ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/359363908

Artificial Intelligence & Machine Learning Unit 4: Development of ML Model Question bank and its solution

Presentation · March 2022

DOI: 10.13140/RG.2.2.26552.83203

CITATIONS
0

READS
12,279

1 author:

Abhishek D. Patange
ABB
115 PUBLICATIONS **753** CITATIONS

SEE PROFILE

All content following this page was uploaded by Abhishek D. Patange on 20 March 2022.

The user has requested enhancement of the downloaded file.

Artificial Intelligence & Machine Learning

Course Code: 302049

Unit 4: Development of ML Model

Third Year Bachelor of Engineering (Choice Based Credit System)
Mechanical Engineering (2019 Course)
Board of Studies – Mechanical and Automobile Engineering, SPPU, Pune
(With Effect from Academic Year 2021-22)

Question bank and its solution

by

Abhishek D. Patange, Ph.D.
Department of Mechanical Engineering
College of Engineering Pune (COEP)

Unit 4: DEVELOPMENT OF ML MODEL

Syllabus:

Content	Theory	Mathematics	Numerical
• **Development of ML model**			
Problem identification	✓	✗	✗
Data Collection, Data pre-processing, Model Selection (C & R)	✓	✗	✗
Model training, Model evaluation, Hyper parameter Tuning, Predictions (C & R)	✓	✓	✓
• **Development of ML models to solve mechanical engineering problems**			

Note: 'C' stands for classification and 'R' stands for regression

Type of question and marks:

Type	Theory	Mathematics	Numerical
Marks	2 or 4 or 6 marks	4 marks	2 or 4 marks

Topic: Problem identification

Theory	Mathematics	Numerical
✓	✗	✗

Theory questions

1. What are four typical problems to be solved using machine learning approach?

- **Regression**

  If the prediction value tends to be a **continuous value** then it falls under Regression type problem in machine learning. Giving area name, size of land, etc. as features and predicting expected cost of the land.

- **Classification**

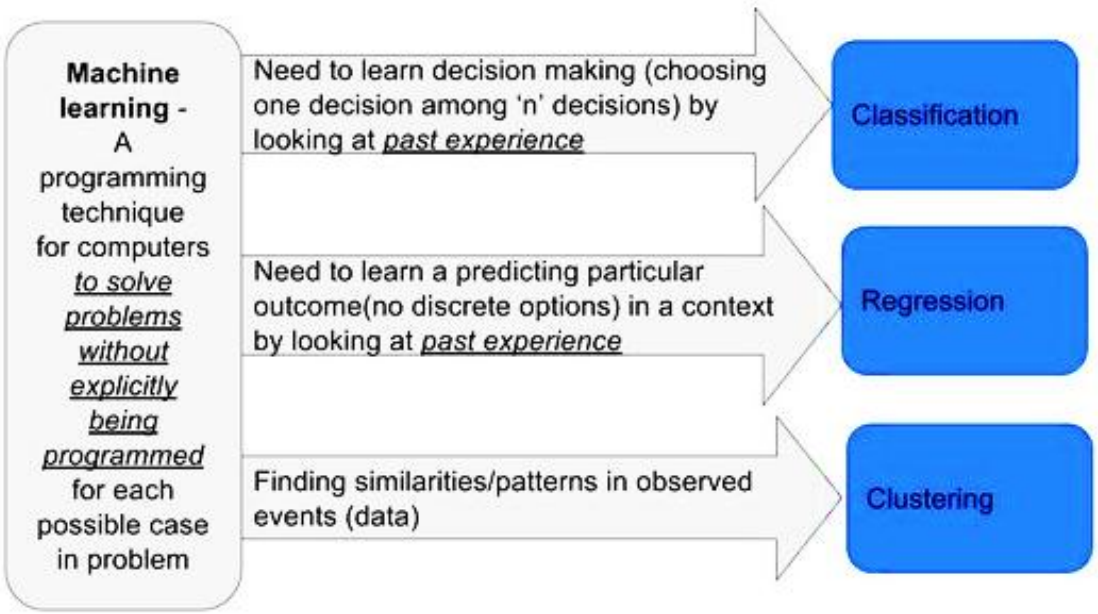
  If the prediction value tends to be **category/discrete** like yes/no , positive/negative , etc. then it falls under classification type problem in machine learning. Given a sentence predicting whether it is negative or positive review

- **Clustering**

  Grouping a set of points to given number of clusters. Given 3, 4, 8, 9 and number of clusters to be 2 then the ML system might divide the given set into cluster **1 - 3, 4** and cluster **2 - 8, 9**

- **Ranking**

  Used for **constructing a ranker** from a set of labelled examples. This example set consists of instance groups that can be scored with a given criteria. The ranking labels are { 0, 1, 2, 3, 4 } for each instance. The ranker is **trained to rank new instance groups with unknown scores** for each instance.



2. Represent classification, regression, and clustering on two-dimensional plane pictorially.

- Regression means the relationship between 2 "things" (one variable-dependent related to one variable-independent or groups of variable dependent against the group of independent as well as a combination 1:n variables, called multivariate regression). Regression "sees" relationship.
- Classification and clustering both manage groups of something according some criteria(s). Example is grouping by gender, age, some preferences. Difference is easy to see: In classification you define the factors that differentiate the population and put each individual in a specific "drawer" according to your grouping criteria.
- The criteria can be single (one unique factor of differentiation) or by a combination of factors, e.g. gender & birth city). You are classifying the sample.
- In clustering, the process senses differences based on value of variables and assign a specific "drawer" to each case of the sample. After this separation based on math and value of variables, the researcher will try to validate if the grouping has a human logical reason and, if possible, characterize it by human feeling.
- This is very tricky because all the process is essentially numbers because math is unable to "read" the label of each factor and assign human meaning.
- Let's see an example: consider you measure customer satisfaction using a group of variables (price, quality, service, delivery time, gender, age, education, ...) that is supposed to segment the sample in groups based on value of those variables.
- You can define how many groups you want to have, or some tools show it graphically in order to allow the researcher decide what configuration (by number of groups called clusters) would be convenient. Regression measures relationship, classification you define criteria of grouping and separate by that.

- Clustering the groups is defined by "distance" among sample cases and at the end, researcher looks for some meaning of that grouping. Regression, classification and clustering are based on sample content and the result reflects that sample.
- To extrapolate the conclusion to population requires validation according to the Level of Confidence you are willing to assume and additional math processes need to be done.

3. Why is 'clustering' not 'classification'? Give example.

- Usually **classification** is referred to as the problem of **observing data and deciding to which class each item belongs** (e.g. cat or dog).
- Typically the classes are known in advance - something we care about.
- **Clustering** is typically referred to when **we have data but no labels that are associated with each item** (i.e. no predefined notion of cats and dogs).
- In this case the problem is to **cluster the items in to groups** so that items **in each group "resemble each other".**
- It can be that a particular algorithm will classify, for example, pet images into cats and dogs but it may also end up clustering them to other groups (based on colour, pose, size or any other characteristic).
- The key is that **clustering is typically done in the unsupervised domain**, that is: there are no predefined classes known in advance.

Examples of classification and clustering are as follows.

Classification

- Email classification: Spam or non-Spam
- Sanction loan to customer: Yes if he is capable of paying EMI for the sanctioned loan amount. No if he can't
- Cancer tumour cells identification: Is it critical or non-critical?
- Sentiment analysis of tweets: Is the tweet positive or negative or neutral
- Classification of news: Classify the news into one of predefined classes - Politics, Sports, Health etc.

Clustering

- Marketing: Discover customer segments for marketing purposes
- Biology: Classification among different species of plants and animals
- Libraries: Clustering different books on the basis of topics and information
- Insurance: Acknowledge the customers, their policies and identifying the frauds
- City Planning: Make groups of houses and to study their values based on their geographical locations and other factors.
- Earthquake studies: Identify dangerous zones

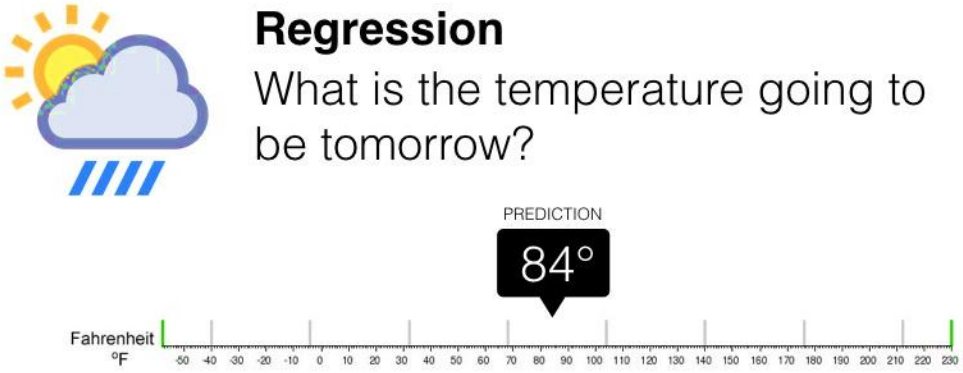
4. Differentiate between clustering and classification.

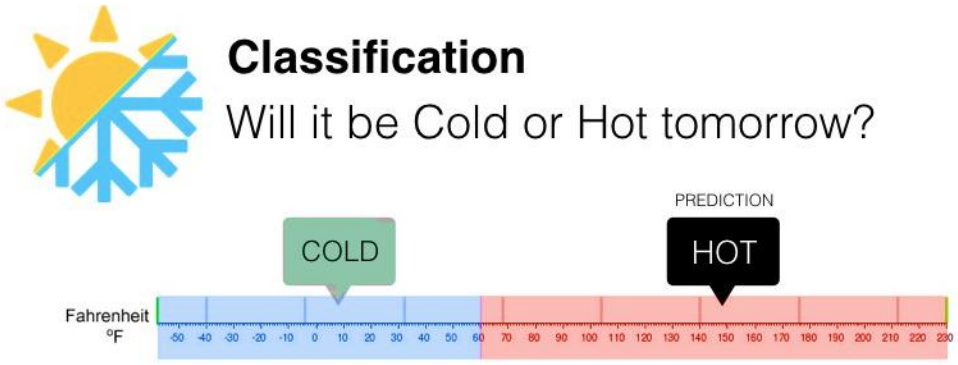
Parameter	CLASSIFICATION	CLUSTERING
Type	used for supervised learning	used for unsupervised learning
Basic	process of classifying the input instances based on their corresponding class labels	grouping the instances based on their similarity without the help of class labels
Need	it has labels so there is need of training and testing dataset for verifying the model created	there is no need of training and testing dataset
Complexity	more complex as compared to clustering	less complex as compared to classification
Uses	It uses algorithms to categorize the new data as per the observations of the training set.	It uses statistical concepts in which the data set is divided into subsets with the same features.
Objective	Its objective is to find which class a new object belongs to form the set of predefined classes.	Its objective is to group a set of objects to find whether there is any relationship between them.
Example Algorithms	Logistic regression, Naive Bayes classifier, Support vector machines, etc.	k-means clustering algorithm, Fuzzy c-means clustering algorithm, Gaussian (EM) clustering algorithm, etc.

5. Differentiate between regression and classification.

Regression	Classification
In Regression, the output variable must be of continuous nature or real value.	In Classification, the output variable must be a discrete value.
The task of the regression algorithm is to map the input value (x) with the continuous output variable(y).	The task of the classification algorithm is to map the input value(x) with the discrete output variable(y).
Regression Algorithms are used with continuous data.	Classification Algorithms are used with discrete data.
In Regression, we try to find the best fit line, which can predict the output more accurately.	In Classification, we try to find the decision boundary, which can divide the dataset into different classes.

Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, etc.	Classification Algorithms can be used to solve classification problems such as Identification of spam emails, Speech Recognition, Identification of cancer cells, etc.
The regression Algorithm can be further divided into Linear and Non-linear Regression.	The Classification algorithms can be divided into Binary Classifier and Multi-class Classifier.
Method of evaluation by measurement of root mean square error	Method of evaluation by measuring accuracy
Nature of the predicted data Ordered	Nature of the predicted data Unordered





6. Explain terminology of understanding ML based classification and regression model.

- **Algorithm** = A method, function, or series of instructions used to generate a machine learning model. Examples include linear regression, decision trees, support vector machines, and neural networks.
- **Attribute** = A quality describing an observation (e.g. color, size, weight). In Excel terms, these are column headers.
- **Categorical Variables** = Variables with a discrete set of possible values. Can be ordinal (order matters) or nominal (order doesn't matter).

- **Classification =** Predicting a categorical output.
- **Binary classification** = predicts one of two possible outcomes (e.g. is the email spam or not spam?)
- **Multi-class classification =** predicts one of multiple possible outcomes (e.g. is this a photo of a cat, dog, horse or human?)
- **Classification Threshold =** The lowest probability value at which we're comfortable asserting a positive classification. For example, if the predicted probability of being diabetic is > 50%, return True, otherwise return False.
- **Classifier** =It is an algorithm that is used to map the input data to a specific category.
- **Classification Model =** The model predicts or draws a conclusion to the input data given for training, it will predict the class or category for the data.
- **Binary Classification** = It is a type of classification with two outcomes, for e.g. – either true or false.
- **Multi-Class Classification** = The classification with more than two classes, in multi-class classification each sample is assigned to one and only one label or target.
- **Multi-label Classification** = This is a type of classification where each sample is assigned to a set of labels or targets.
- **Clustering =** Unsupervised grouping of data into buckets.
- **Confusion Matrix =** Table that describes the performance of a classification model by grouping predictions into 4 categories.
- **True Positives**: we *correctly* predicted they do have diabetes
- **True Negatives**: we *correctly* predicted they don't have diabetes
- **False Positives**: we *incorrectly* predicted they do have diabetes (Type I error)
- **False Negatives**: we *incorrectly* predicted they don't have diabetes (Type II error)
- **Continuous Variables =** Variables with a range of possible values defined by a number scale (e.g. sales, lifespan).
- **Convergence =** A state reached during the training of a model when the loss changes very little between each iteration.
- **Epoch =** An epoch describes the number of times the algorithm sees the entire data set.
- **Feature =** With respect to a dataset, a feature represents an attribute and value combination. Color is an attribute. "Color is blue" is a feature. In Excel terms, features are similar to cells. The term feature has other definitions in different contexts.
- **Feature Selection =** Feature selection is the process of selecting relevant features from a data-set for creating a Machine Learning model.

- **Feature Vector** = A list of features describing an observation with multiple attributes. In Excel we call this a row.
- **Hyperparameters** = Hyperparameters are higher-level properties of a model such as how fast it can learn (learning rate) or complexity of a model. The depth of trees in a Decision Tree or number of hidden layers in a Neural Networks are examples of hyper parameters.
- **Instance** = A data point, row, or sample in a dataset. Another term for observation.
- **Label** = The "answer" portion of an observation in supervised learning. For example, in a dataset used to classify flowers into different species, the features might include the petal length and petal width, while the label would be the flower's species.
- **Model** = A data structure that stores a representation of a dataset (weights and biases). Models are created/learned when you train an algorithm on a dataset.
- **Neural Networks** = Neural Networks are mathematical algorithms modeled after the brain's architecture, designed to recognize patterns and relationships in data.
- **Normalization** = Restriction of the values of weights in regression to avoid overfitting and improving computation speed.
- **Noise** = Any irrelevant information or randomness in a dataset which obscures the underlying pattern.
- **Observation** = A data point, row, or sample in a dataset. Another term for instance.
- **Outlier** = An observation that deviates significantly from other observations in the dataset.
- **Overfitting** = Overfitting occurs when your model learns the training data too well and incorporates details and noise specific to your dataset. You can tell a model is overfitting when it performs great on your training/validation set, but poorly on your test set (or new real-world data).
- **Regression** = Predicting a continuous output (e.g. price, sales).
- **Supervised Learning** = Training a model using a labeled dataset.
- **Test Set** = A set of observations used at the end of model training and validation to assess the predictive power of your model. How generalizable is your model to unseen data?
- **Training Set** = A set of observations used to generate machine learning models.
- **Underfitting** = Underfitting occurs when your model over-generalizes and fails to incorporate relevant variations in your data that would give your model more predictive power. You can tell a model is underfitting when it performs poorly on both training and test sets.

- **Unsupervised Learning** = Training a model to find patterns in an unlabeled dataset (e.g. clustering).
- **Validation Set** = A set of observations used during model training to provide feedback on how well the current parameters generalize beyond the training set. If training error decreases but validation error increases, your model is likely overfitting and you should pause training.

7. Enlist steps involved in development of classification model.

Following are the steps to be considered in development of classification model.

1 - Data Collection

- The quantity & quality of your data dictate how accurate our model is
- The outcome of this step is generally a representation of data which we will use for training
- Using experimental data, data generated by simulations, pre-collected data, by way of datasets from Kaggle, UCI, etc., still fits into this step

2 - Data Preparation

- Wrangle data and prepare it for training
- Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, and data type conversions, etc.)
- Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data
- Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis
- Split into training and evaluation sets

3 - Choose a Model

- Different algorithms are for different tasks; choose the right one

4 - Train the Model

- The goal of training is to answer a question or make a prediction correctly as often as possible
- Linear regression example: algorithm would need to learn values for m (or W) and b (x is input, y is output)
- Each iteration of process is a training step

5 - Evaluate the Model

- Uses some metric or combination of metrics to "measure" objective performance of model
- Test the model against previously unseen data

- This unseen data is meant to be somewhat representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not)
- Good train/evaluation split? 80/20, 70/30, or similar, depending on domain, data availability, dataset particulars, etc.

6 – Hyper parameter Tuning

- This step refers to *hyperparameter* tuning, which is an "artform" as opposed to a science
- Tune model parameters for improved performance
- Simple model hyperparameters may include: number of training steps, learning rate, initialization values and distribution, etc.

7 - Make Predictions

- Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model; a better approximation of how the model will perform in the real world

8. Enlist steps involved in development of regression model.

- Regression Analysis is an analytical process whose end goal is to understand the inter-relationships in the data and find as much useful information as possible.
- According to the book, there are a number of steps which are loosely detailed below.

1 - Problem definition

- The very first step is to, off course; define the problem we are trying to solve. Perhaps a business question that needs to be answered or simply a prediction we want to make based on some set of data. In this stage we must know the target variable and the attributes we presume affects the target variable. This would be later analysed to judge its credibility. For the sake of our discussion let's take the Titanic Dataset as an example.
- In this dataset we have data of about 900 passengers. The question or the problem we must solve is predicting which passenger likely survived the tragedy given their data.

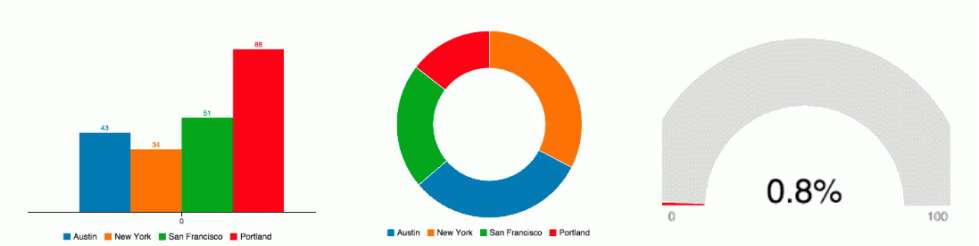
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

A look at the Titanic Dataset

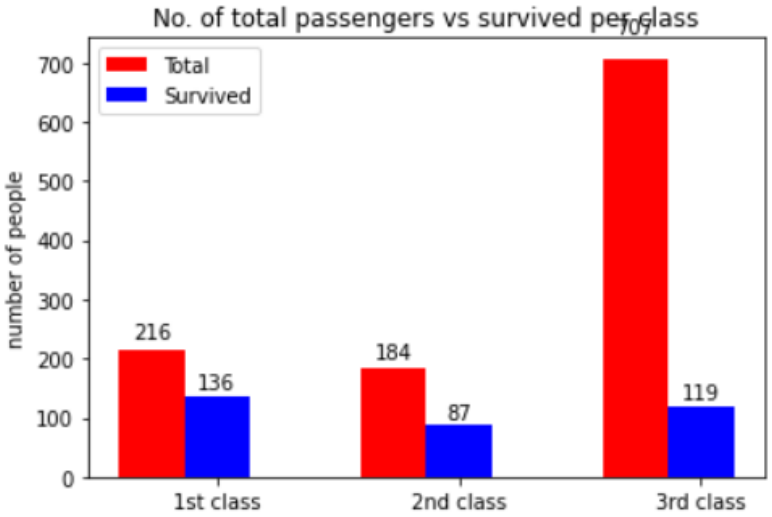
- So now we know, that 'Survival' is the response variable but of the 10 attributes given for each passenger, how do we determine which of these predictor variables affect the result? That's where data analysis comes in .

2 - Analyse Data

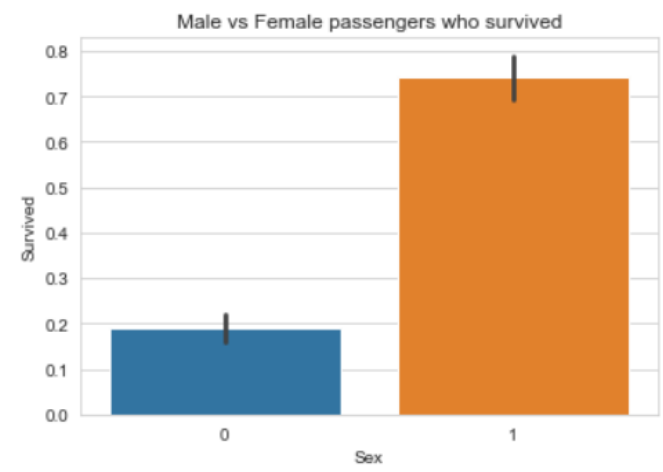
The **key** is to have visual representations of our data so we can better understand the 'inter-relationships' of the variables and likely so, the book I was referring to earlier, highly recommends using visual tools to make the EDA(Exploratory Data Analysis) process easier. For the afore-mentioned dataset, we could try answering a number of things that might give us a better understanding of the problem at hand. What's the survival rate of passengers from each class?



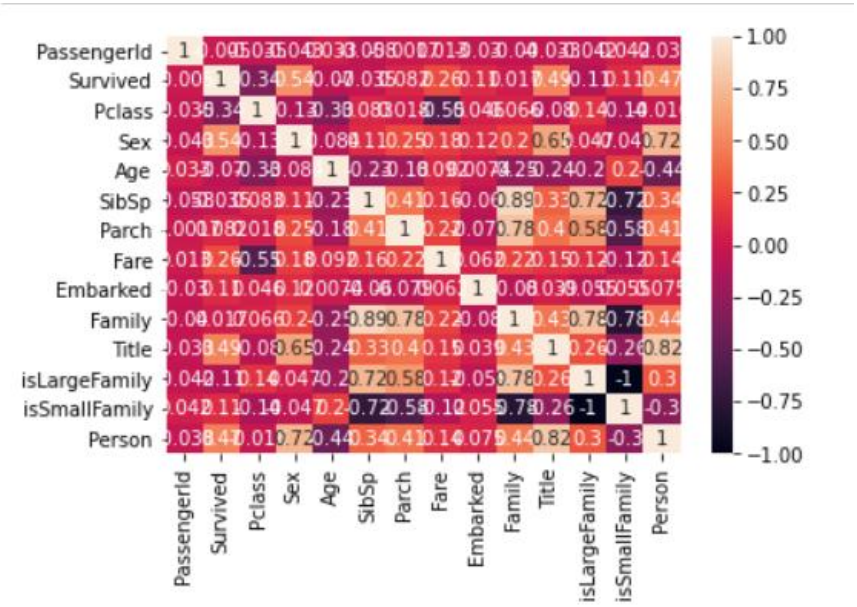
Graphs and charts



How about the survival rate based on gender?



