19IT601 - DATA SCIENCE & ANALYTICS

UNIT III

Linear Regression – Polynomial Regression – Multivariate Regression – Multi Level Models – Data Warehousing Overview – Bias/Variance Trade Off – K Fold Cross Validation – Data Cleaning and Normalization – Cleaning Web Log Data – Normalizing Numerical Data – Detecting Outliers – Introduction to Supervised And Unsupervised Learning Reinforcement Learning – Dealing with Real World Data – Machine Learning Algorithms – Clustering – Python Based Application.

Regression:

- Regression is a statistical technique to determine the linear relationship between two or more variables. Regression is primarily used for prediction.

- In its simplest (bivariate) form, regression shows the relationship between one independent variable (X) and a dependent variable (Y).

- Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables.

- It is mainly used for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

- Regression shows a line or curve that passes through all the data points on target-predictor graph in such a way that the vertical distance between the data points and the regression line is minimum.

Types of Regression

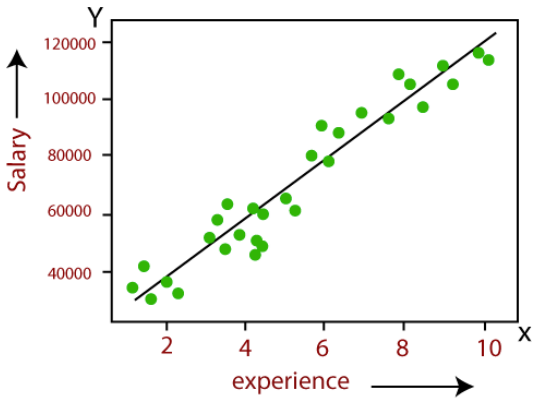
- Linear Regression
- Polynomial Regression
- Multivariate Regression

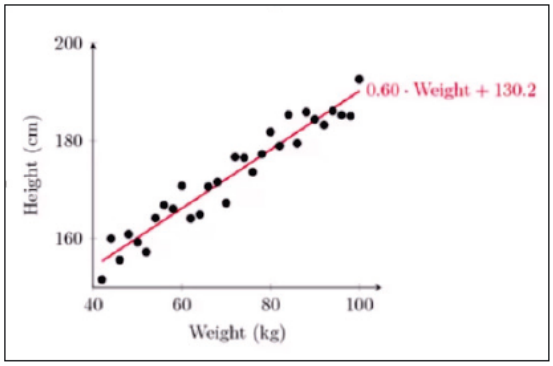
Some examples of regression can be as:

- Prediction of rain using temperature and other factors
- Determining Market trends
- Prediction of road accidents due to rash driving.

Linear Regression

- Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x).
- So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.
- It can be described as Y=f(x). It is fitting a straight line to a set of data points.
- Generally, a linear model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the bias term (also called the intercept term).
- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.
- In this technique, the dependent variable is continuous, the independent variable(s) can be continuous or discrete, and the nature of the regression line is linear.





The equation of a straight line is y=mx+b,
where

- Y is dependent variable (Target variable)
- m is the slope (linear coefficients)
- b is the y-intercept (Bias)

Calculating linear regression
The ordinary least squares technique - it tries to minimize the **squared error** between each point and the line, where the error is just the distance between each point and the line that you have. Then we sum up all the squares of those errors.

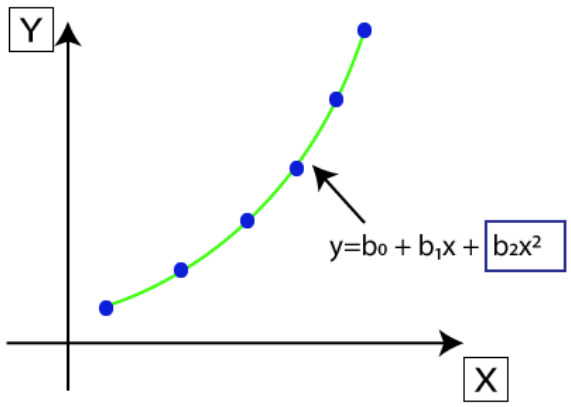
The gradient descent technique –
Gradient descent is an optimization technique used to tune the coefficient and bias of a linear equation. Using the gradient descent technique can make sense when dealing with 3D data.

The co-efficient of determination or r-squared
It is the fraction of the total variation in Y that is captured by your models.

$$1.0 - \frac{sum\ of\ squared\ errors}{sum\ of\ squared\ variation\ from\ mean}$$

Polynomial Regression

- Polynomial Regression is a type of regression which models the non-linear dataset using a linear model
- It is used when data doesn't actually have a linear relationship, or maybe there's some sort of a curve to it.
- Suppose there is a dataset which consists of datapoints which are present in a non-linear fashion, so for such case, linear regression will not best fit to those datapoints.
- To cover such datapoints, we need Polynomial regression. In Polynomial regression, the original features are transformed into polynomial features of given degree and then modeled using a linear model.
- Which means the datapoints are best fitted using a polynomial line.
- The equation for polynomial regression Y= b0+b1x+ b2x2+ b3x3+.....+ bnxn.
- Here Y is the predicted/target output, b0, b1,... bn are the regression coefficients. x is our independent/input variable. The model is still linear as the coefficients are still linear with quadratic.



