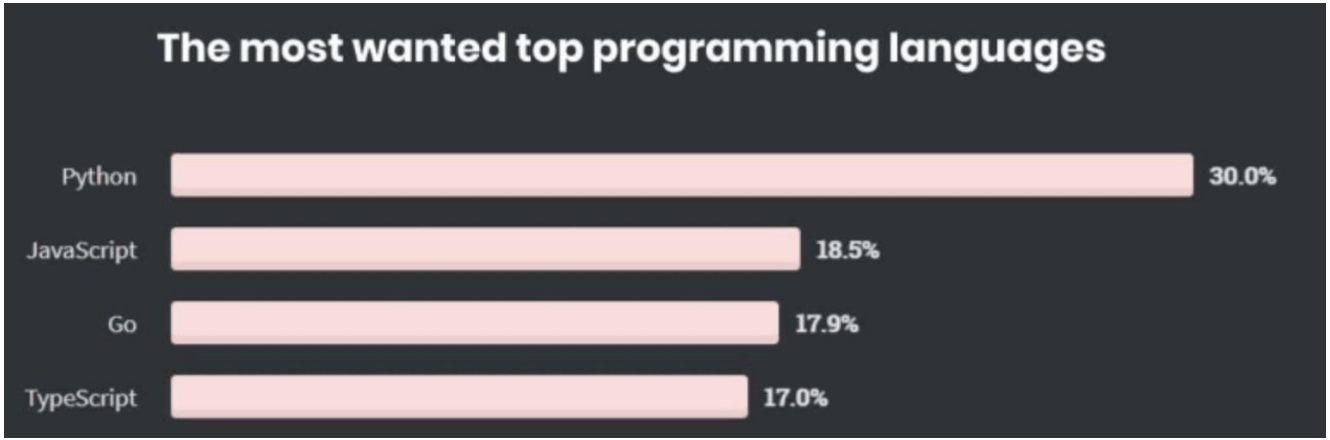
ML Coding

Pattarawit Polpinit

Coding in ML

Programming Language



Python

- Python2 VS Python3
- Learn Python:
  - CodeCademy
  - Udemy
  - Google's Python Class
  - Microsoft's Free Python Course (edX)
  - Coursera (Python for Everybody Specialization : University of Michigan)

Python IDE

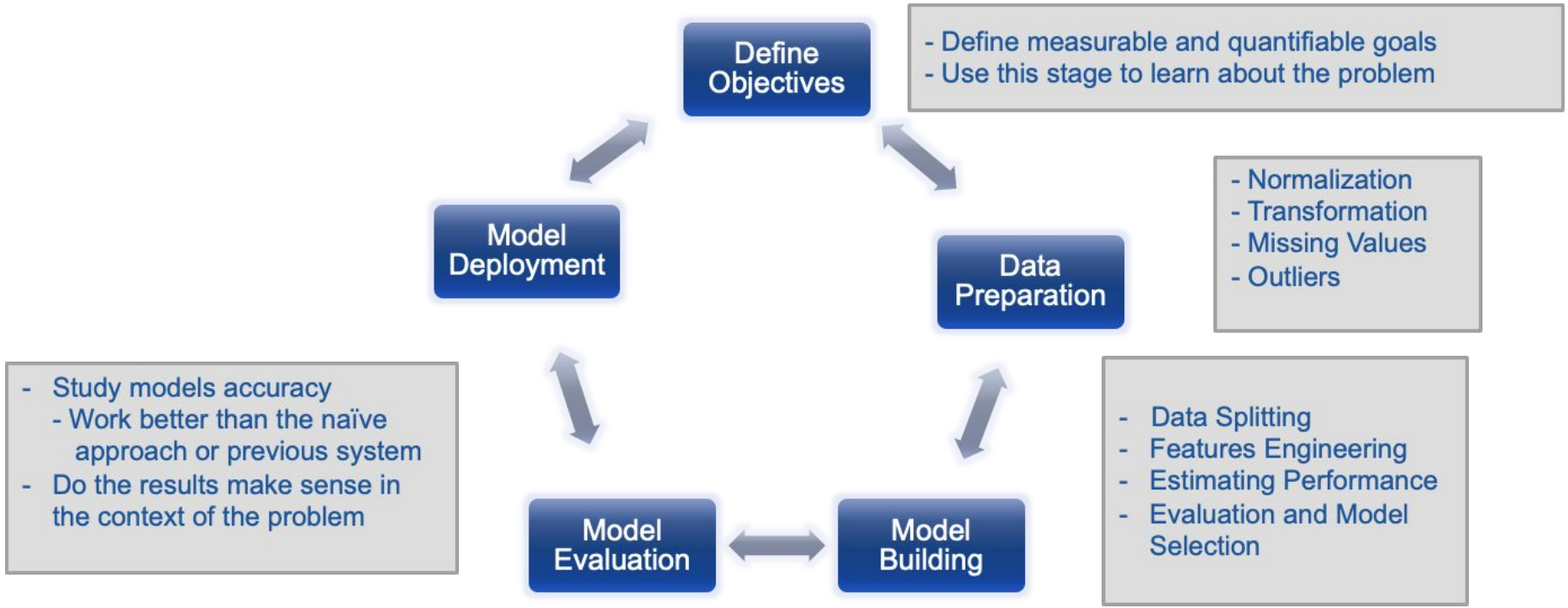
- Visual Studio code
- Spyder
- Jupyter Notebook
- Etc.
- Google Codelabs* : https://colab.research.google.com/