How about the Correlation of all the attributes?



Heatmap showing correlation

Finding correlation is an important step as it allows us to roughly pick the attributes that have a relation with the response variable. We are most likely to pick the attributes/variables that show a positive correlation with respect to the target variable. From this section we can deduce that plotting graphs are vital for the next step which is choosing a model. Graphs before model fitting can range from histograms, boxplots, root and leaf display, scatter plots etc.

3 - Model Selection

Based on the data, we are to pick a suitable model or regression equation. You may be familiar with many such models like Linear Regression, Support Vector Machine, Random Forest etc. The task in this step is to pick one that we assume will express the relationships of our data in the best way possible. This assumption can be later accepted or refuted based on analysis after fitting the model.

4 - Model Fitting

For simplicity's sake, let's consider linear regression. ***Y= mx+c***. We have the data, we have a model. At this stage we are going to ***train*** the model on the given dataset but what of the parameters of this equation? We must estimate these parameters when fitting the model however they can be optimised with many algorithms. Perhaps this is when terms like 'Gradient Descent' or 'Adam optimiser' rings a bell. The purpose of an optimiser is simply to update the values in every iteration of training so we can minimise loss or error. This is the part where our model learns to correct itself and provide a best fitting solution or model that would likely have high accuracy. For a simple model like linear regression, we can use **Least Squares method** to estimate the parameters '**m (slope)'** and **'c (y-intercept)'** to get the best fit line that crosses through most of the data points. The least squares method basically

minimizes the sum of the square of the errors as small as possible given that no outliers are present in the data.

5 - Model evaluation

Final step is model evaluation - measuring and criticising exactly how well is the model fitting the data points. We run the model on the test data and check to see how accurately it was able to predict the output values. Now, there are a number of measures to check this as discussed below:

i) We can find **RMSE** (root mean squared error) of the actual Y values and predicted Y values. There are other variations of it that can be explored.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

Formula for RMSE

ii) We can calculate **R-squared** value which measures the goodness of fit or varaince within a range of 0 to 1 where ideal value is 1.

$$R^2 = \frac{SSR}{SST}$$

$$SSR = \sum_i (\hat{y}_i - \bar{y})^2$$

$$SST = \sum_i (y_i - \bar{y})^2$$

Where,

- SSR is Sum of Squared Regression also known as variation explained by the model
- SST is Total variation in the data also known as sum of squared total
- y_i is the y value for observation i
- y_bar is the mean of y value
- y_bar_hat is predicted value of y for observation i

Formula to find R-squared value

iii) We can perform cross validation to asses which model among a few chosen performed the best for our given problem.

iv) Finding statistical significance of parameters. This involves stating a hypothesis, a null hypothesis and an alpha level (probability of error level). An example is **Chi-squared Test** which tests if there is any relation between two variables.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

χ^2 = chi squared

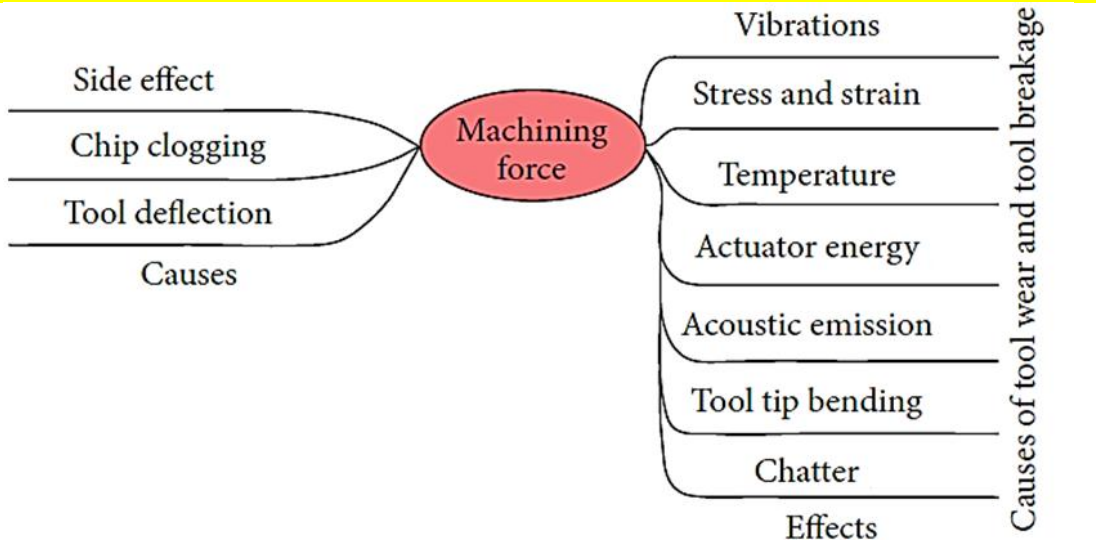
O_i = observed value

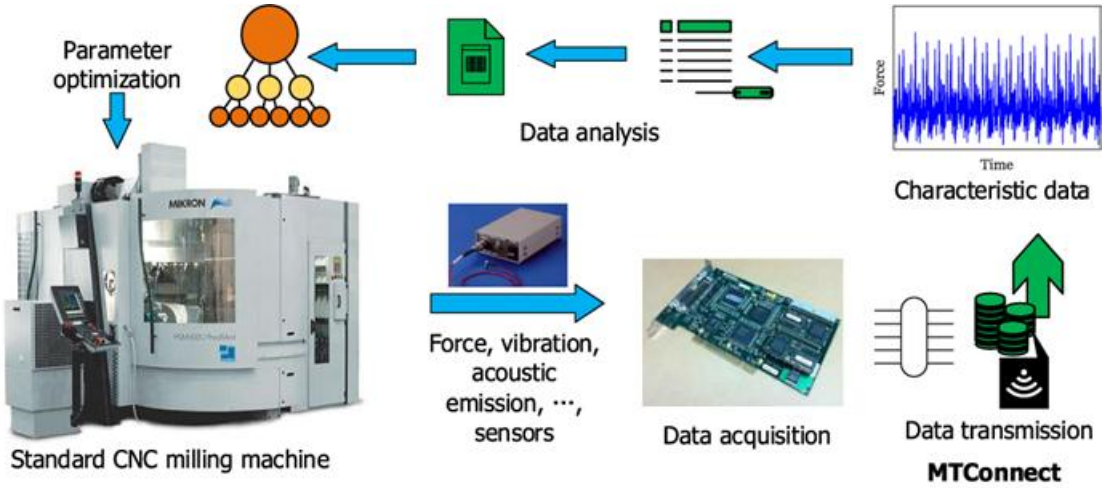
E_i = expected value

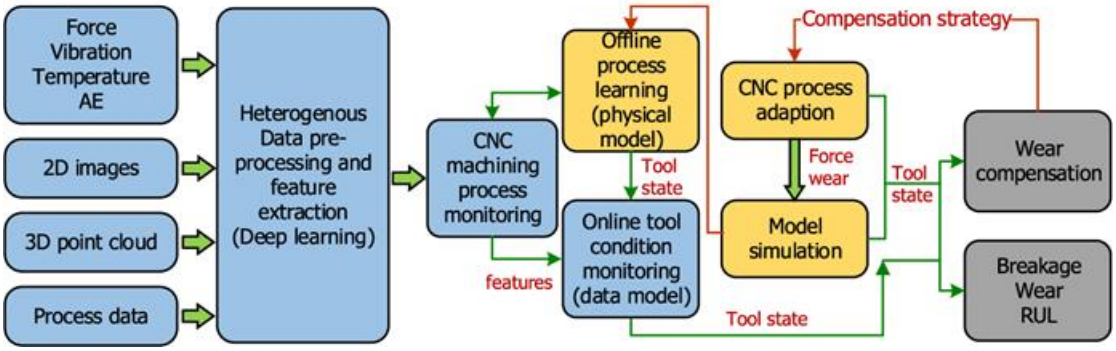
Formula for Chi-Square Statistical Test

There are many other methods, some more complex than others but these are usually a good place to start. Based on this analysis, the model is updated and perfected after which it can be used for its intended purpose.

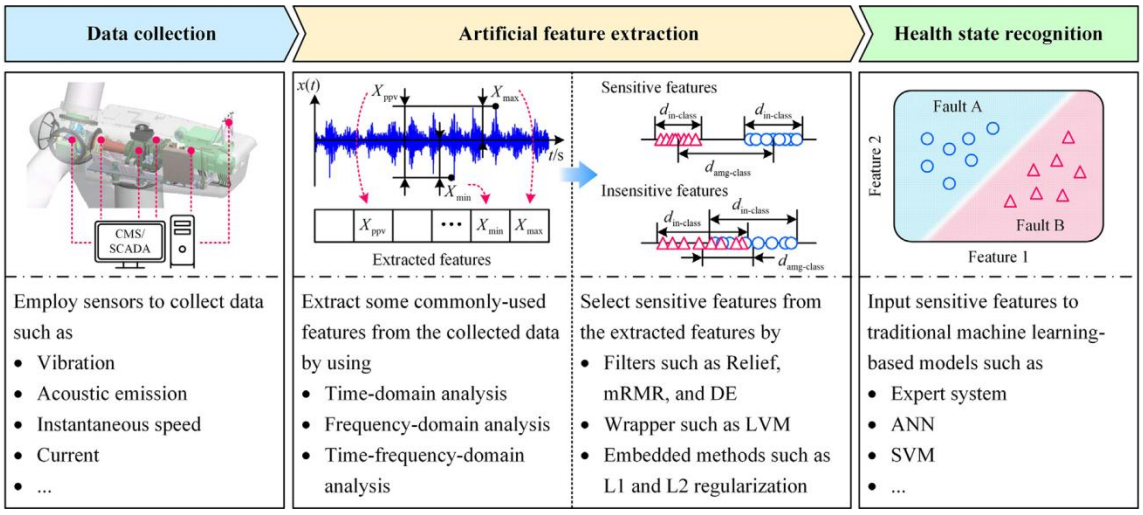
9. What are the sources of the data that is needed for training the classification model for identifying cutting tool state in milling/drilling/lathe?