Multivariate Regression

- Multivariate Regression is a supervised machine-learning algorithm involving multiple data variables for analysis.

- Multivariate regression is an extension of multiple regression with one dependent variable and multiple independent variables.

- Based on the number of independent variables, we try to predict the output.

Here, the plane is the function that expresses y as a function of x and z. The linear regression equation can now be expressed as: **y = m1.x + m2.z+ c**

- y is the dependent variable, that is, the variable that needs to be predicted.
- x is the first independent variable. It is the first input.
- m1 is the slope of x1. It lets us know the angle of the line (x).
- z is the second independent variable. It is the second input.
- m2 is the slope of z. It helps us to know the angle of the line (z).
- c is the intercept.

Below is the generalized equation for the multivariate regression model- $y = \beta0 + \beta1.x1 + \beta2.x2 + \ldots.. + \beta n.xn$

Where n represents the number of independent variables, $\beta0 \sim \beta n$ represents the coefficients, and x1~xn is the independent variable.

Multi Level Models

- Multilevel modelling is a statistical model that is used to model the relationship between dependent data and independent data when there is a correlation between observations.

- These models are also known as hierarchical models, mixed effect models, nested data models or random coefficient models.

- Here, the individual observations are nested inside different groups. The observations within each group are correlated

- The concept of multi-level models is that some effects happen at various levels in the hierarchy.

- In Multi-level models there is a hierarchy of effects that influence each other at larger and larger scales.

Types

- Random Intercept Model
- Random Coefficient Model or Random Slopes and Intercepts Model

Data Warehousing Overview

- It's basically a giant database that contains information from many different sources and ties them together.

- A Data Warehouse (DW) is a relational database that is designed for query and analysis rather than transaction processing. It includes historical data derived from transaction data from single and multiple sources.

- A Data Warehouse provides integrated, enterprise-wide, historical data and focuses on providing support for decision-makers for data modeling and analysis.

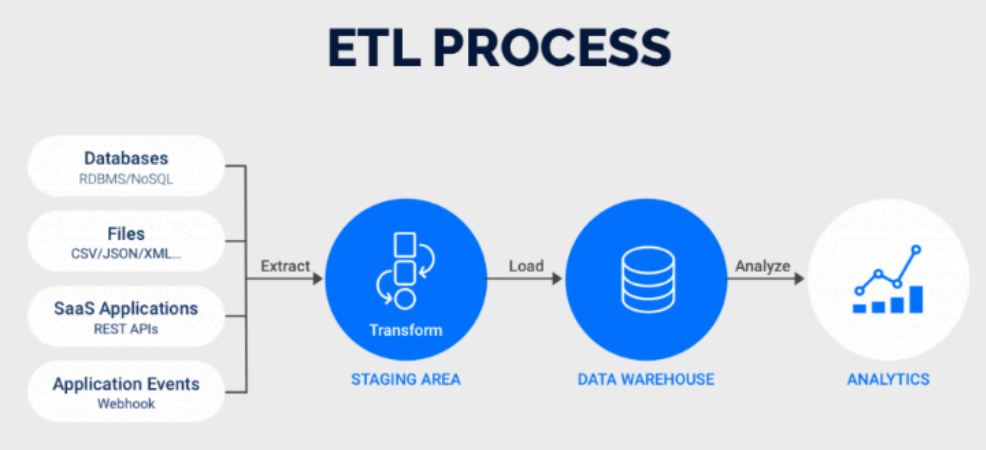
- A data warehouse has the challenge of taking data from many different sources, transforming them into some sort of schema that allows us to query these different data sources simultaneously, and it lets us make insights, through data analysis.

- Data Warehouse environment contains an extraction, transportation, and loading (ETL) solution, an online analytical processing (OLAP) engine, customer analysis tools, and other applications that handle the process of gathering information and delivering it to business users.

ETL - Extract, Transform, and Load

Extract, transform, and load (ETL) is a data integration methodology that extracts raw data from sources, transforms the data on a secondary processing server, and then loads the data into a target database.

The method emerged in the 1970s, and remains prevalent amongst on-premise databases that possess finite memory and processing power.



Extract:

- Extraction refers to pulling the source data from the original database or data source. With ETL, the data goes into a temporary staging area.

- In a traditional ETL scenario, the source data is extracted to a staging area and moved into the target system

Transform:

- Transformation refers to the process of changing the structure of the information, so it integrates with the target data system and the rest of the data in that system.

- In staging area, Transformation can involve converting all data types to the same format, cleansing by removing inconsistent or inaccurate data, combining data elements from multiple data models, pulling in data from other sources, and other processes.

Load:

- Loading refers to the process of depositing the information into a target data storage system.

