Laboratory Session 4

Logistic Regression-SVM

1. Machine Learning

- Machine Learning is a concept which allows the machine to learn from examples and experience, and that too without being explicitly programmed. So instead of you writing the code, what you do is you feed data to the generic algorithm, and the algorithm/ machine builds the logic based on the given data.

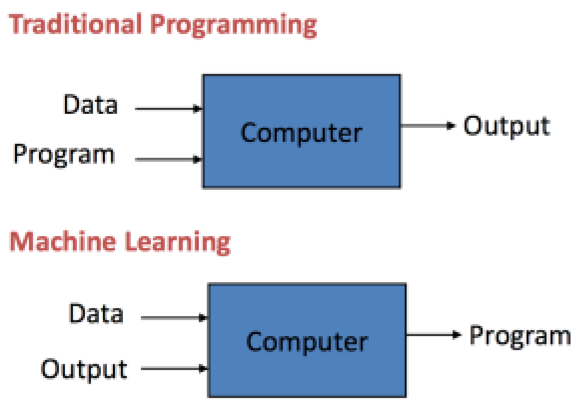

Figure 1 Machine Learning Model

How does Machine Learning Work?

- Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it **makes a prediction on the basis of the model.**
- The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set.

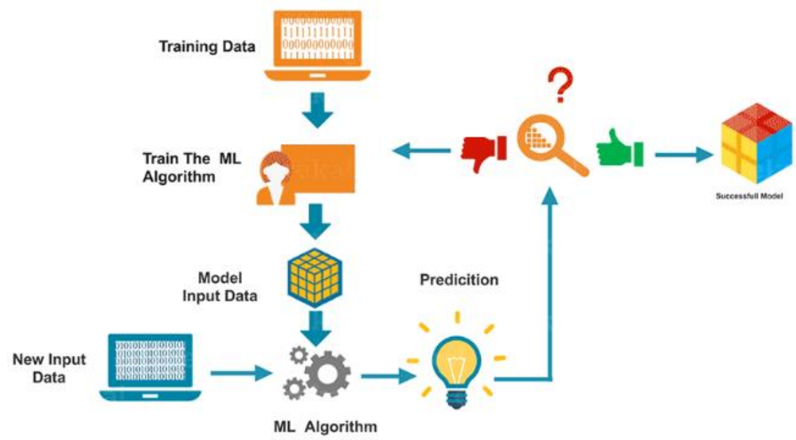

Figure 2 How does ML work?

Types of Machine Learning

- What is Supervised Learning?
    - Supervised Learning is the one, where you can consider the learning is guided by a teacher. We have a dataset which acts as a teacher and its role is to train the model or the machine. Once the model gets trained it can start making a prediction or decision when new data is given to it.
- What is Unsupervised Learning?
    - The model learns through observation and finds structures in the data. Once the model is given a dataset, it automatically **finds patterns and relationships in the dataset by creating clusters in it.**
- What is Reinforcement Learning?
    - It is the ability of an agent to interact with the environment and find out what is the best outcome. It follows the concept of hit and trial method. The agent is rewarded or penalized with a point for a correct or a wrong answer, and on the basis of the positive reward points gained the model trains itself. And again once trained it gets ready to predict the new data presented to it.