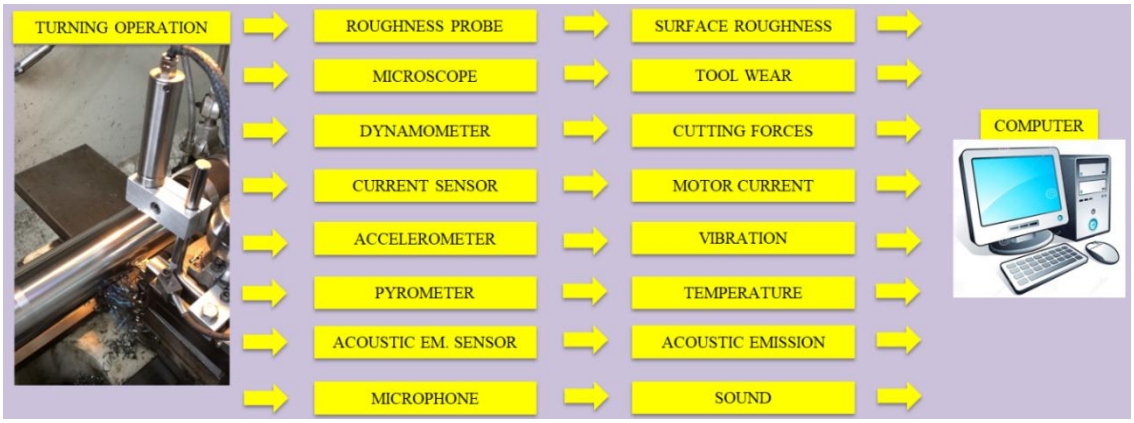


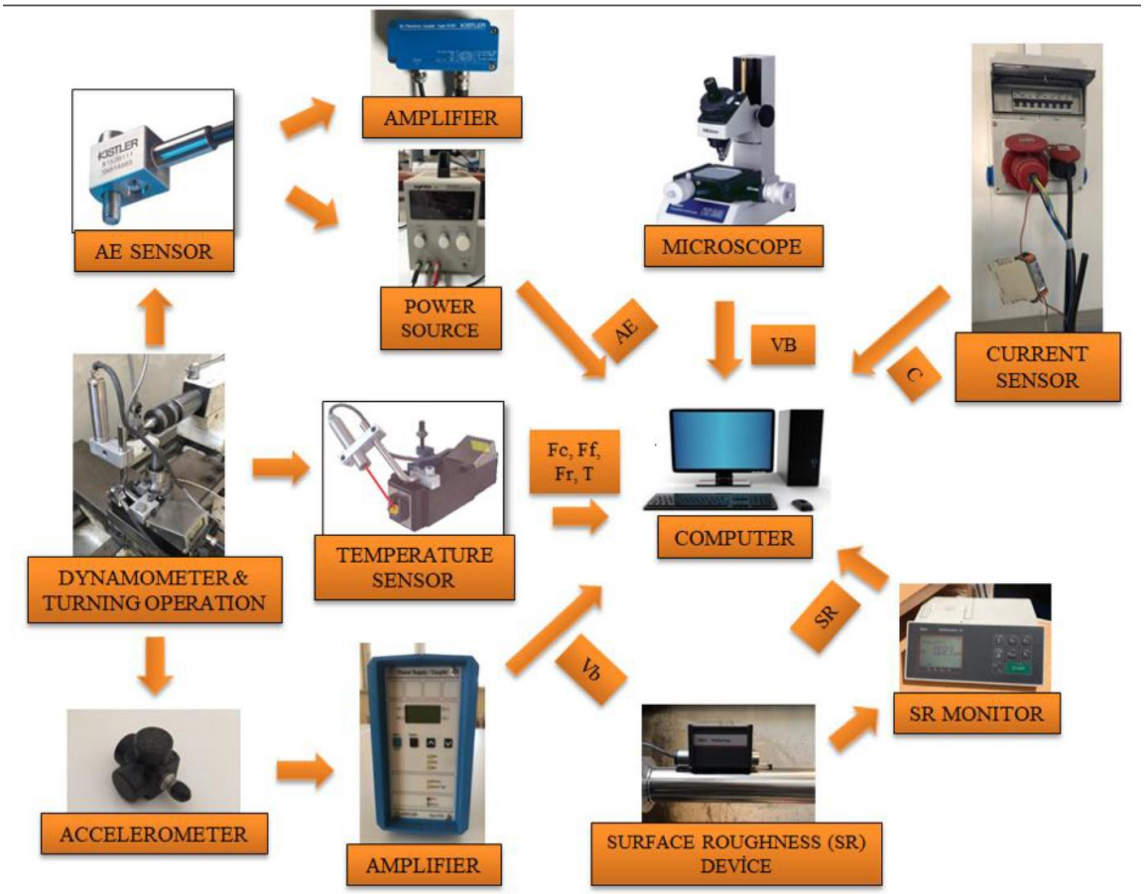




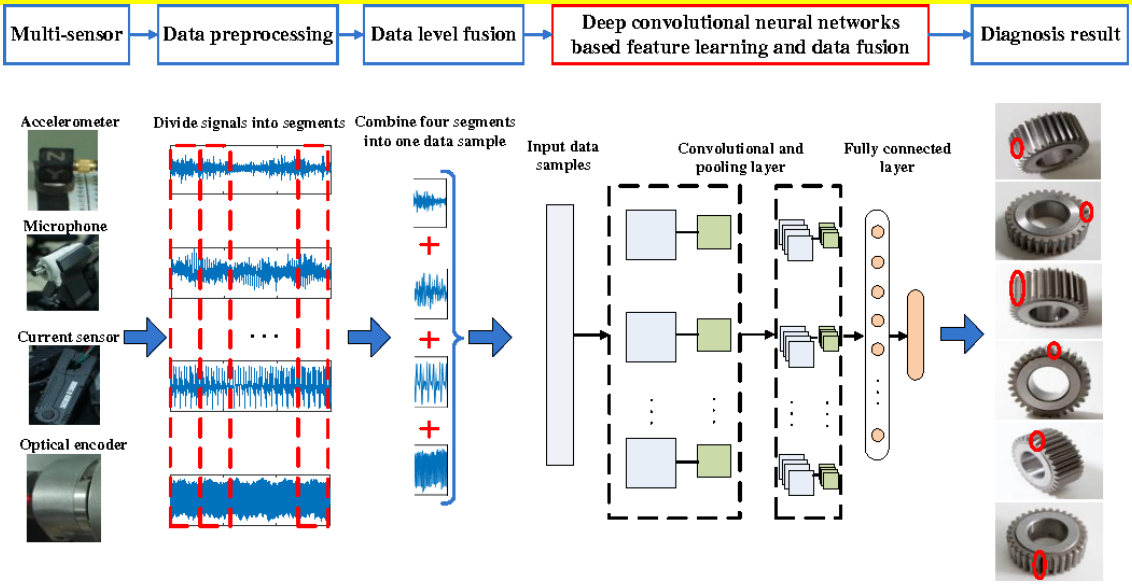
QUESTION BANK FOR UNIT 4: DEVELOPMENT OF ML MODEL



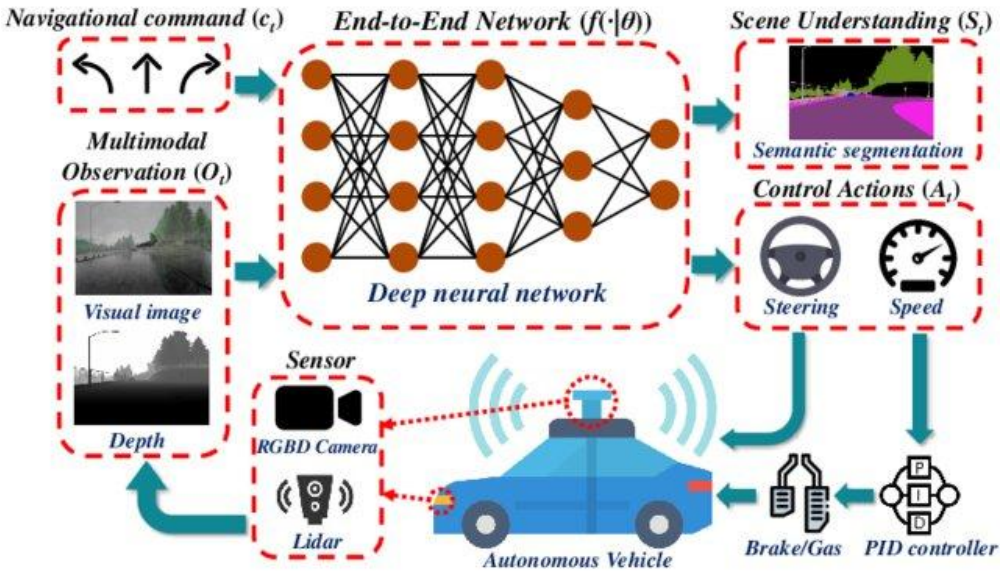




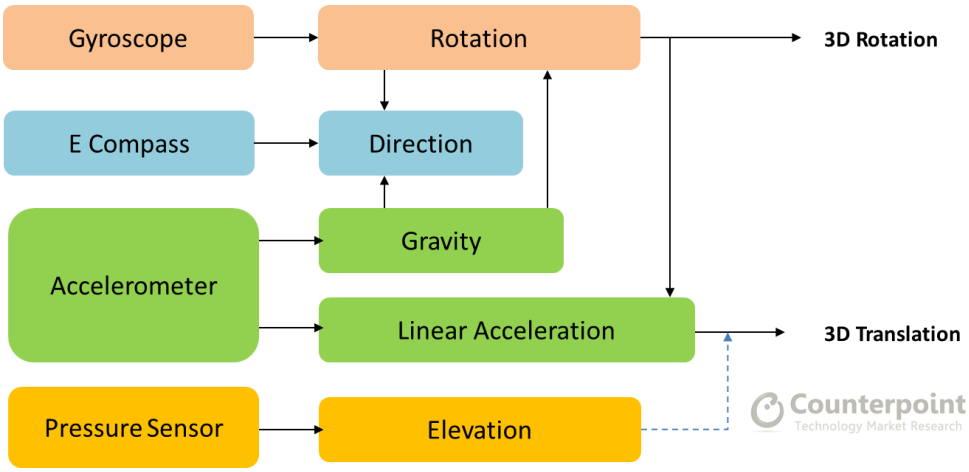
10. What are the sources of the data that is needed for training the classification model for condition monitoring of bearings, gears, rotating elements?



11. What are the sources of the data that is needed for training the ML model for End-to-End Autonomous Driving With Scene Understanding?



12. What are the sources of the data that is needed for training the ML model for accurately predicting the device's position in three dimensions (3D motion sensing using Sensor Fusion)?



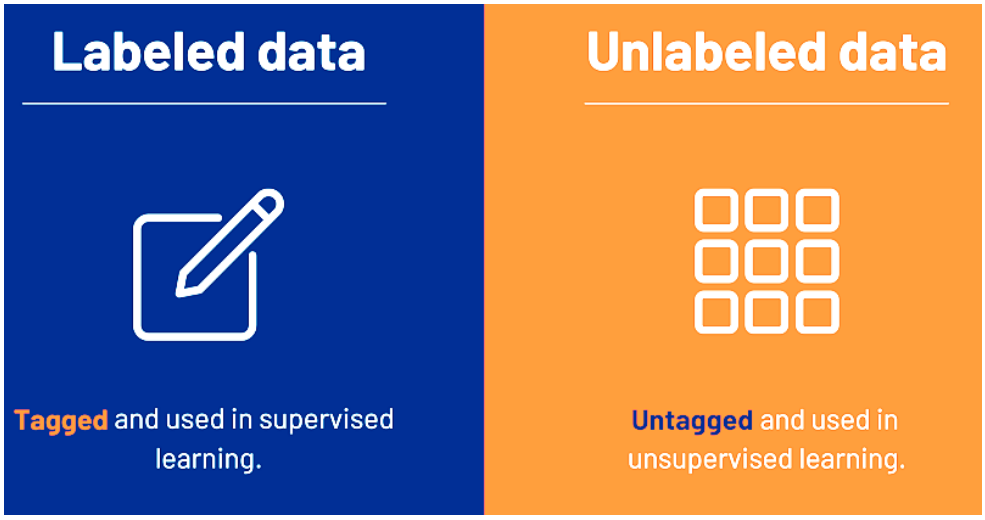
13. You have given a task of developing a classification model for identifying cutting tool state in milling/drilling/lathe. So initially you will collect the data corresponds to tool state either as healthy or faulty. So what are the possible sources of the data that is needed for said application?

Questions 9-12 can be rephrased as question 13. So it is same.

14. What is training data? What is labeled data? What is unlabeled data? What are key steps involved in developing training data?

