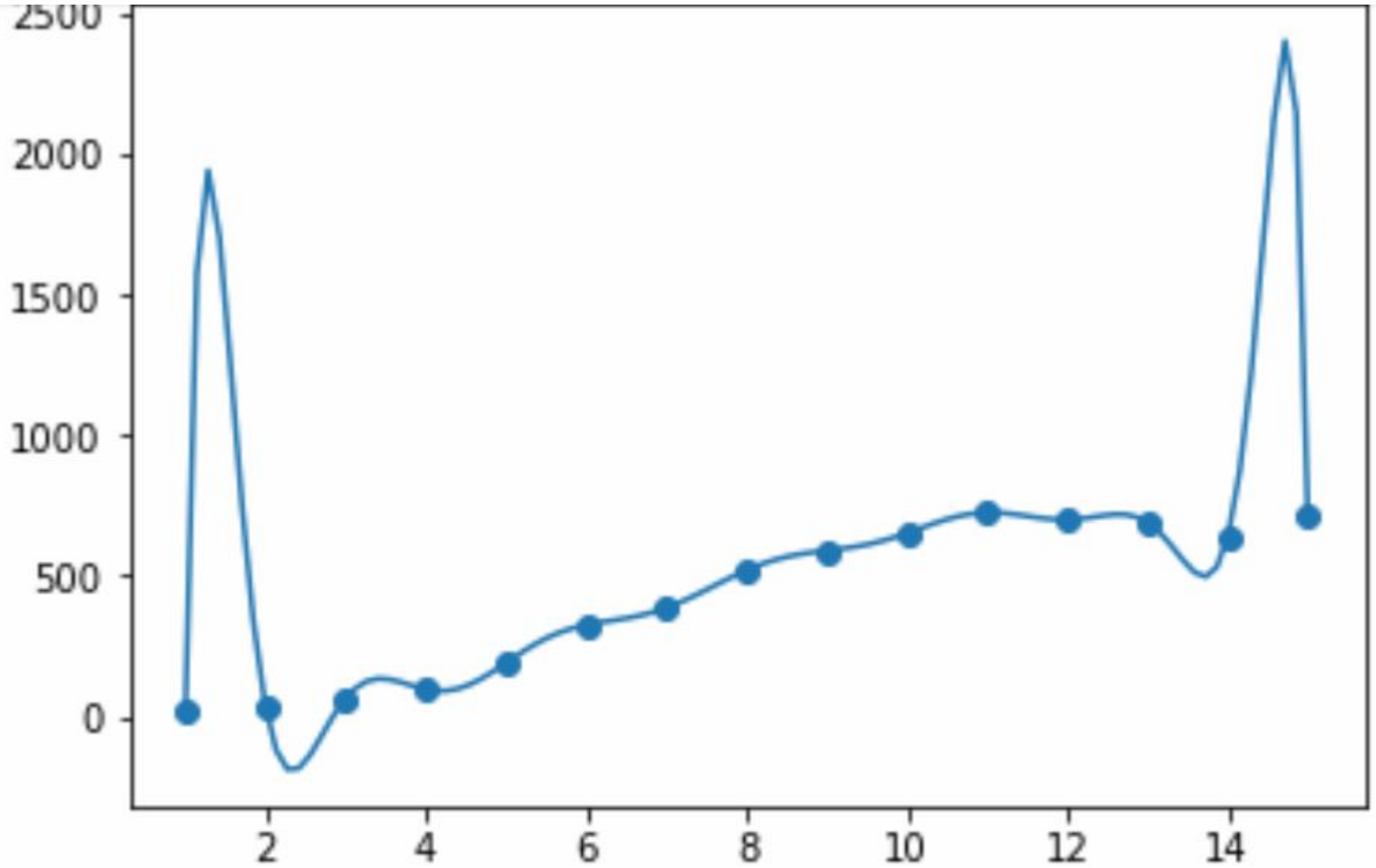


Nonlinear Regression

$$Y = \frac{\alpha_0}{1 + \exp\left\{\frac{\alpha_2 - x}{\alpha_3}\right\}}$$



Overfitting



THANK YOU

Exercise- In session

Read the iris data as per your skills in your notebook.

Pre-process your data using standard scalar functionality.

Split your data with 80% for training and 20%for testing.

Create a pipeline having Standard Scalar and Naïve Bayes Estimator

Apply fit with training data and predict with testing data.

Calculate the classification evaluation metrics.