Google Colabs

- Tutorial : https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c
- Hello World Example
- โหลดไฟลล์ข้อมูลเพื่อแสดงผล

Resource files: https://bit.ly/3CiCMvw

ML Overview


Define Objectives
- Define measurable and quantifiable goals
- Use this stage to learn about the problem
Data Preparation
- Normalization
- Transformation
- Missing Values
- Outliers
Model Building
- Data Splitting
- Features Engineering
- Estimating Performance
- Evaluation and Model Selection
Model Evaluation
- Study models accuracy
- Work better than the naïve approach or previous system
- Do the results make sense in the context of the problem
Model Deployment

Data Preparation

Importing the library

- numpy: https://numpy.org/
- matplotlib: https://matplotlib.org/
- pandas: https://pandas.pydata.org/
- sklearn: https://scikit-learn.org/
- Etc.

Open the file : data_preprocessing_tools.ipynb

Importing the datasets

- pandas library
- read_csv function - read in csv file.
- iloc function : dataframe.iloc[start_row:end_row,start_col:end_col]
- Do not input row or column number means choose all. Eg. dataset.iloc[:,1:2]

Missing Data

- Showing ‘nan’ when print
- Need to be taken care of before proceed
- Replacing the missing datas with good values.
- sklearn.impute.SimpleImputer
- imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
- strategy : mean, median, most_frequent, constant

Encoding Categorical Data

- Machine learning algorithms requires input and output variables to be numbers.
- Most popular techniques are Ordinal Encoding and One-Hot Encoding.
- Ordinal Encoding - each unique category value is assigned an integer value.
- One-Hot Encoding - binary variable
- LabelEncoder - value between 0 and n_classes-1. This should be used with target values, i.e. y.

Splitting the dataset into the Training set and Test set

- Split the dataset into two subsets.
  - Train Dataset: Used to fit the machine learning model.
  - Test Dataset: Used to evaluate the fit machine learning model.
- When
  - Sufficient large datasets
  - Computational efficiency.
  - Alternative for small datasets : the k-fold cross-validation procedure
- No optimal split percentage.
  - Common splits are 80:20, 2:1, 1:1.

Splitting the dataset into the Training set and Test set

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 1)

- Test_size : [0,1]
- Random_state : 0 - 42
- Shuffle : true/false, default is false.