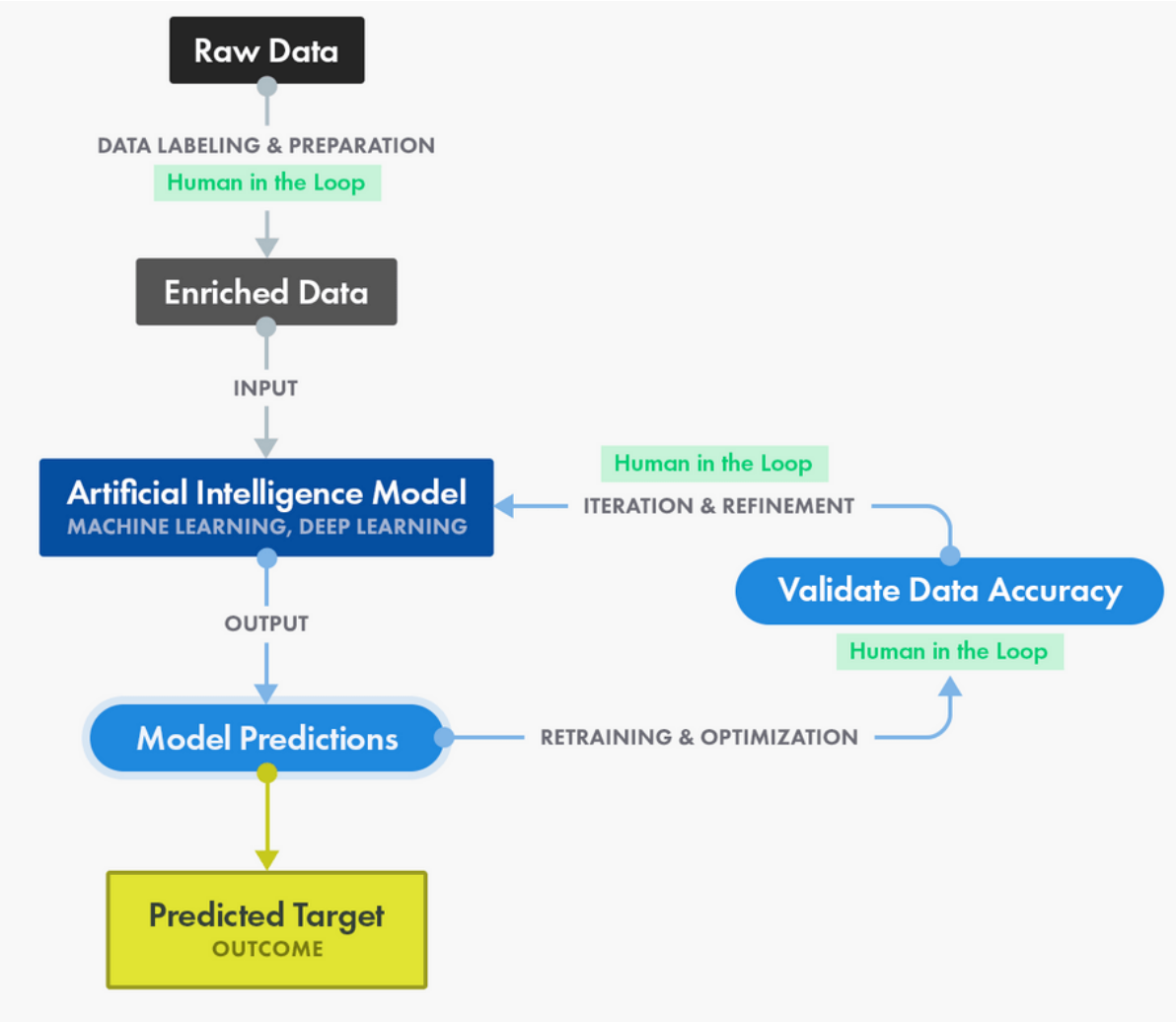
- Machine learning models are as good as the data they're trained on. Without high-quality training data, even the most efficient machine learning algorithms will fail to perform.
- The need for quality, accurate, complete, and relevant data starts early on in the training process.
- Only if the algorithm is fed with good training data can it easily pick up the features and find relationships that it needs to predict down the line.
- More precisely, quality **training data** is the most significant aspect of machine learning (and artificial intelligence) than any other.
- If you introduce the machine learning (ML) algorithms to the right data, you're setting them up for accuracy and success.
- Training data is the initial dataset used to train machine learning algorithms. Models create and refine their rules using this data. It's a set of data samples used to fit the parameters of a machine learning model to training it by example.

- Training data is also known as training dataset, learning set, and training set. It's an essential component of every machine learning model and helps them make accurate predictions or perform a desired task.
- Simply put, training data builds the machine learning model. It teaches what the expected output looks like. The model analyzes the dataset repeatedly to deeply understand its characteristics and adjust itself for better performance.
- In a broader sense, training data can be classified into two categories: **labeled data** and **unlabeled data**.



- **Labeled data** is a group of data samples tagged with one or more meaningful labels. It's also called annotated data, and its labels identify specific characteristics, properties, classifications, or contained objects.
- For example, the images of fruits can be tagged as *apples, bananas,* or *grapes*.
- Labeled training data is used in supervised learning. It enables ML models to learn the characteristics associated with specific labels, which can be used to classify newer data points. In the example above, this means that a model can use labeled image data to understand the features of specific fruits and use this information to group new images.
- Data labeling or annotation is a time-consuming process as humans need to tag or label the data points. Labeled data collection is challenging and expensive. It isn't easy to store labeled data when compared to unlabeled data.
- As expected, **unlabeled data** is the opposite of labeled data. It's raw data or data that's not tagged with any labels for identifying classifications, characteristics, or properties. It's used in unsupervised machine learning, and the ML models have to find patterns or similarities in the data to reach conclusions.

- Going back to the previous example of *apples*, *bananas*, and *grapes*, in unlabeled training data, the images of those fruits won't be labeled. The model will have to evaluate each image by looking at its characteristics, such as color and shape.
- After analyzing a considerable number of images, the model will be able to differentiate new images (new data) into the fruit types of *apples*, *bananas*, or *grapes*. Of course, the model wouldn't know that the particular fruit is called an apple. Instead, it knows the characteristics needed to identify it.
- There are hybrid models that use a combination of supervised and unsupervised machine learning.



15. How does training data used in machine learning?

- Unlike machine learning algorithms, traditional programming algorithms follow a set of instructions to accept input data and provide output. They don't rely on historical data, and every action they make is rule-based. This also means that they don't improve over time, which isn't the case with machine learning.

- For machine learning models, historical data is fodder. Just as humans rely on past experiences to make better decisions, ML models look at their training dataset with past observations to make predictions.
- Predictions could include classifying images as in the case of image recognition, or understanding the context of a sentence as in natural language processing (NLP).
- Think of a data scientist as a teacher, the machine learning algorithm as the student, and the training dataset as the collection of all textbooks.
- The teacher's aspiration is that the student must perform well in exams and also in the real world. In the case of ML algorithms, testing is like exams. The textbooks (training dataset) contain several examples of the type of questions that'll be asked in the exam.
- Of course, it won't contain all the examples of questions that'll be asked in the exam, nor will all the examples included in the textbook will be asked in the exam. The textbooks can help prepare the student by teaching them what to expect and how to respond.
- No textbook can ever be fully complete. As time passes, the kind of questions asked will change, and so, the information included in the textbooks needs to be changed. In the case of ML algorithms, the training set should be periodically updated to include new information.
- In short, training data is a textbook that helps data scientists give ML algorithms an idea of what to expect. Although the training dataset doesn't contain all possible examples, it'll make algorithms capable of making predictions.

16. What makes training data good?

High-quality data translates to accurate machine learning models. Low-quality data can significantly affect the accuracy of models, which can lead to severe financial losses. It's almost like giving a student textbook containing wrong information and expecting them to excel in the examination. The following are the four primary traits of quality training data.

- **Relevant**

The data needs to be relevant to the task at hand. For example, if you want to train a computer vision algorithm for autonomous vehicles, you probably won't require images of fruits and vegetables. Instead, you would need a training dataset containing photos of roads, sidewalks, pedestrians, and vehicles.

- **Representative**

The AI training data must have the data points or features that the application is made to predict or classify. Of course, the dataset can never be absolute, but it must have at least the attributes the AI application is meant to recognize. For example, if the model is meant to recognize faces within images, it must be fed with diverse data containing people's faces

from various ethnicities. This will reduce the problem of AI bias, and the model won't be prejudiced against a particular race, gender, or age group.

- **Uniform**

All data should have the same attribute and must come from the same source. Suppose your machine learning project aims to predict churn rate by looking at customer information. For that, you'll have a customer information database that includes customer name, address, number of orders, order frequency, and other relevant information. This is historical data and can be used as training data. One part of the data can't have additional information, such as age or gender. This will make training data incomplete and the model inaccurate. In short, uniformity is a critical aspect of quality training data.

- **Comprehensive**

Again, the training data can never be absolute. But it should be a large dataset that represents the majority of the model's use cases. The training data must have enough examples that'll allow the model to learn appropriately. It must contain real-world data samples as it will help train the model to understand what to expect. If you're thinking of training data as values placed in large numbers of rows and columns, sorry, you're wrong. It could be any data type like text, images, audio, or videos.

17. What affects training data quality?

Humans are highly social creatures, but there are some prejudices that we might have picked as children and require constant conscious effort to get rid of. Although unfavourable, such biases may affect our creations, and machine learning applications are no different. For ML models, training data is the only book they read. Their performance or accuracy will depend on how comprehensive, relevant, and representative the very book is. That being said, three factors affect the quality of training data:

- **People:** The people who train the model have a significant impact on its accuracy or performance. If they're biased, it'll naturally affect how they tag data and, ultimately, how the ML model functions.
- **Processes:** The data labeling process must have tight quality control checks in place. This will significantly increase the quality of training data.
- **Tools:** Incompatible or outdated tools can make data quality suffer. Using robust data labeling software can reduce the cost and time associated with the process.

18. How much training data is enough?

- There isn't a specific answer to how much training data is enough training data. It depends on the algorithm you're training – its expected outcome, application, complexity, and many other factors.

- Suppose you want to train a text classifier that categorizes sentences based on the occurrence of the terms "cat" and "dog" and their synonyms such as "kitty," "kitten," "pussycat," "puppy," or "doggy". This might not require a large dataset as there are only a few terms to match and sort.
- But, if this was an image classifier that categorized images as "cats" and "dogs," the number of data points needed in the training dataset would shoot up significantly. In short, many factors come into play to decide what training data is enough training data.
- The amount of data required will change depending on the algorithm used.
- For context, deep learning, a subset of machine learning, requires millions of data points to train the artificial neural networks (ANNs). In contrast, machine learning algorithms require only thousands of data points. But of course, this is a far-fetched generalization as the amount of data needed varies depending on the application.
- The more you train the model, the more accurate it becomes. So it's always better to have a large amount of data as training data.

Garbage in, garbage out

- The phrase "garbage in, garbage out" is one of the oldest and most used phrases in data science. Even with the rate of data generation growing exponentially, it still holds true.
- The key is to feed high-quality, representative data to machine learning algorithms. Doing so can significantly enhance the accuracy of models. Good quality training data is also crucial for creating unbiased machine learning applications.

19. How should you split up a dataset into test and training sets?

- When we are working on the model development, we need to train it and test it on the same dataset. Since it is challenging to possess a vast number of data while the model is in the development phase, the most obvious answer is to split the data into two separate sets, out of which one will be for training and the other will be testing.

A. Splitting the data set into a training set and test set:

The two conditions that need to be taken care of before proceeding with the splitting of the dataset:

- The test set needs to be large to give statistically essential outputs.
- The characteristics of the training set and test set should be similar.

Therefore, after the satisfaction of the above two conditions, the ultimate goal should be to develop a model that can easily perform functions with the new dataset.

B. Validation of the trained model over the test data.

The model should not train over the test data. Many times good results on evaluation metrics are an indication that inadvertently, you are training on test data.

20. What is test data?

- A test set in machine learning is a secondary (or tertiary) data set that is used to test a machine learning program after it has been trained on an initial training data set. The idea is that predictive models always have some sort of unknown capacity that needs to be tested out, as opposed to analysed from a programming perspective.
- A test set is also known as a test data set or test data.