Example – OLAP Data warehouse

ELT – Extract, Load, Transform

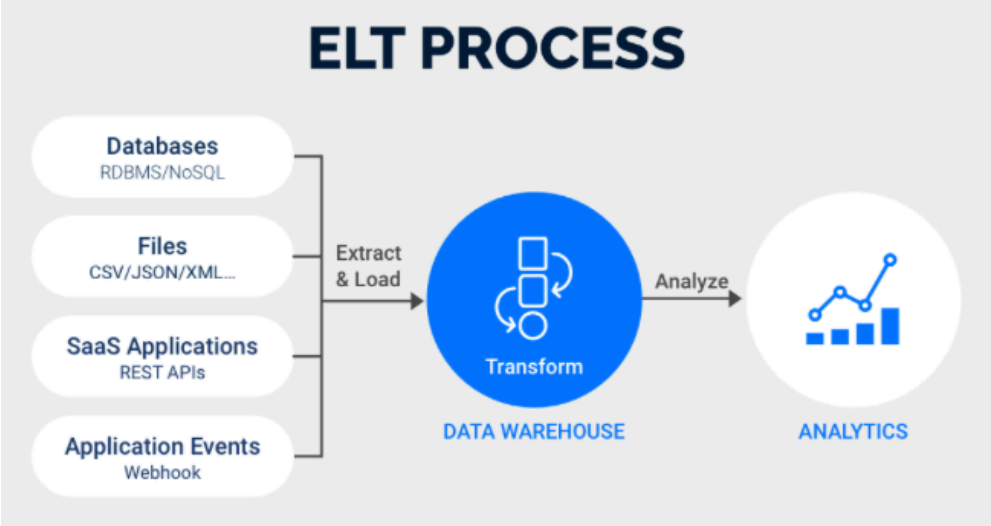
- Unlike ETL, extract, load, and transform (ELT) does not require data transformations to take place before the loading process.

- ELT loads raw data directly into a target data warehouse, instead of moving it to a processing server for transformation.

- With ELT, data cleansing, enrichment, and transformation all occur inside the data warehouse itself. Raw data is stored indefinitely in the data warehouse, allowing for multiple transformations.

- ELT is a relatively new development, made possible by the invention of scalable cloud-based data warehouses.

- Example – Hadoop, Data Lakes are special kinds of data stores that—unlike OLAP data warehouses—accept any kind of structured or unstructured data.



Advantages of ELT

- ELT lets the data destination do the transformation, eliminating the need for data staging.
- Flexibility,
- Faster loading time,
- Scalability.

Advantages of ETL

ETL can help with data privacy and compliance, cleansing sensitive data before loading into the data destination, while ELT is simpler and for companies with minor data needs.

Overfitting and Underfitting

Overfitting

- A statistical model is said to be overfitted when we train it with a lot of data.
- When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set.
- Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.
- This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model.
- The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.
- Then the model does not categorize the data correctly, because of too many details and noise.
- Overfitting – High variance and low bias.

Underfitting

- A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data.
- Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough.
- It usually happens when we have fewer data to build an accurate model and also when we try to build a linear model with fewer non-linear data.
- Underfitting – High bias and low variance

Bias vs Variance

Bias: It measures the difference between the model's prediction and the target value. If the model is oversimplified, the predicted value would be far from the ground truth resulting in more bias.

Variance: Variance measures the inconsistency of different predictions over a varied dataset. Suppose the model's performance is tested on different datasets—the closer the prediction, the lesser the variance. Higher variance indicates overfitting, in which the model loses the ability to generalize.

Variance is just a measure of how spread out, how scattered your predictions are.

Bias-Variance Trade-Off

- The goal of any supervised machine learning algorithm is to achieve low bias and low variance. In turn the algorithm should achieve good prediction performance.
- There is no escaping the relationship between bias and variance in machine learning.
- Increasing the bias will decrease the variance.
- Increasing the variance will decrease the bias.
- At the end you're not out to just reduce bias or just reduce variance, you want to reduce error.
- A good practice is to check the training error and test error
- That's what really matters, and it turns out you can express error as a function of bias and variance:

$$Error = Bias^2 + Variance$$

K-fold cross-validation

- We split all of our data that we're building a machine learning model based off of into two segments
- A training dataset, and a test dataset.
- The idea is that we train our model only using the data in our training dataset, and then we evaluate its performance using the data that we reserved for our test dataset.
- K-fold cross-validation is one of the most common techniques used to detect overfitting.
- Here, we split the data points into k equally sized subsets in K-folds cross-validation, called "folds." One split subset acts as the testing set while the remaining groups are used to train the model.

- k-fold cross-validation splits the dataset into ‘k’ number of folds, then uses one of the ‘k’ folds as a validation set, and the other k-1 folds as a training set. This process is repeated k times, such that each of the k folds is used once as the test set. The scores obtained from this k times training and testing are then averaged to obtain the final score.

The idea, although it sounds complicated, is fairly simple:
1. Instead of dividing our data into two buckets, one for training and one for testing, we divide it into K buckets.

2. We reserve one of those buckets for testing purposes, for evaluating the results of our model.

3. We train our model against the remaining buckets that we have, K-1, and then we take our test dataset and use that to evaluate how well our model did amongst all of those different training datasets.

4. We average those resulting error metrics, that is, those r-squared values, together to get a final error metric from k-fold cross-validation.

Data cleaning

Cleaning raw input data is often the most important, and time consuming, part as a data scientist.

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

Why data cleaning is important?

- When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

- If data cleaning process is not done, then it's going to skew the results, and it will ultimately end up in the wrong decisions.