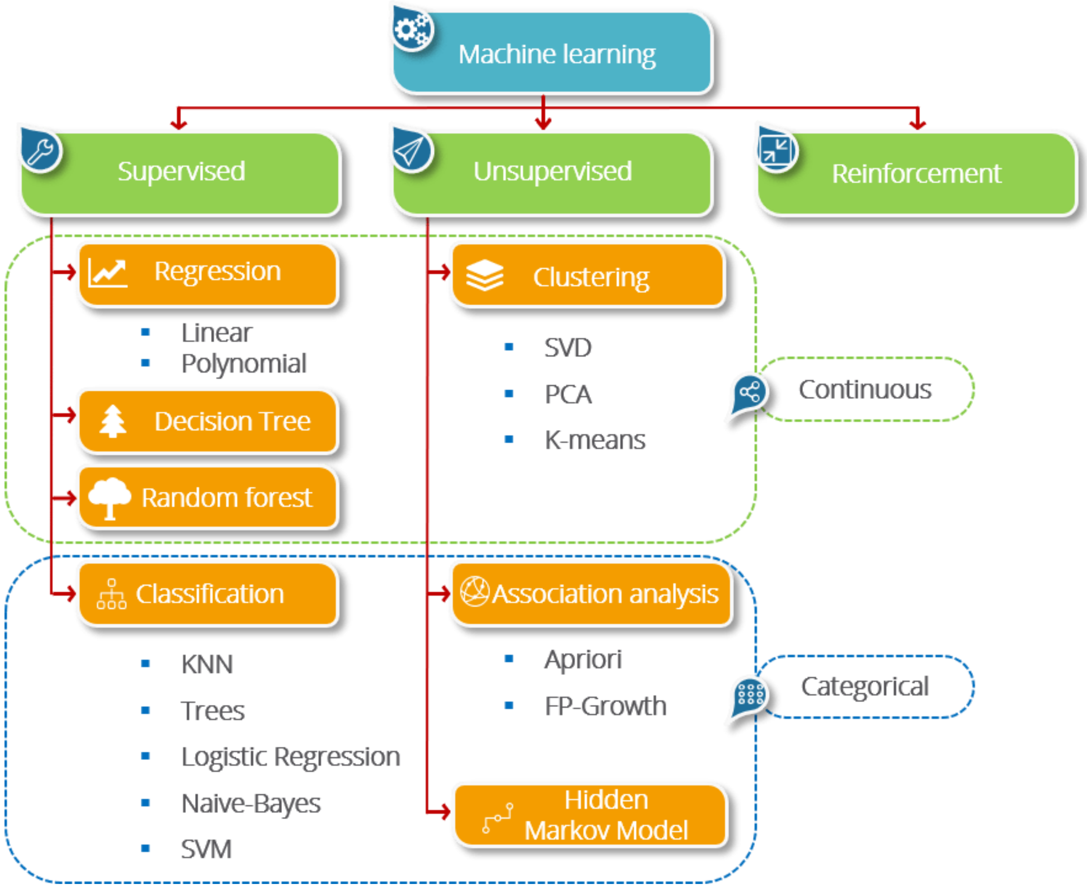

Figure 3 Types of ML

2. Linear/Logistic Regression

Tutorial links: 1. **Logistic Regression:** [https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html] **2. Step by step** [https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8]

- Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.
- **Linear Regression: (Prediction)**
    - Businesses use linear regression to predict such things as future sales, stock prices, currency exchange rates, and productivity gains resulting from a training program.

- For example, imagine that your company wants to understand how past advertising expenditures have related to sales in order to make future decisions about advertising. The dependent variable in this instance is sales and the independent variable is advertising expenditures.
- The typical procedure for finding the line of best fit is called the **least-squares method**. The least-squares method is based upon the principle that the sum of the squared errors should be made as small as possible so the regression line has the least error.
- You **may recall the equation of a straight line from your review of the Linear Functions topic in the Algebra section of this course : f(x) = mx + b**
  - Variables, constants, and coefficients are represented in the equation of a line as
  - x represents the independent variable
  - f(x) represents the dependent variable
  - the constant b denotes the y-intercept—this will be the value of the dependent variable if the independent variable is equal to zero
  - the coefficient m describes the movement in the dependent variable as a result of a given movement in the independent variable.

- **Logistic Regression (Classification) :**
  - Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, **it can be used for cancer detection problems. It computes the probability of an event occurrence.** It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.
  - **Linear Regression Equation**:

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots\ldots + \beta nXn$$

Where, y is dependent variable and x1, x2 ... and Xn are explanatory variables.

- Sigmoid Function:

$$p = 1/1 + e^{-y}$$

- Apply Sigmoid function on linear regression:

$$p = 1/1 + e^{-(\beta 0 + \beta 1 X1 + \beta 2 X2 \ldots\ldots \beta n Xn)}$$

- Linear regression gives you a **continuous output,** but logistic regression provides a **constant output.** An example of the continuous output is house price and stock price. Example's of the discrete output is predicting whether a patient has cancer or not, predicting whether the customer will churn. **Linear regression is estimated using Ordinary Least Squares (OLS) while logistic regression is estimated using Maximum Likelihood Estimation (MLE) approach.**