Feature Scaling

- Feature Scaling is a technique to standardize the independent features present in the data in a fixed range
- Example 3,000 meters vs than 5 km
- Two normal technique
  - Min-Max Normalization : x_new = (xi - min)/(max-min) : distribute value to [0,1]
  - Standardization : x_new = (xi-mean)/stdv : distribute with 0 mean and variance = 1
- Usually do this after Train_test split
- StandardScaler()

Machine Learning Methods

Regression

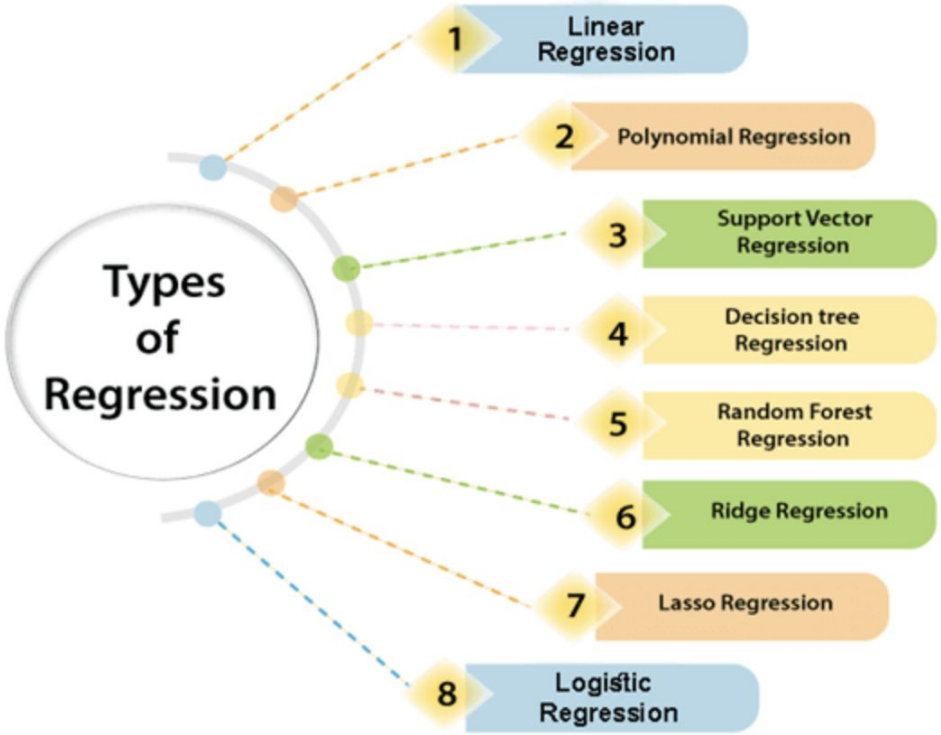
- A statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables.

Want to know the prediction about the sales when the advertisement is $200.

Advertisement	Sales
$90	$1000
$120	$1300
$150	$1800
$100	$1200
$130	$1380
$200	??

Adapted from https://www.javatpoint.com/

Types of Regression

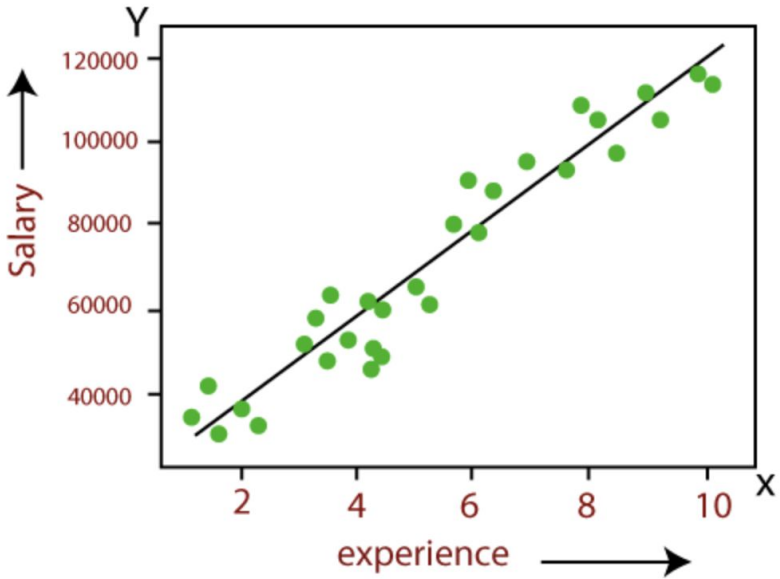


Linear Regression

- simple linear regression : one variable
- multiple linear regression : multiple variables

y= ax+b

Where y = dependent variables (target variables),
x= Independent variables (predictor variables),
a and b are the linear coefficients