- If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct.

There are a lot of different kinds of problems and data that you need to watch out for:

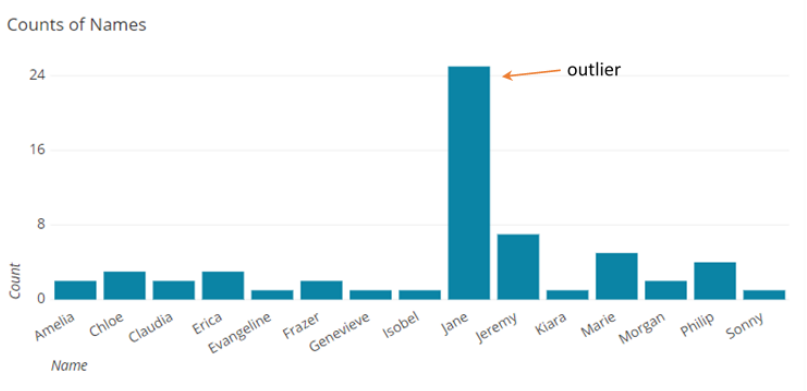
- Outliers,
- Missing Data
- Erroneous data
- Irrelevant data
- Inconsistent data
- Formatting

Outliers

- Outliers are those data points that are really far from the rest of your data points. In other words an outlier is a value or data point that differs substantially from the rest of the data.

- An outlier is an extremely high or extremely low data point relative to the rest of the data points in a dataset.

- They can show up due to errors in data entry or measurement, or just because there's variation in the the population you're looking at and you happened to see one of the more unusual values



Missing Data

A missing value can signify a number of different things. Perhaps the field was not applicable, the event did not happen, or the data was not available.

Erroneous data – Wrong data, invalid data that the program cannot process and should not accept.

Irrelevant data - Irrelevant data are those that are not actually needed, and don't fit under the context of the problem we're trying to solve.

Inconsistent data - Data inconsistency is a situation where there are multiple tables within a database that deal with the same data but may receive it from different inputs.

Formatting - Data can be inconsistently formatted. Take the example of dates: in the US we always do month, day, year (MM/DD/YY), but in other countries they might do day, month, year (DD/MM/YY).

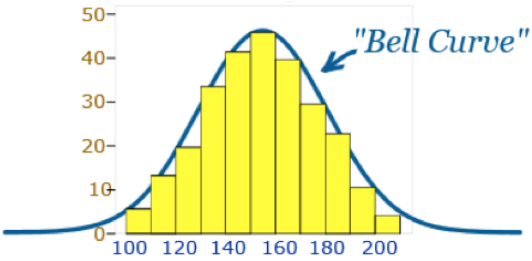
Data Normalization

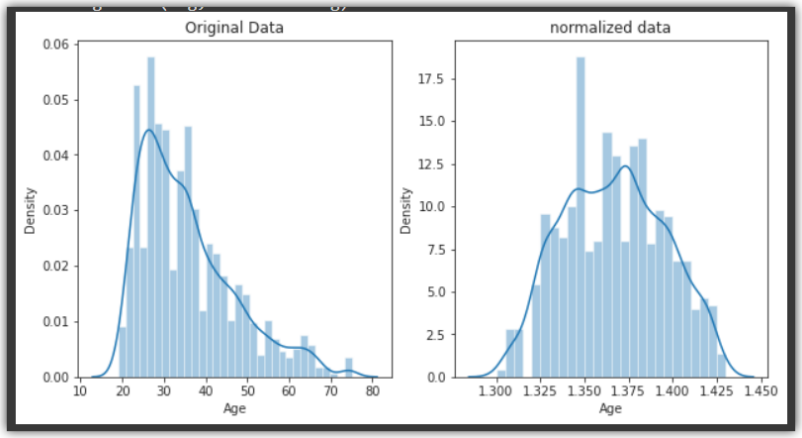
- Normalization is a data preparation technique that is frequently used in machine learning.

- The process of transforming the columns in a dataset to the same scale is referred to as normalization.

- Every dataset does not need to be normalized for machine learning.

- It is only required when the ranges of characteristics are different.
- Normalization is used to scale the data of an attribute so that it falls in a smaller range, such as -1.0 to 1.0 or 0.0 to 1.0.
- Normalization helps to improve the performance as well as the accuracy of your model better.
- It will not affect regression model that much but it needed for linear discriminant analysis (LDA) and Gaussian naive Bayes.
- It is useful when the feature distribution of data does not follow a Gaussian (bell curve) distribution.





Normalization techniques in Machine Learning

Although there are so many feature normalization techniques in Machine Learning, few of them are most frequently used. These are as follows:

Min-Max Scaling:

This technique is also referred to as scaling. The Min-Max scaling method helps the dataset to shift and rescale the values of their attributes, so they end up ranging between 0 and 1.

Standardization scaling:

Standardization scaling is also known as Z-score normalization, in which values are centered around the mean with a unit standard deviation, which means the attribute becomes zero and the resultant distribution has a unit standard deviation.

This technique is helpful for various machine learning algorithms that use distance measures such as KNN, K-means clustering, and Principal component analysis.