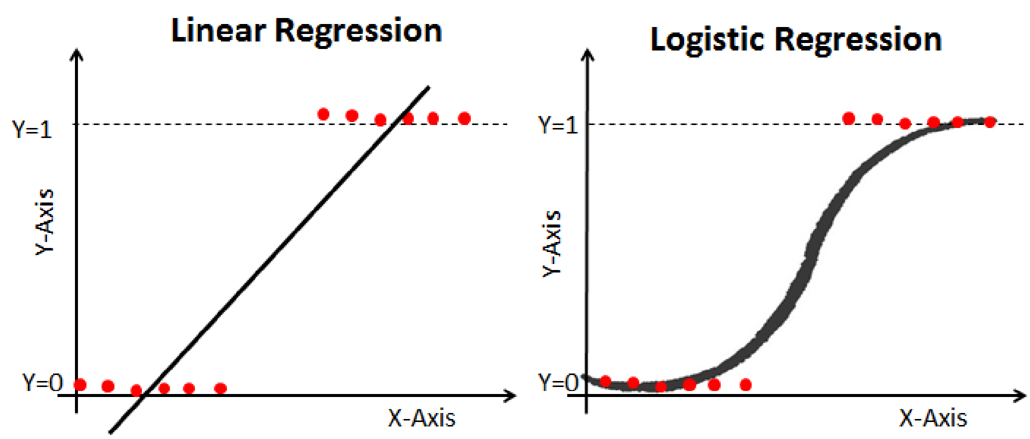

Figure 4 Linear Regression and Logistic Regression

- Maximum Likelihood Estimation(MLE) Vs. Least Square Method:
    - The MLE is a "likelihood" maximization method, while OLS is **a distance-minimizing approximation method.** Maximizing the likelihood function determines the parameters that are most likely to produce the observed data. From a statistical point of view, MLE sets the mean and variance as parameters in determining the specific parametric values for a given model. This set of parameters can be used for predicting the data needed in a normal distribution.

- Ordinary Least squares (OLS) estimates are computed by fitting a regression line on given data points that has the minimum sum of the squared deviations (least square error). Both are used to estimate the parameters of a **linear regression model.** MLE assumes a **joint probability mass function**, while OLS doesn't require any stochastic assumptions for minimizing distance.

3. Support Vector Machine -SVM

Tutorial links: **1.SVM:** [https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/] **2.Theory of SVM** [https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html]

3.SVM+Learn Open CV [https://www.learnopencv.com/svm-using-scikit-learn-in-python/]

4.Python code to demonstrate the application of zip () [https://www.geeksforgeeks.org/zip-in-python/]

- SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. **It is known for its kernel trick to handle nonlinear input spaces**. It is used in a variety of applications such as **face detection, intrusion detection**, classification of emails, news **articles and web pages, classification of genes, and handwriting recognition.**
- The SVM classifier separates data points using a **hyperplane with the largest amount of margin**. That's why an SVM classifier is also known as **a discriminative classifier**. SVM finds an optimal hyperplane which helps in classifying new data points.
    - Support vectors are the data points, **which are closest to the hyperplane**. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.
    - A hyperplane is a **decision plane** which separates between a set of objects having different class memberships.
    - A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance **from the line to support vectors or closest points**. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

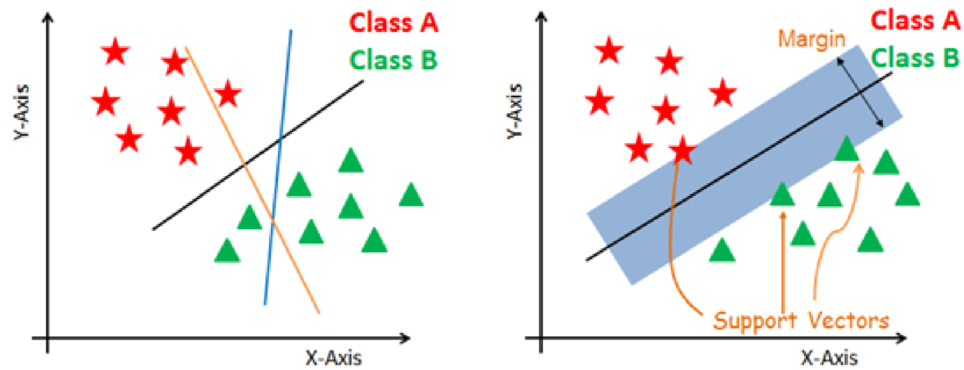

Figure 5 Bound Margin of SVM

- Dealing with non-linear and inseparable planes:
  - In such situation, SVM uses a kernel trick to transform the input space to a higher dimensional space as shown on the right. The data points are plotted on the x-axis and z-axis (Z is the squared sum of both x and y: z=x^2=y^2). Now you can easily segregate these points using linear separation.
  - **Linear Kernel** A linear kernel can be used as normal dot product any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values.
  - **Polynomial Kernel** A polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel can distinguish curved or nonlinear input space
  - **Radial Basis Function Kernel**
    - The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification. RBF can map an input space in infinite dimensional space.
    - Here gamma is a parameter, which ranges from 0 to 1. A higher value of gamma will perfectly fit the training dataset, which causes over-fitting. Gamma=0.1 is considered to be a good default value. The value of gamma needs to be manually specified in the learning algorithm.