Simple Linear Regression

- Open the file simple_linear_regression.ipynb in Codelabs
- Import the libraries
- Import the dataset
- Splitting the data between training set and test set
- Training the model
- Predicting the test result
- Visualize the results

Evaluate Regression Models

- Mean Absolute Error (MAE)
  - from sklearn.metrics import mean_absolute_error

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|e_t|$$

- Mean Squared Error (MSE)
  - from sklearn.metrics import mean_squared_error

$$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n}e_t^2$$

Evaluate Regression Models

R^2 (R squared)

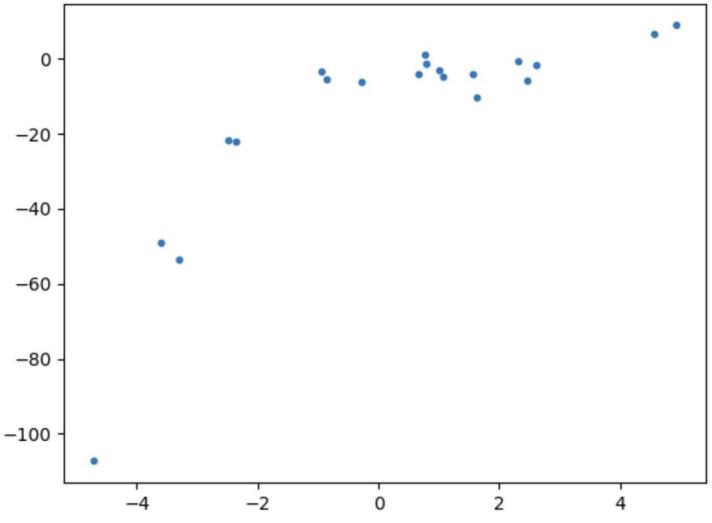
$$R^2\left(y_{true}, y_{pred}\right) = 1 - \frac{\sum\left(y_{true} - y_{pred}\right)^2}{\sum\left(y_{true} - \bar{y}\right)^2}$$