Detecting Outliers

- Outliers are those data points that are really far from the rest of your data points. In other words an outlier is a value or data point that differs substantially from the rest of the data.

Reasons for outliers in data

- Errors during data entry or a faulty measuring device (a faulty sensor may result in extreme readings).
- Natural occurrence

Box plots

- Box plots are a visual method to identify outliers. Box plots is one of the many ways to visualize data distribution.

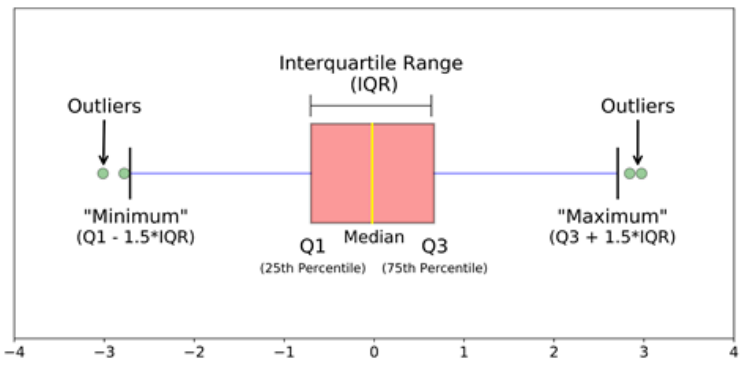
- Box plot plots the q1 (25th percentile), q2 (50th percentile or median) and q3 (75th percentile) of the data along with (q1–1.5*(q3-q1)) and (q3+1.5*(q3-q1)).

- Outliers, if any, are plotted as points above and below the plot.

IQR method

IQR method is used by box plot to highlight outliers. IQR stands for interquartile range, which is the difference between q3 (75th percentile) and q1 (25th percentile).

The IQR method computes lower bound and upper bound to identify outliers.
Lower Bound = q1–1.5*IQR
Upper Bound = q3+1.5*IQR



IQR to detect outliers

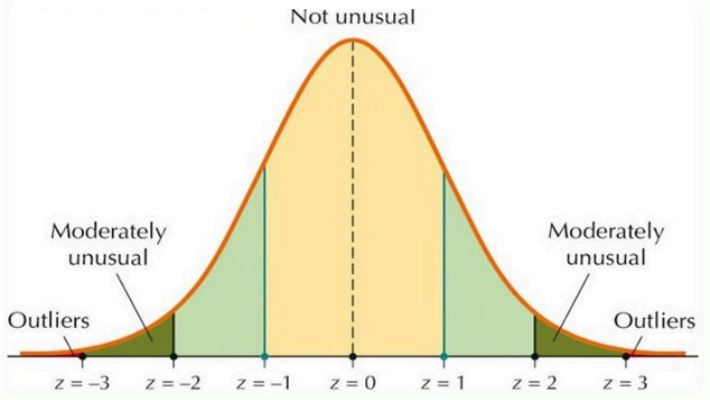
Z-score method

- Z-score method is another method for detecting outliers. This method is generally used when a variable' distribution looks close to Gaussian.

- Z-score is the number of standard deviations a value of a variable is away from the variable' mean. Z-Score = (X-mean) / Standard deviation

- when the values of a variable are converted to Z-scores, then the distribution of the variable is called standard normal distribution with mean=0 and standard deviation=1.

- The Z-score method requires a cut-off specified by the user, to identify outliers. The widely used lower end cut-off is -3 and the upper end cut-off is +3.

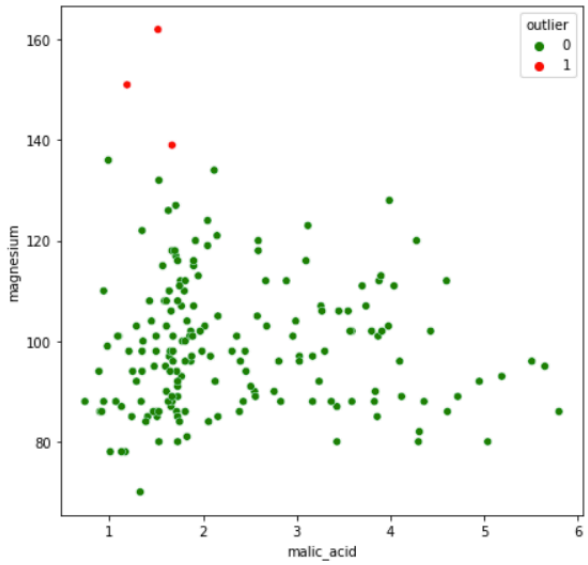


Distance from the mean' method (Multivariate method)

- Unlike the previous methods, this method considers multiple variables in a data set to detect outliers.

- This method calculates the Euclidean distance of the data points from their mean and converts the distances into absolute z-scores.

- Any z-score greater than the pre-specified cut-off is considered to be an outlier.



Introduction to Supervised And Unsupervised Learning – Reinforcement Learning

Machine learning algorithms are classified into 3 types:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

Supervised ML algorithms is a type of ML technique that can be applied according to what was previously learned to get new data using labeled data and to predict future events or labels.

In supervised learning, we give it a set of training data, that the model learns from. It can then infer relationships between the features and the categories that we want, and apply that to unseen new values - and predict information about them.