21. What is validation data?

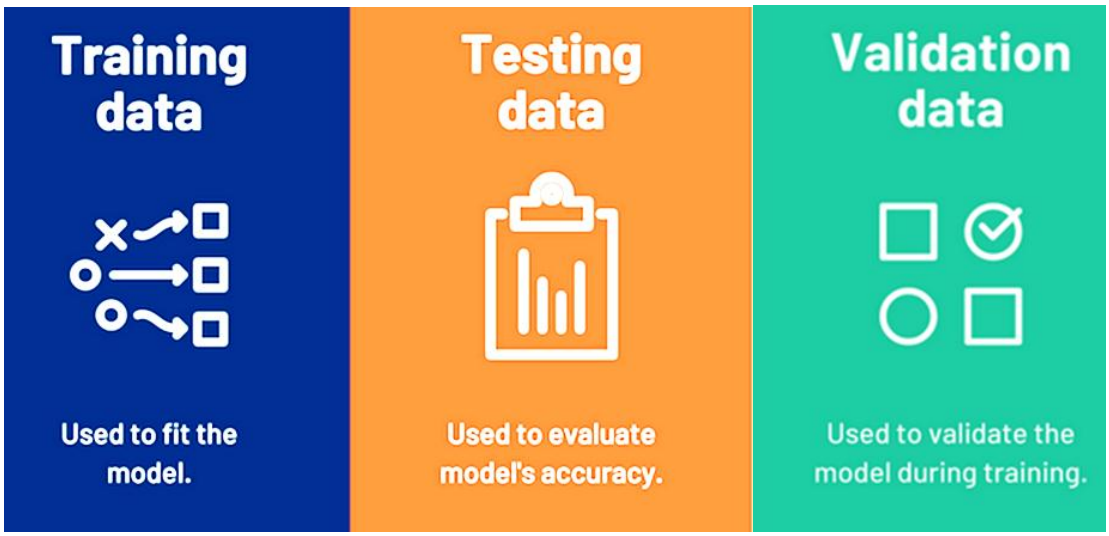
- In machine learning, a validation set is used to "tune the parameters" of a classifier. The validation test evaluates the program's capability according to the variation of parameters to see how it might function in successive testing.
- The validation set is also known as a validation data set, development set or dev set.

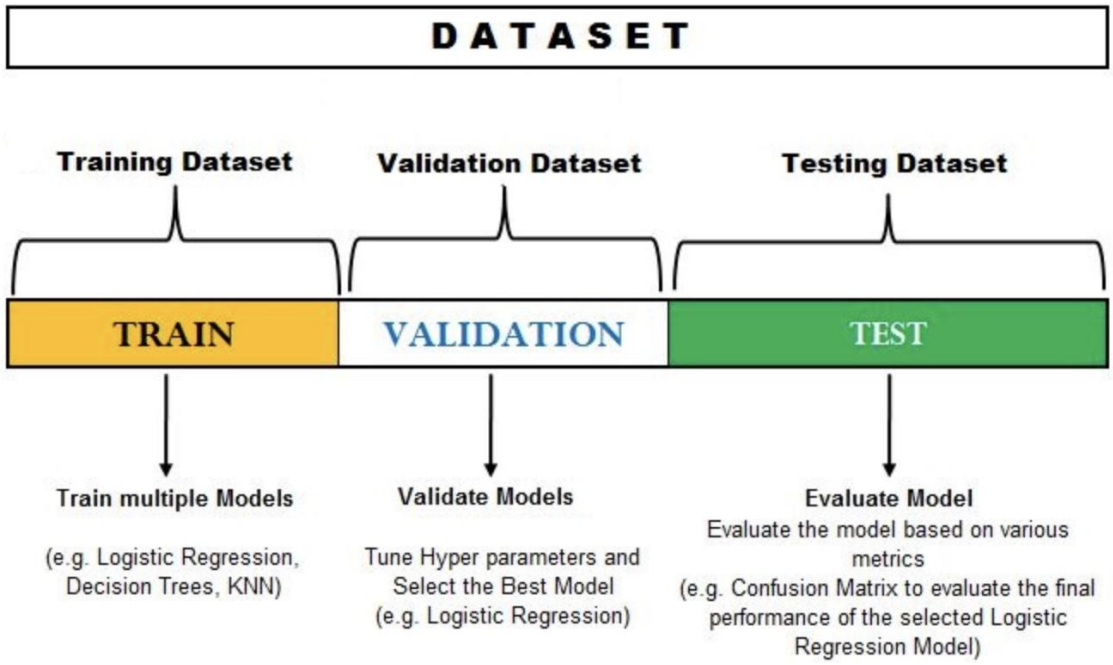
22. Compare Training data vs. test data vs. validation data.

- **Training data** is used in model training, or in other words, it's the data used to fit the model. On the contrary, **test data** is used to evaluate the performance or accuracy of the model. It's a sample of data used to make an unbiased evaluation of the final model fit on the training data.
- A training dataset is an initial dataset that teaches the ML models to identify desired patterns or perform a particular task. A testing dataset is used to evaluate how effective the training was or how accurate the model is.
- Once an ML algorithm is trained on a particular dataset and if you test it on the same dataset, it's more likely to have high accuracy because the model knows what to expect. If the training dataset contains all possible values the model might encounter in the future, all well and good.
- But that's never the case. A training dataset can never be comprehensive and can't teach everything that a model might encounter in the real world. Therefore a test dataset, containing *unseen* data points, is used to evaluate the model's accuracy.
- Then there's **validation data**. This is a dataset used for frequent evaluation during the training phase. Although the model sees this dataset occasionally, it doesn't *learn* from it. The validation set is also referred to as the development set or dev set. It helps protect models from overfitting and underfitting.
- Although validation data is separate from training data, data scientists might reserve a part of the training data for validation. But of course, this automatically means that the validation data was kept away during the training.
- Many use the terms "test data" and "validation data" interchangeably. The main difference between the two is that validation data is used to validate the model during

the training, while the testing set is used to test the model after the training is completed.

- The validation dataset gives the model the first taste of unseen data. However, not all data scientists perform an initial check using validation data. They might skip this part and go directly to testing data.



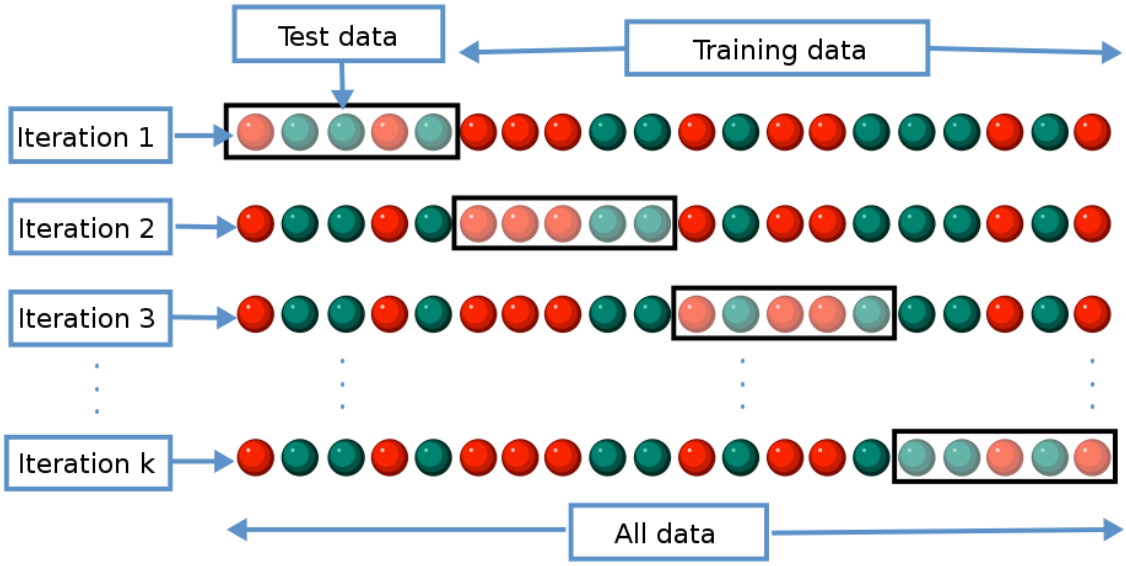


23. Explain with neat sketch K-fold cross-validation mode.

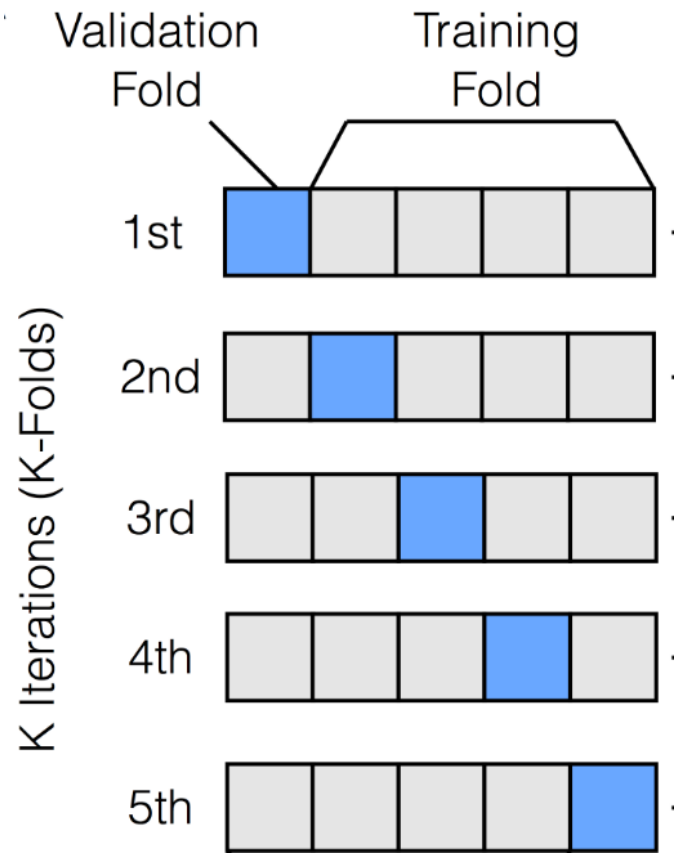
- The classifier model can be designed/trained and performance can be evaluated based on **K-fold cross-validation mode, training mode and test mode.**
- The main idea behind **K-Fold cross-validation** is that **each sample** in our dataset has the **opportunity of being tested**. It is a special case of cross-validation where we **iterate over a dataset set k times**. In each round, we **split the dataset into k parts**:

one part is used for validation, and the **remaining k-1 parts are merged into a training subset** for model evaluation

- **Computation time is reduced** as we repeated the process only 10 times when the value of k is 10. It has **Reduced bias.**
- **Every data points get to be tested exactly once** and is used in training k-1 times
- The **variance of the resulting estimate is reduced as k increases**



24. Explain with neat sketch 5-fold cross-validation mode.



25. What is hyper parameter tuning?

- Machine learning algorithms have hyperparameters that **allow you to tailor the behavior of the algorithm to your specific dataset.**
- Hyperparameters are different from parameters, which are the **internal coefficients or weights** for a model found by the learning algorithm. Unlike parameters, hyperparameters are specified by the practitioner when configuring the model.
- Typically, it is challenging to know what values to use for the hyperparameters of a given algorithm on a given dataset, therefore it is common to **use random or grid search strategies for different hyperparameter values**.
- The **more hyperparameters** of an algorithm that you need to tune, **the slower the tuning process.** Therefore, it is desirable **to select a minimum subset of model hyperparameters to search or tune.**

26. Explain hyper parameter tuning for simple decision tree.

Max_Depth: The maximum depth of the tree. If this is not specified in the Decision Tree, the nodes will be expanded until all leaf nodes are pure or until all leaf nodes contain less than min_samples_split.

- Default = None
- Input options → integer

Min_Samples_Split: The minimum samples required to split an internal node. If the amount of sample in an internal node is less than the min_samples_split, then that node will become a leaf node.

- Default = 2
- Input options → integer or float (if float, then min_samples_split is fraction)

Min_Samples_Leaf: The minimum samples required to be at a leaf node. Therefore, a split can only happen if it leaves at least the min_samples_leaf in both of the resulting nodes.