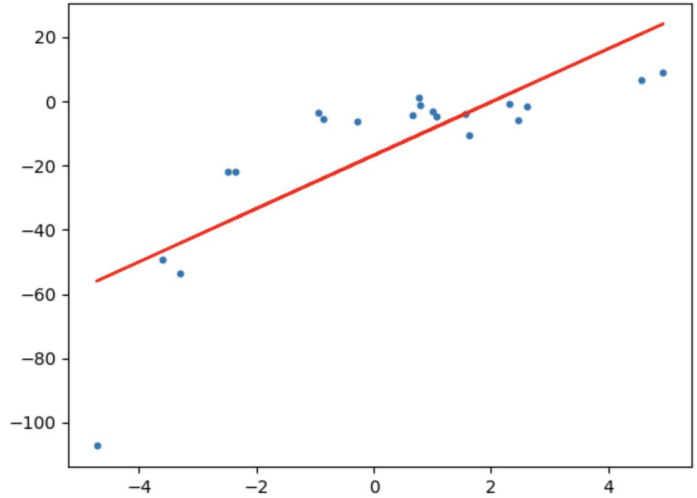
Multiple Linear Regression

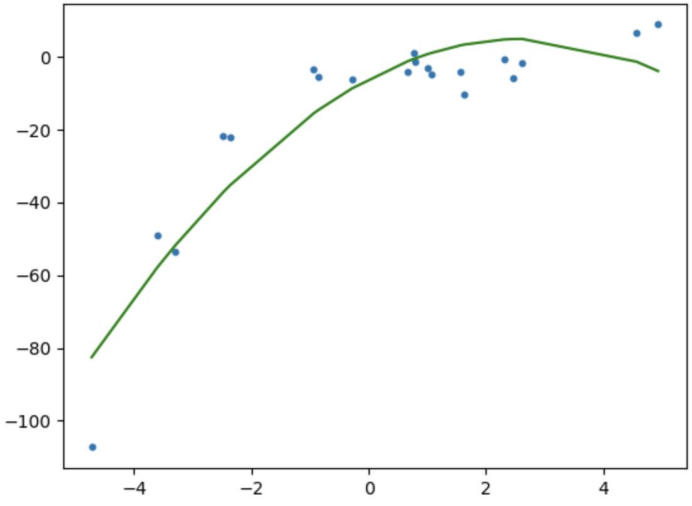
- $y= a_1x_1+ a_2x_2 + a_3x_3 + \ldots + a_nx_n + b$
- where a_1, a_2, a_3, … a_n and b are linear coefficients and x_1,x_2,...x_n are independent variables.
- File : multiple_linear_regression.ipynb
- Dataset : 50_Startups.csv

Polynomial Regression

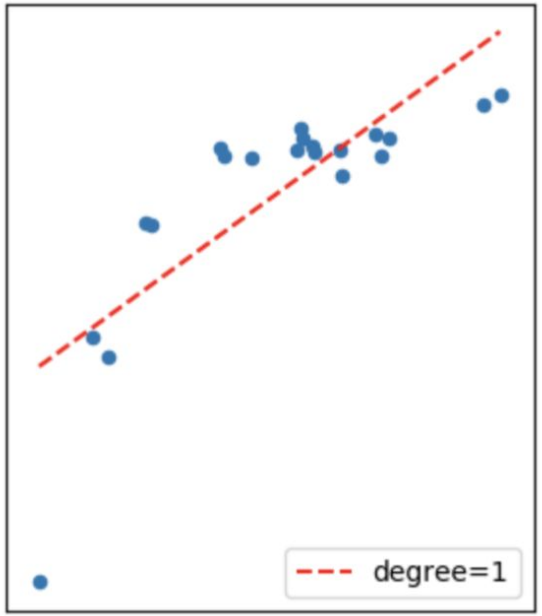
- Generate a curve that best captures the data