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output..

Supervised learning can be further divided into two types:

- Classification
- Regression

Classification

Classification is used when the output variable is categorical i.e. with 2 or more classes. For example, yes or no, male or female, true or false, etc.

Regression

Regression is used when the output variable is a real or continuous value. In this case, there is a relationship between two or more variables.

Types of Supervised Learning Algorithms

- Linear Regression
- Logistic Regression
- Naive Bayes Classifiers
- Decision Trees
- Random Forest
- Support Vector Machine

Unsupervised Learning

In Unsupervised Learning, the machine uses unlabeled data and learns on itself without any supervision. The machine tries to find a pattern in the unlabeled data and gives a response.

The goal of unsupervised learning is to find the structure and patterns from the input data. Unsupervised learning does not need any supervision. Instead, it finds patterns from the data by its own.

Unsupervised learning can be further grouped into types:

- Clustering
- Association

Clustering
Clustering is the method of dividing the objects into clusters that are similar between them and are dissimilar to the objects belonging to another cluster. It is grouping of data's based on similarity.

For example, finding out which customers made similar product purchases.

Association
Association is a rule-based machine learning to discover the probability of the co-occurrence of items in a collection. For example, finding out which products were purchased together.

The most commonly used unsupervised learning algorithms are:

- K-means clustering
- Hierarchical clustering
- Apriori algorithm
- Principal Component Analysis

Reinforcement Learning

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.

Types

- Q-learning and
- SARSA (State-Action-Reward-State-Action)

Machine Learning Algorithms

Supervised Learning

Naive Bayes Classifiers

The Bayes theorem is a mathematical formula for calculating conditional probability in probability and statistics. In other words, it's used to figure out how likely an event is based on its proximity to another. Simply put, it is a way of calculating conditional probability.

The probability of A given B is equal to the probability of A times the probability of B given A over the probability of B.

The key insight is that the probability of something that depends on B depends very much on the base probability of B and A.

We can find conditional probability using Bayes' Theorem with the following formula:

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

The components has special names:

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

posterior: $P(A|B)$; likelihood: $P(B|A)$; prior: $P(A)$; evidence: $P(B)$

'A' is the event of interest.

P(A) represents our prior belief: probability of event A occurring.

With new evidence B, the posterior belief or updated probability is represented P(A|B): probability of event A given evidence B has occurred.

P(B | A) is the conditional probability of event B occurring, given that A is true

Example of Spam Filtering

Let's just figure out the probability of an e-mail being spam given that it contains the word "free".

The probability of an email being spam given that you have the word "free" in that e-mail works out to the overall probability of it being a spam message times the probability of containing the word "free" given that it's spam over the probability overall of being free:

$$P(Spam \mid Free) = \frac{P(Spam)P(Free \mid Spam)}{P(Free)}$$

The numerator can just be thought of as the probability of a message being spam and containing the word free.
But that's a little bit different than what we're looking for, because that's the odds out of the complete dataset and not just the odds within things that contain the word free.
The denominator is just the overall probability of containing the word free. Sometimes that won't be immediately accessible to you from the data that you have. If it's not, you can expand that out to the following expression if you need to derive it:

$$P(Free|Spam)P(Spam) + P(Free|Not\ Spam)P(Not\ Spam))$$

This gives you the percentage of e-mails that contain the word "free" that are spam, which would be a useful thing to know when you're trying to figure out if it's spam or not.

Implementing a spam classifier with Naïve Bayes

```
import os
import io
import numpy
from pandas import DataFrame
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB

def readFiles(path):
    for root, dirnames, filenames in os.walk(path):
        for filename in filenames:
            path = os.path.join(root, filename)

            inBody = False
            lines = []
            f = io.open(path, 'r', encoding='latin1')
            for line in f:
                if inBody:
                    lines.append(line)
                elif line == '\n':
                    inBody = True

            f.close()
            message = '\n'.join(lines)
            yield path, message


def dataFrameFromDirectory(path, classification):
    rows = []
    index = []
    for filename, message in readFiles(path):
        rows.append({'message': message, 'class': classification})
        index.append(filename)

    return DataFrame(rows, index=index)

data = DataFrame({'message': [], 'class': []})

data = data.append(dataFrameFromDirectory(
                   'e:/sundog-consult/Udemy/DataScience/emails/spam',
                   'spam'))
data = data.append(dataFrameFromDirectory(
                   'e:/sundog-consult/Udemy/DataScience/emails/ham',
                   'ham'))
```

Decision Trees

A decision tree employs a structure of test points (called nodes) and branches, which represent the decision being made. A node without further branches is called a leaf node. The leaf nodes return class labels.

Decision trees have two varieties: **classification trees and regression trees**.

Classification trees usually apply to output variables that are categorical—often binary—in nature, such as yes or no, purchase or not purchase, and so on.

Regression trees, on the other hand, can apply to output variables that are numeric or continuous, such as the predicted price of a consumer good or the likelihood a subscription will be purchased.

General Algorithm

In general, the objective of a decision tree algorithm is to construct a tree T from a training set S.