- Default = 1
- Input options → integer or float (if float, then min_samples_leaf is fraction)

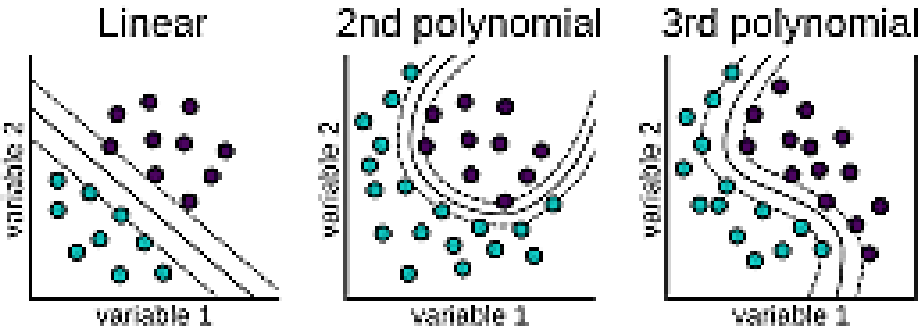
Max_Features: The number of features to consider when looking for the best split. For example, if there are 35 features in a dataframe and max_features is 9, only 9 of the 35 features will be used in the decision tree.

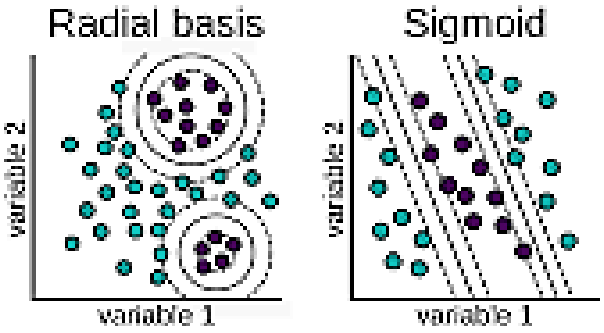
- Default = None
- Input options → integer, float (if float, then max_features is fraction) or {"auto", "sqrt", "log2"}
- "auto": max_features=sqrt(n_features)
- "sqrt": max_features = sqrt(n_features)

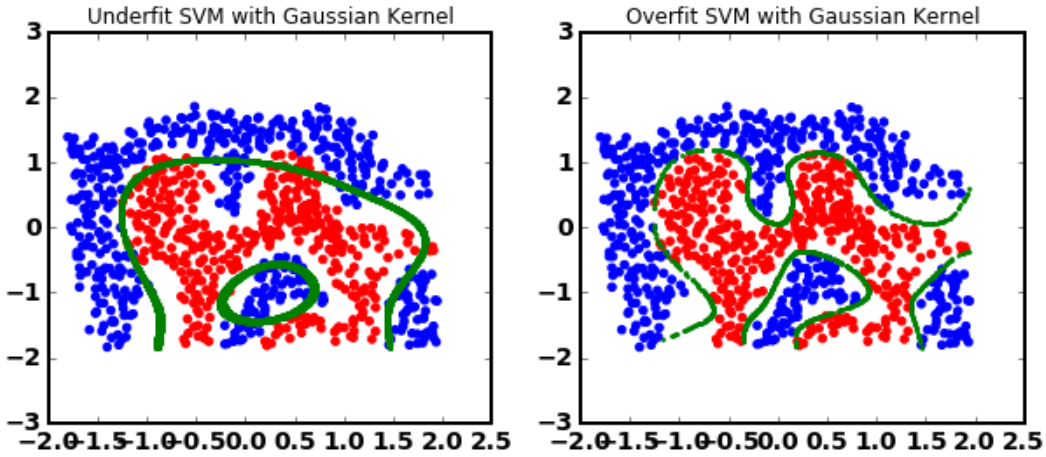
- "log2": max_features=log2(n_features)

27. Explain hyper parameter tuning for SVM.

- The **choice of kernel that will control the manner in which the input variables will be projected.** There are many to choose from, but **linear, polynomial, and RBF** are the most common, perhaps just linear and RBF in practice.
- **kernels in ['linear', 'poly', 'rbf', 'sigmoid']**
- If the polynomial kernel works out, then it is a good idea to dive into the degree hyperparameter.
- Another critical parameter is the **penalty (C)** that can take on a range of values and has a dramatic effect on the shape of the resulting regions for each class. A log scale might be a good starting point.
- **C in [100, 10, 1.0, 0.1, 0.001]**

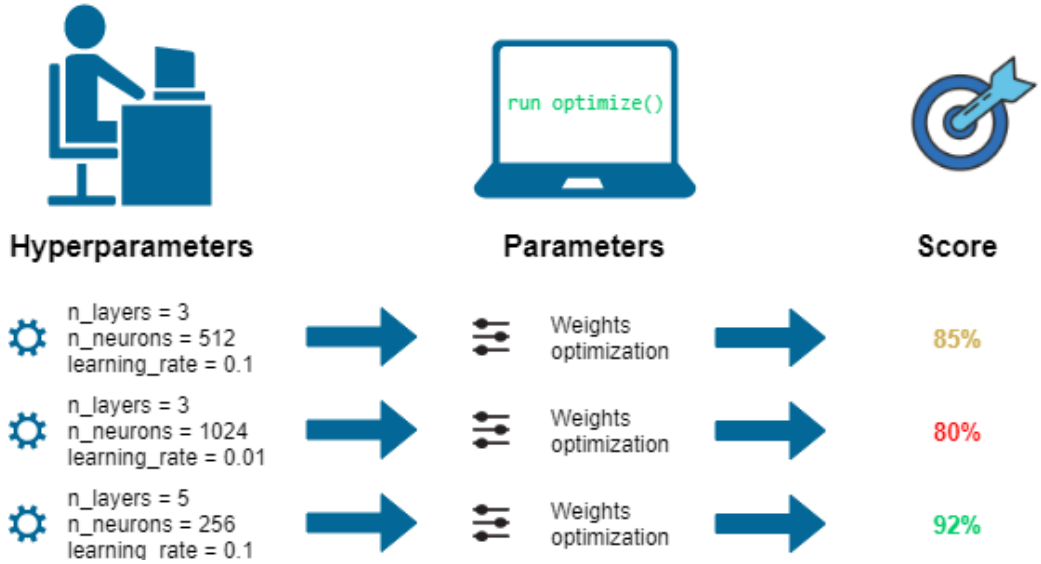






28. Explain hyper parameter tuning for ANN.

- **Number of neurons:** A weight is the amplification of input signals to a neuron and bias is an additive bias term to a neuron.
- **Activation function:** Defines how a neuron or group of neurons activate ("spiking") based on input connections and bias term(s).
- **Learning rate:** Step length for gradient descent update
- **Batch size:** Number of training examples in each gradient descent (gd) update.
- **Epochs:** The number of times all training examples have been passed through the network during training.
- **Loss function:** Loss function specifies how to calculate the error between prediction and label for a given training example. The error is backpropagated during training in order to update learnable parameters.
- **Number of layers:** Typically layers between input and output layer, which are called hidden layers



Theory/Mathematics based questions/ Problems/Numerical

29. Explain different performance evaluators used for interpretation/assessment of classification model. Explain 2 x 2 confusion matrix and explain its terminology. Explain Cohen's Kappa, F Score, ROC Curve. & Problems on calculating parameters required for model evaluation such True positive, True negative, False positive, False negative etc.

CLASSIFICATION MODEL:

- **Confusion matrix:** A Confusion matrix is an **N x N matrix** used for evaluating the performance of a classification model, where **N is the number of target classes**. The matrix compares the **actual target values with those predicted by the machine learning model**
- **What can we learn from this matrix?**
- There are **two possible predicted classes: "yes" and "no".** If we were predicting the presence of a disease, for example, **"yes" would mean they have the disease,** and **"no" would mean they don't have the disease.**
- The classifier made a total of **165 predictions** (e.g., **165 patients were being tested** for the presence of that disease).
- **Out of those 165 cases,** the classifier **predicted "yes" 110 times,** and **"no" 55 times.**
- **In reality, 105 patients in the sample have the disease,** and **60 patients do not.**

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

- **True positives (TP):** these are cases in which we predicted yes (they have the disease), and they do have the disease.
- **True negatives (tn):** we predicted no, and they don't have the disease.
- **False positives (fp):** we predicted yes, but they don't actually have the disease. (Also known as a "type I error.")
- **False negatives (fn):** we predicted no, but they actually do have the disease. (Also known as a "type II error.")

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

- **Accuracy:** Overall, how often is the classifier correct? (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong? (FP+FN)/total = (10+5)/165 = 0.09 which is equivalent to 1 minus Accuracy
- **True Positive Rate:** When it's actually yes, how often does it predict yes? TP/actual yes = 100/105 = 0.95 also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes? FP/actual no = 10/60 = 0.17
- **True Negative Rate:** When it's actually no, how often does it predict no? TN/actual no = 50/60 = 0.83 which is equal to 1 minus False Positive Rate
- **Precision:** When it predicts yes, how often is it correct? TP/predicted yes = 100/110 = 0.91
- **Cohen's Kappa:** This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate. (More details about Cohen's Kappa.)
- **F Score:** This is a weighted average of the true positive rate (recall) and precision. (More details about the F Score.)
- **ROC Curve:** This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class. (More details about ROC Curves.)

- **REGRESSION MODEL:**
- **Mean Absolute Error (MAE)**

The mean absolute error (MAE) is defined the MAE as

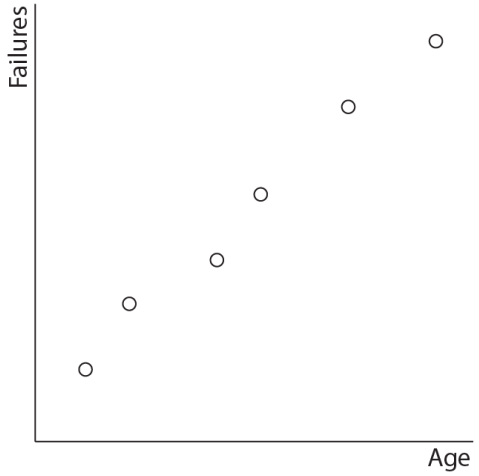
$$MAE = \frac{\Sigma|y - \hat{y}|}{N}$$

Where y is the actual value $\hat{y}$ is the predicted value and $\Sigma|y - \hat{y}|$ is the absolute value of the difference between the actual and predicted value.

N is the number of sample points.

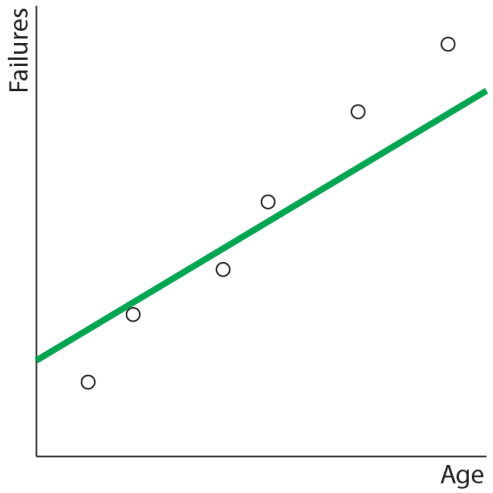
Let's dig into this a bit deeper to understand what this calculation represents.