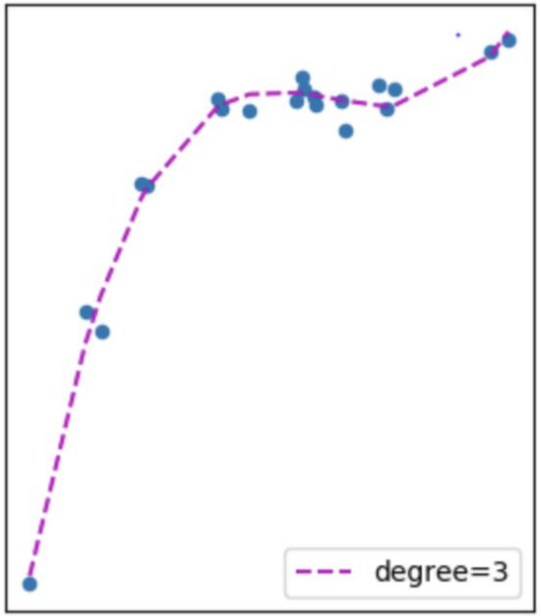




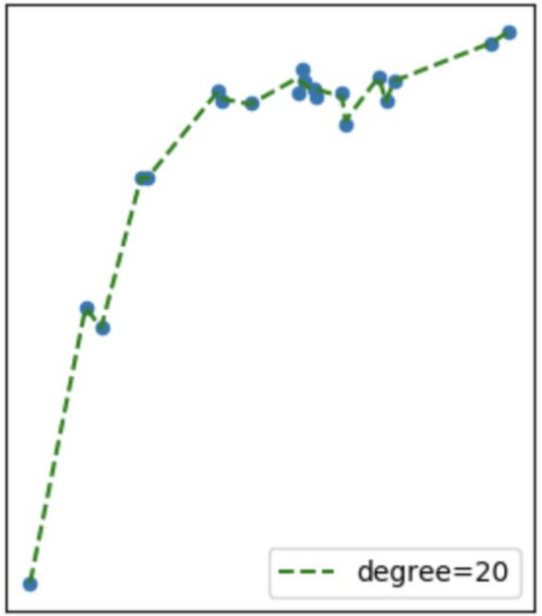
Polynomial Regression



Underfit
High Bias
Low Variance



Correct Fit
Low Bias
Low Variance



Overfit
Low Bias
High Variance

Other regressions

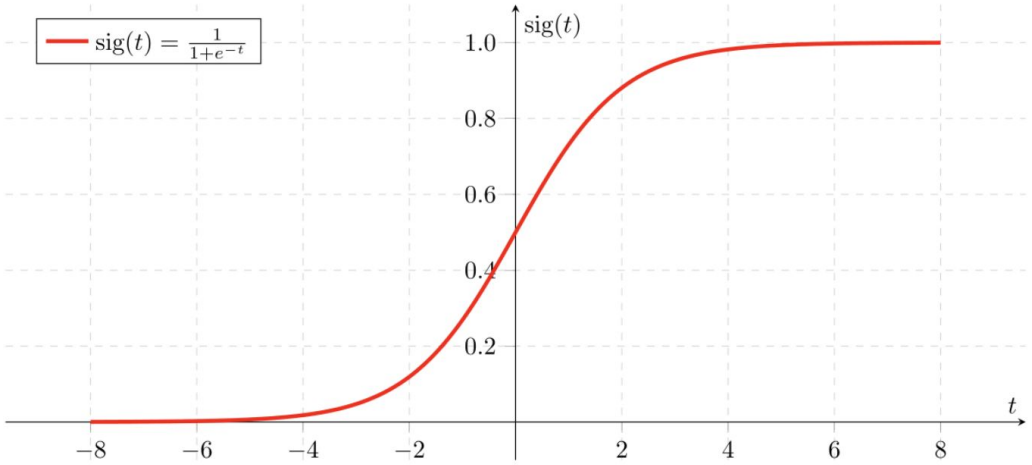
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- Etc.

Classification

- Classification is the process of predicting the class of given data points.
- Predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).
- Classification Methods
  - Logistic regression
  - SVM
  - Naive Bayes
  - Etc.