Kernel name	Kernel function
Linear kernel	$K(x, y) = x \times y$
Polynomial kernel	$K(x, y) = (x \times y + 1)^d$
RBF kernel	$K(x, y) = e^{-\gamma \lVert x-y \rVert^2}$

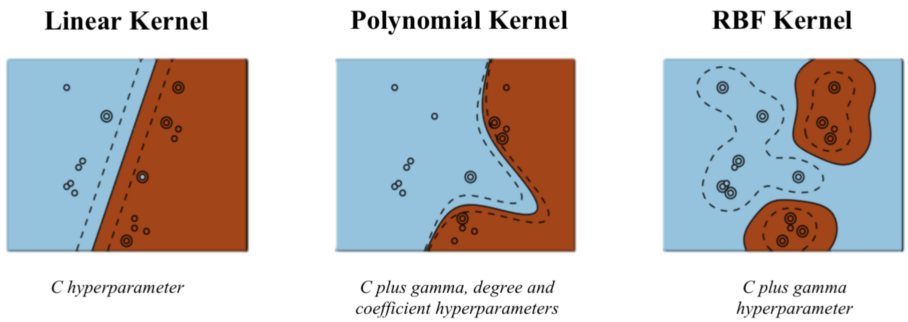

Figure 6 Linear, Polynomial, RBF Kernel bound decision

4. Practice

-read data 'CustomerChurn.csv'

-Import machine learning libraries

```
# import ml libraries
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn import linear_model, datasets
from sklearn.metrics import accuracy_score
from sklearn.svm import LinearSVC, SVC
```

-Split data and train data with encoder set similar to Lab 3

-define the function **def LogisticRegressionLearning(DataTrain, TargetTrain):**

#type your code

return logreg # logreg = *LogisticRegression()*

- define the function **def LogisticRegressionTesting(LRModel, DataTest, TargetTest):**

#type your code

return AccuracyLR, PredictTestLR

-Learn LR methods and print out the accuracy and running time.

-define the function **def SVMLearning(DataTrain, TargetTrain, ClassifierType= " "):**

#type your code

#using if elif else for selected classifierType 'Linear', 'RBF','Polynominal Kernel'…

return SVC #

- define the function **def SVMTesting(SVMModel, DataTest, TargetTest):**

#type your code

return AccuracySVM, PredictTestSVM

-Learn all SVM models and print out all the accuracy and running time values.

-Evaluation algorithm:

1. **Linear:** evaluate the algorithm such as **confusion matrix**, **precision**, **recall**, and **F1** measures are the most commonly used metrics for classification tasks. Hint: Scikit-Learn's metrics library contains the ***classification_report*** and ***confusion_matrix*** methods, which can be readily used to find out the values for these important metrics.

2. ***Polynomial Kernel:*** In the case of polynomial kernel, you also have to **pass a value for the degree parameter** of the SVCclass. This basically is the degree of the polynomial. Take a look at how we can use a polynomial kernel to implement kernel SVM. As usual, the final step of any machine learning algorithm is to make **evaluations for polynomial kernel**.

3. *Gaussian Kernel: u*sing **GridSearchCV** for searching the best parameter for our classifier.

Next, we **plot the decision boundary and support vectors. The decision boundary** is estimated based on only the training data. Given a new data point (say from the test set), we simply need to check which side of the line the point lies to classify it as 0 (red) or 1 (blue). The module **sklearn.model_selection** allows us to do a grid search over parameters using **GridSearchCV**. All we need to do is specify which parameters we want to vary and by what value. In the following example, the parameters **C and Gamma are varied**. Every combination of C and gamma is tried and the best one is **chosen based**. The best estimator can be accessed using **clf.best_estimator_**.

5. Please make conclusions about Accuracy, Running Time, Figures of bound margin/decision of all methods.