- If all the records in S belong to some class C, or if S is sufficiently pure (greater than a preset threshold), then that node is considered a leaf node and assigned the label C.

- In contrast, if not all the records in S belong to class C or if S is not sufficiently pure, the algorithm selects the next most informative attribute A and partitions S according to A‘s values.

- The algorithm constructs subtrees , T1,T2… for the subsets of S recursively until one of the following criteria is met:
    - All the leaf nodes in the tree satisfy the minimum purity threshold.
    - The tree cannot be further split with the preset minimum purity threshold.
    - Any other stopping criterion is satisfied (such as the maximum depth of the tree).

The first step in constructing a decision tree is to choose the most informative attribute.

A common way to identify the most informative attribute is to use entropy-based methods.

The entropy methods select the most informative attribute based on two basic measures:

- Entropy, which measures the impurity of an attribute
- Information gain, which measures the purity of an attribute

Entropy

Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information.

$$H_X = -\sum_{\forall x \in X} P(x) \log_2 P(x)$$

Conditional Entropy

The next step is to identify the conditional entropy for each attribute. Given an attribute X, its value x , its outcome Y, and its value y, conditional entropy is the remaining entropy of given , formally defined as

$$H_{Y|X} = \sum_{x} P(x) H(Y|X = x)$$
$$= -\sum_{\forall x \in X} P(x) \sum_{\forall y \in Y} P(y|x) \log_2 P(y|x)$$

Information Gain

The information gain of an attribute A is defined as the difference between the base entropy and the conditional entropy of the attribute

$$InfoGain_A = H_S - H_{S|A}$$

Where H_S - Base Entropy

$H_{S|A}$ – Conditional Entropy

Information gain compares the degree of purity of the parent node before a split with the degree of purity of the child node after a split.

At each split, an attribute with the greatest information gain is considered the most informative attribute. Information gain indicates the purity of an attribute.

Decision Tree Algorithms

Multiple algorithms exist to implement decision trees, and the methods of tree construction vary with different algorithms.

Some popular algorithms include

- ID3,
- C4.5, and
- CART

ID3

- ID3 (or Iterative Dichotomiser 3) is one of the first decision tree algorithms, and it was developed by John Ross Quinlan.
- Let A be a set of categorical input variables, P be the output variable (or the predicted class), and T be the training set.
- ID3 follows the rule — A branch with an entropy of zero is a leaf node and A branch with entropy more than zero needs further splitting.
- It uses Entropy and Information gain to select most informative attribute.

```
1 ID3 (A, P, T)
2 if T∈ϕ
3  return ϕ
4 if all records in T have the same value for P
5  return a single node with that value
6 if A∈ϕ
7  return a single node with the most frequent value of P in T
8 Compute information gain for each attribute in A relative to T
9 Pick attribute D with the largest gain
10 Let {d1,d2…dm} be the values of attribute D
11 Partition T into {T1,T2…Tm} according to the values of D
12 return a tree with root D and branches labeled d1,d2…dm
   going respectively to trees ID3(A-{D}, P, T1),
   ID3(A-{D}, P, T2), … ID3(A-{D}, P, Tm)
```

C4.5

- The C4.5 algorithm introduces a number of improvements over the original ID3 algorithm.
- The C4.5 algorithm can handle missing data. If the training records contain unknown attribute values, the C4.5 evaluates the gain for an attribute by considering only the records where the attribute is defined.
- Both categorical and continuous attributes are supported by C4.5.
- For the corresponding records of each partition, the gain is calculated, and the partition that maximizes the gain is chosen for the next split
- The ID3 algorithm may construct a deep and complex tree, which would cause overfitting.
- The C4.5 algorithm addresses the overfitting problem in ID3 by using a bottom-up technique called pruning to simplify the tree by removing the least visited nodes and branches.
- It uses Entropy and Information gain to select most informative attribute.

CART

- CART (or Classification And Regression Trees) is often used as a generic acronym for the decision tree, although it is a specific implementation.
- Similar to C4.5, CART can handle continuous attributes.
- Whereas C4.5 uses entropy based criteria to rank tests, CART uses the Gini diversity index defined.
- Whereas C4.5 employs stopping rules, CART constructs a sequence of subtrees, uses cross-validation to estimate the misclassification cost of each subtree, and chooses the one with the lowest cost.

Advantages of Decision Tree

- Decision trees are computationally inexpensive, and it is easy to classify the data.
- Decision trees are able to handle both numerical and categorical attributes and are robust with redundant or correlated variables.
- Decision trees can handle categorical attributes with many distinct values, such as country codes for telephone numbers.
- Decision trees can also handle variables that have a nonlinear effect on the outcome, so they work better than linear models (for example, linear regression and logistic

regression) for highly nonlinear problems.
- Decision trees can also be used to prune redundant variables.

Disadvantages of Decision Tree
- Decision trees are not a good choice if the dataset contains many irrelevant variables.
- This is different from the notion that they are robust with redundant variables and correlated variables.
- Although decision trees are able to handle correlated variables, decision trees are not well suited when most of the variables in the training set are correlated, since overfitting is likely to occur.

Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

In random forest, each decision tree takes a different random sample from our set of training data and constructs a tree from it. Then each resulting tree can vote on the right result.