Take a look at the following plot, which shows the number of failures for a piece of machinery against the age of the machine:



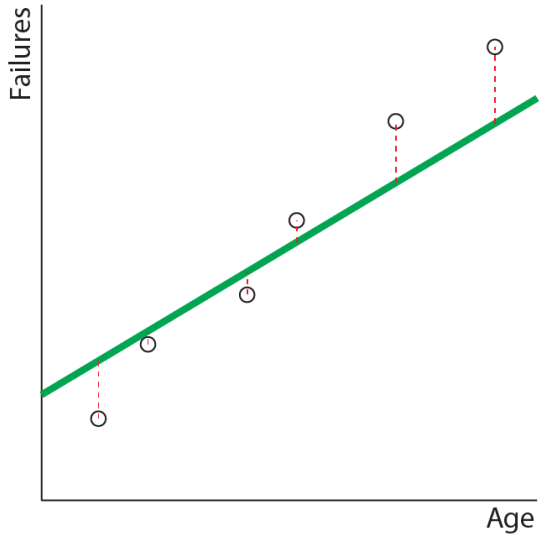
Age	Failures
10	15
20	30
40	40
50	55
70	75
90	90

In order to predict the number of failures from the age, we would want to fit a regression line such as this:



Age	Failures	Prediction
10	15	26
20	30	32
40	40	44
50	55	50
70	75	62
90	90	74

In order to understand how well this line represents the actual data, we need to measure *how good a fit it is*. We can do this by measuring the distance from the actual data points to the line:



Age	Failures	Prediction	Error
10	15	26	11
20	30	32	2
40	40	44	4
50	55	50	-5
70	75	62	-13
90	90	74	-16

You may recall that these distances are called residuals or errors. The mean size of these errors is the MAE. We can calculate it as follows:

Age	Failures	Prediction	Error	abs(Error)
10	15	26	11	11
20	30	32	2	2
40	40	44	4	4
50	55	50	-5	5
70	75	62	-13	13
90	90	74	-16	16

Mean abs(Error)	8.5

The mean of the absolute errors (MAE) is 8.5. Why do we take the absolute value? To **remove** the sign on the error value! If we don't, the positive and negative errors will tend to cancel each other out, giving a misleadingly small value for our evaluation metric. If mathematical symbols are not your strong point, you may not immediately see how this calculation relates to the formula at the start of this chapter:

$$MAE = \frac{\Sigma|y - \hat{y}|}{N}$$

So here is how the table and formula relate:

Age	y Failures	$\hat{y}$ Prediction	$y-\hat{y}$ Error	$\lvert y-\hat{y}\rvert$ abs(Error)
10	15	26	11	11
20	30	32	2	2
40	40	44	4	4
50	55	50	-5	5
70	75	62	-13	13
90	90	74	-16	16

Mean abs(Error)	$\frac{\Sigma\lvert y-\hat{y}\rvert}{N}$	8.5

Mean Absolute Error (MAE) tells us the average error in units of *y*, the predicted feature. A value of 0 indicates a perfect fit, i.e. all our predictions are spot on.

The MAE has a big advantage in that **the units of the MAE are the same as the units of *y*,** the feature we want to predict. In the example above, we have an MAE of 8.5, so it means that on average our predictions of the number of machine failures are incorrect by 8.5 machine failures. This makes MAE very intuitive and the results are easily conveyed to a non-machine learning expert!

- **Root Mean Square Error (RMSE)**

Another evaluation metric for regression is the **root mean square error (RMSE)**. Its calculation is very similar to MAE, but instead of taking the *absolute* value to get rid of the sign on the individual errors, we *square* the error (because the square of a negative number is positive). The formula for RMSE is:

$$RMSE = \sqrt{\frac{\Sigma(y-\hat{y})^2}{N}}$$

Here is the calculation for RMSE on our example scenario:

Age	y Failures	$\hat{y}$ Prediction	$y-\hat{y}$ Error	$(y-\hat{y})^2$ Error^2
10	15	26	11	121
20	30	32	2	4
40	40	44	4	16
50	55	50	-5	25
70	75	62	-13	169
90	90	74	-16	256

Mean of Error^2	$\frac{\Sigma(y-\hat{y})^2}{N}$	98.5
Square root of Mean of Error^2	$\sqrt{\frac{\Sigma(y-\hat{y})^2}{N}}$	**9.9**

As with MAE, we can think of **RMSE as being measured in the y units.** So the above error can be read as an error of 9.9 machine failures on average per observation.

- **MAE vs. RMSE**

Compared to MAE, RMSE gives a higher total error and the gap increases as the errors become larger. **It penalizes a few large errors more than a lot of small errors.** If you want your model to avoid large errors, use RMSE over MAE.

Root Mean Square Error (RMSE) indicates the average error in units of y, the predicted feature, but penalizes larger errors more severely than MAE. A value of 0 indicates a perfect fit. You should also be aware that as the sample size increases, the accumulation of slightly

higher RMSEs than MAEs means that the gap between these two measures also increases as the sample size increases.

- **R-Squared**

As stated above that an advantage of both MAE and RMSE is that they can be thought of as errors in the units of *y*, the predicted feature. This is helpful when relaying the results to non-data scientists.

We can say things like "our model can predict the reliability of our machinery to within 8.5 machine failures on average" or "our model can predict the selling price of a house to within £15k on average".

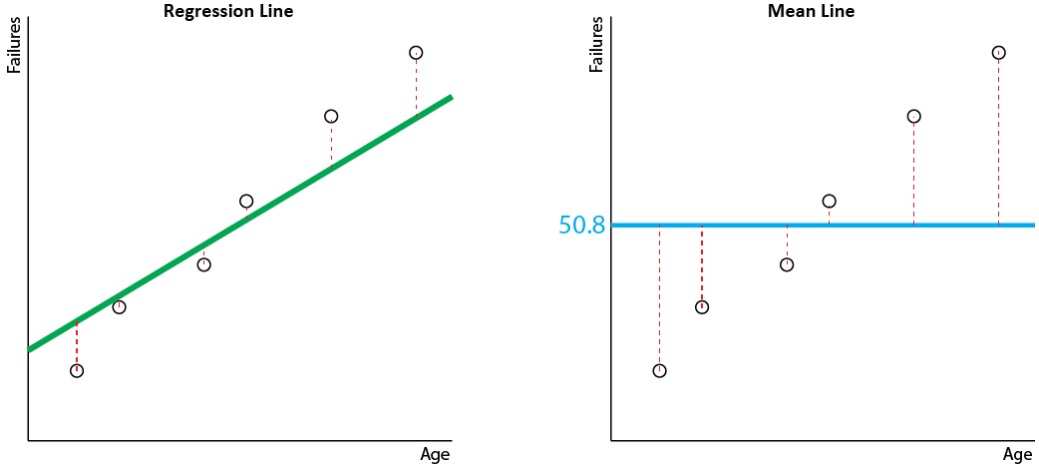
But take heed! This advantage can also be considered a disadvantage! It says nothing about whether an error of 8.5 machine failures or an error of £15k on a house price is *good or bad*. We can't compare *how good* different models are for different scenarios. This is where R-squared or R^2 comes in. Here is the formula for R^2.

$$R^2 = \frac{\Sigma(y - \bar{y})^2 - \Sigma(y - \hat{y})^2}{\Sigma(y - \bar{y})^2}$$

R^2 computes **how much better the regression line fits the data than the mean line**. Another way to look at this formula is to compare the *variance* around the mean line to the variation around the regression line:

$$R^2 = \frac{var(mean) - var(line)}{var(mean)}$$

Take our example above, predicting the number of machine failures. We can examine the errors for our regression line as we did before. We can also compute a mean line (by taking the mean *y (*value) and examine the errors against this mean line. That is to say, we can see the errors we would get if our model just predicted the mean number of failures (50.8) for *every* age input. Here are the regression and mean lines, and their respective errors:



You can see that the regression line fits the data better than the mean line, which is what we expected (the mean line is a pretty simplistic model, after all). But can you say *how* much better it is? That's exactly what R^2 does! Here is the calculation.

			Regression Line	Mean Line	Regression Line	Mean Line
	y	$\hat{y}$	$y-\hat{y}$	$y-\bar{y}$	$(y-\hat{y})^2$	$(y-\bar{y})^2$
Age	Failures	Prediction	Error	Error	Error^2	Error^2
10	15	26	11	-35.8	121	1281.6
20	30	32	2	-20.8	4	432.6
40	40	44	4	-10.8	16	116.6
50	55	50	-5	4.2	25	17.6
70	75	62	-13	24.2	169	585.6
90	90	74	-16	39.2	256	1536.6
Mean of Error^2					$\frac{\Sigma(y-\hat{y})^2}{N}$ 98.5	$\frac{\Sigma(y-\bar{y})^2}{N}$ 661.8
R^2					$\frac{\Sigma(y-\bar{y})^2-\Sigma(y-\hat{y})^2}{\Sigma(y-\bar{y})^2}$	**0.85**

Notice something? Most of this is the same as the calculation of RMSE. The additional parts to the calculation are the column on the far right (in blue) and the final calculation row, computing R^2. So we have an R-squared of 0.85. Without even worrying about the units of y we can say this is a decent model. Why? Because the model explains 85% of the variation in the data. That's exactly what an R-squared of 0.85 tells us!

R-squared (R^2) tells us the degree to which the model explains the variance in the data. In other words, how much better it is than just predicting the mean. Here's another example. What if our data points and regression line looked like this?

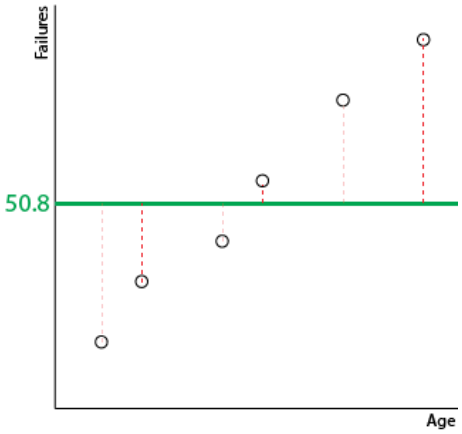
The variance around the regression line is 0. In other words, var(line) is 0. There are no errors. Now, remember that the formula for R-squared is:

$$R^2 = \frac{var(mean) - var(line)}{var(mean)}$$

So, with var(line) = 0 the above calculation for R-squared is

$$R^2 = \frac{var(mean) - 0}{var(mean)} = \frac{var(mean)}{var(mean)} = 1$$

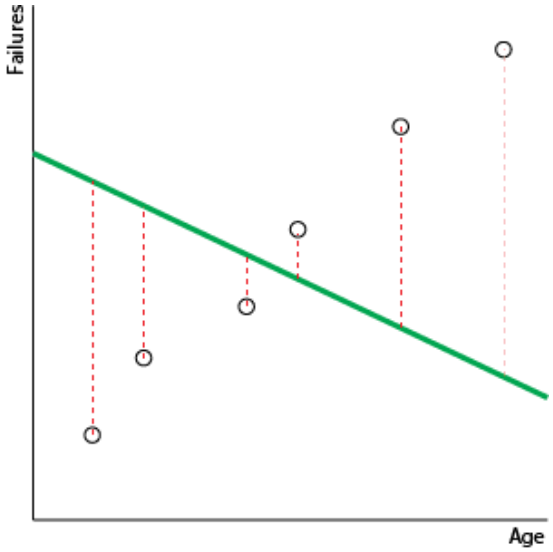
So, if we have a perfect regression line, with no errors, we get an R-squared of 1. Let's look at another example. What if our data points and regression line looked like this, with the regression line *equal* to the mean line?



In this case, var(line) and var(mean) are the same. So the above calculation will yield an R-squared of 0:

$$R^2 = \frac{var(mean) - var(mean)}{var(mean)} = \frac{0}{var(mean)} = 0$$

So, if our regression line is only as good as the mean line, we get an R-squared of 0. What if our regression line was *really* bad, worse than the mean line?



It's unlikely to get this bad! But if it does, var(mean)-var(line) will be negative, so R-squared will be negative. An R-squared of 1 indicates a perfect fit. An R-squared of 0 indicates a model no better or worse than the mean. An R-squared of less than 0 indicates a model worse than just predicting the mean.

30. Identify methodology to attempt following problems and enlist general steps involved in it.

Methodology used for building regression models:

- To estimate remaining useful life of bearings /gears /cutting tool
- To guess dryness fraction in the steam generated by boiler
- To forecast material property
- To estimate engine emissions for remaining useful life
- To predict refrigerant two-phase pressure drop inside brazed plate heat exchangers
- To project enhanced state of charge of Lithium-ion Batteries used in EV

Methodology used for building clustering models:

- Discover product segments for marketing purposes
- Identify high temperature zones in process equipment.
- To recognize microstructure of material

Methodology used for building classification models:

- To diagnose condition of rotating machine element as healthy or faulty
- To identify quality of steam generated by boiler as wet, dry saturated or superheated
- To provide the correct quality of air-fuel mixture during the conditions such as starting or idling or cruising
- To estimate wear state of Rolling Element Bearings

Feel free to contact me on +91-8329347107 calling / +91-9922369797 whatsapp, email ID: adp.mech@coep.ac.in and abhipatange93@gmail.com

****

View publication stats