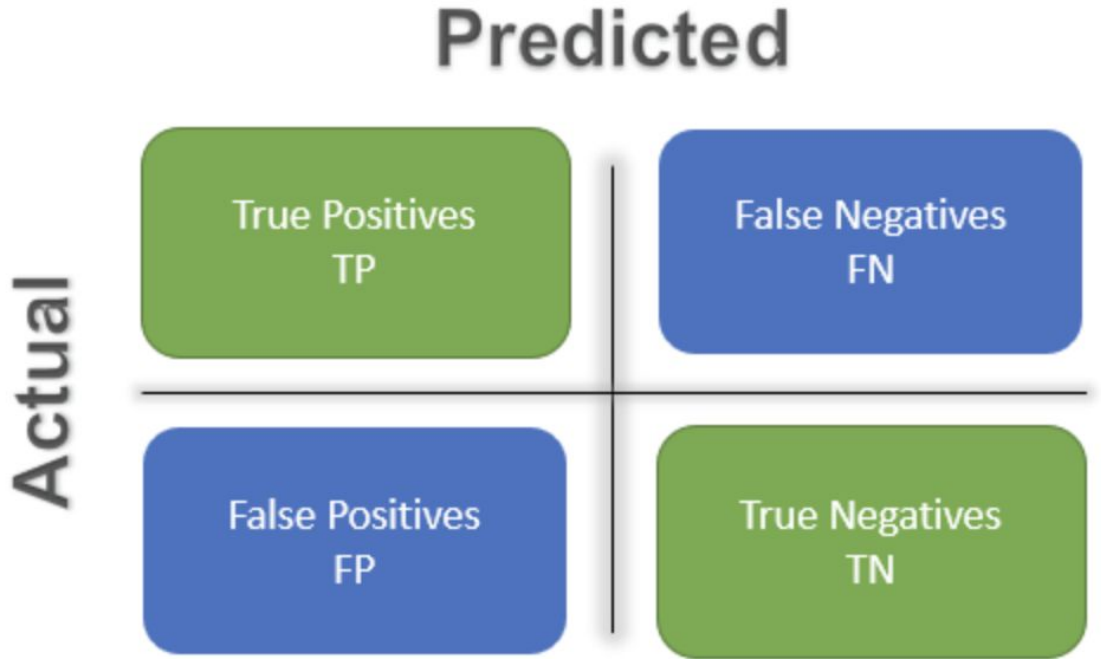
Logistic Regression

- Logistic Regression is used when the dependent variable(target) is categorical. For example to predict whether an email is spam (1) or (0), or whether the tumor is malignant (1) or not (0)



Evaluate Classification

Confusion matrix



Evaluate Classification

Accuracy

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Precision : Percentage of positive instances out of the total predicted positive instances

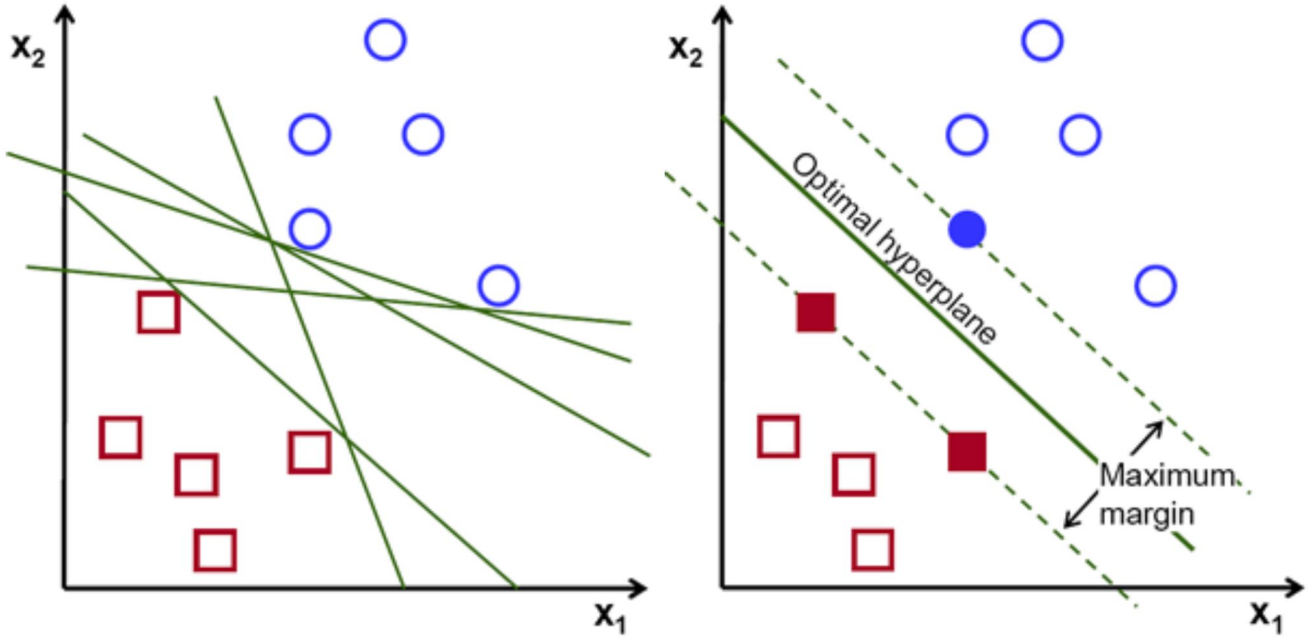
$$\frac{TP}{TP + FP}$$

Recall : Percentage of positive instances out of the total actual positive instances.

$$\frac{TP}{TP + FN}$$

Support Vector Machine

The objective is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.