This technique of randomly resampling our data with the same model is a term called bootstrap aggregating, or bagging.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Random forest uses ensemble technique. Ensemble simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

Types of ensemble learning:
Bootstrap aggregating or bagging: It takes random subsamples of our training data and feed them into different versions of the same model and let them all vote on the final result.

Boosting: Boosting is an alternate model, and the idea here is that start with a model, but each subsequent model boosts the attributes that address the areas that were misclassified by the previous model.

Bucket of models: Another technique, is called a bucket of models, where you might have entirely different models that try to predict something.
For example, using k-means, a decision tree, and regression. We run all three of those models together on a set of training data and let them all vote on the final classification result when trying to predict something.

Stacking: In this we run multiple models on the data, combine the results together somehow. The subtle difference here between bucket of models and stacking, is that, pick the model that wins. So, we run train/test, and find the model that works best for data, and use that model. By contrast, stacking will combine the results of all those models together, to arrive at a final result.

Steps involved in random forest algorithm:
Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems.

In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate.

Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

Support vector machines (SVM), which is a very advanced way of clustering or classifying higher dimensional data. If the dataset has many different features that we trying to predict from, then support vector machines might be a good way of doing that.

Support vector machines finds higher-dimensional support vectors across which to divide the data (these support vectors define hyperplanes). It finds higher dimensional support vectors that define the higher-dimensional planes that split the data into different clusters.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen.

Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes.

Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Unsupervised Learning Algorithms

Clustering

- Clustering techniques are unsupervised in the sense that the data scientist does not determine, in advance, the labels to apply to the clusters.
- The structure of the data describes the objects of interest and determines how best to group the objects. Clustering is a method often used for exploratory analysis of the data.
- In clustering, there are no predictions made. Rather, clustering methods find the similarities between objects according to the object attributes and group the similar objects into clusters.
- Clustering techniques are utilized in marketing, economics, and various branches of science.

There are primarily two categories of clustering:

- Hierarchical clustering
- Partitioning clustering

Hierarchical clustering is further subdivided into:

- Agglomerative clustering
- Divisive clustering

Partitioning clustering is further subdivided into:

- K-Means clustering
- Fuzzy C-Means clustering

K-Means Clustering

Given a collection of objects each with n measurable attributes and a chosen value k of the number of clusters, the algorithm identifies the k clusters of objects based on the objects proximity to the centers of the k groups.

The algorithm is iterative with the centers adjusted to the mean of each cluster's n-dimensional vector of attributes

K-means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster.

In K-Means, each cluster is associated with a centroid. The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid**.**

K-Means Algorithm

1.Choose the value of k (i.e number of clusters) and the initial guesses for the centroids
2. Compute the distance from each data point to each centroid, and assign each point to the closest centroid
3.Compute the centroid of each newly defined cluster from step 2
4.Repeat steps 2 and 3 until the algorithm converges (no changes occur)

Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

- Centroids of newly formed clusters do not change
- Points remain in the same cluster
- Maximum number of iterations are reached

Distance Measure

Distance measure determines the similarity between two elements and influences the shape of clusters.

K-Means clustering supports various kinds of distance measures, such as:

- Euclidean distance measure
- Manhattan distance measure
- A squared euclidean distance measure
- Cosine distance measure

Determining the Number of Clusters

- The value of k can be chosen based on a reasonable guess or some predefined requirement.

- However, even then, it would be good to know how much better or worse having k clusters versus k – 1 or k + 1 clusters would be in explaining the structure of the data.

- Next, a heuristic using the Within Sum of Squares (WSS) metric is examined to determine a reasonably optimal value of k.

- In other words, WSS is the sum of the squares of the distances between each data point and the closest centroid.

- The term indicates the closest centroid that is associated with the ith point.

- If the points are relatively close to their respective centroids, the WSS is relatively small.

Application of K-Means

Some specific applications of k-means are

- image processing,
- medical, and
- customer segmentation

Reinforcement Learning

Reinforcement Learning(RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

The goal is to find a suitable action model that would maximize the total cumulative reward of the agent.

Q-learning

Q-Learning is a Reinforcement learning policy that will find the next best action, given a current state. It chooses this action at random and aims to maximize the reward.

Q-learning is a model-free, off-policy reinforcement learning that will find the best course of action, given the current state of the agent. Depending on where the agent is in the environment, it will decide the next action to be taken.

The objective of the model is to find the best course of action given its current state. To do this, it may come up with rules of its own or it may operate outside the policy given to it to follow. This means that there is no actual need for a policy, hence we call it off-policy.

Important Terms in Q-Learning

- States: The State, S, represents the current position of an agent in an environment.
- Action: The Action, A, is the step taken by the agent when it is in a particular state.
- Rewards: For every action, the agent will get a positive or negative reward.
- Episodes: When an agent ends up in a terminating state and can't take a new action.
- Q-Values: Used to determine how good an Action, A, taken at a particular state, S, is. Q (A, S).
- Temporal Difference: A formula used to find the Q-Value by using the value of current state and action and previous state and action

Algorithm

Start with a set of environmental states of the agent called as S
A set of possible actions that can take in those states, called as A
Value for each state/action pair that we'll call Q;