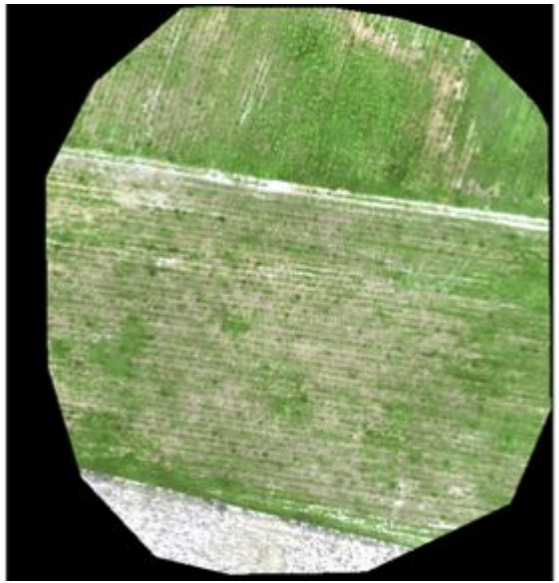
Naive Bayes

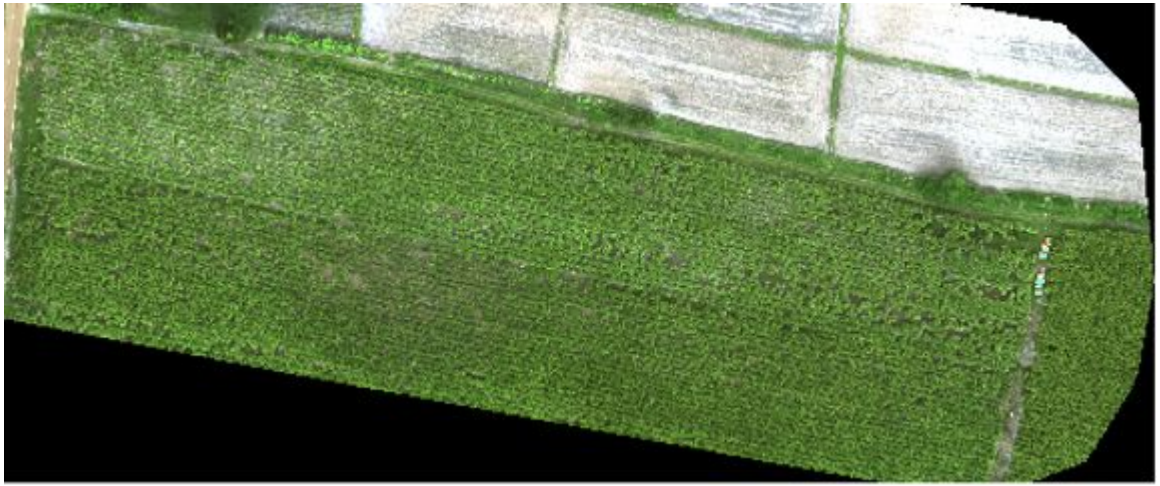
Bayes Theorem

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Assignment

Nutrient Prediction using images from drone





Nutrient Prediction using images from drone

- Image Preprocessing
- Extracting color map
- Nutrient measurement at the experimental sites

Input : R G B

Output : Potassium level (deficiency/optimal/over-optimal)

Data File : K.csv

What to submit

- Submit your code
- Submit the results
  - Confusion matrix
  - Accuracy
  - Precision
  - Recall

- Google Form : https://forms.gle/Ld895X2d3eCKWewY9

Conclusion

- Python
- Google Codelabs
- Data Preparation
- Build Machine Learning Models using sklearn
- Regression
- Classification
- Evaluation of the models