Machine learning

Revision Notes M1-M8

1. What are the main steps in the machine learning workflow?

Define the problem, gather data, preprocess data, train the model, evaluate the model, tune the model, and deploy it.

2. Why is data critical in a machine learning workflow?

Data is the foundation for model training. Quality, diversity, and quantity of data influence model performance and generalizability.

3. What is data preprocessing, and why is it important?

Data preprocessing involves cleaning, transforming, and encoding data to make it suitable for machine learning. It helps improve model accuracy and efficiency.

4. Explain data wrangling. How does it differ from preprocessing?

Data wrangling focuses on converting raw data into a usable format (e.g., removing duplicates, handling missing values). Preprocessing involves additional steps like normalization or encoding.

5. What is data skewness, and how can sampling help remove it?

Data skewness occurs when data distribution is imbalanced. Sampling (oversampling, undersampling, SMOTE) adjusts the distribution for balanced representation.

6. How is the direct solution method different from gradient descent?

The direct method computes exact parameter values (e.g., Normal Equation). Gradient descent iteratively adjusts parameters to minimize the cost function.

7. Differentiate between batch, stochastic, and mini-batch gradient descent.

Batch uses all data per step; stochastic uses one data point per step; mini-batch uses subsets of data.

8. What is the purpose of bias-variance decomposition?

It analyzes model performance by breaking error into bias (error from incorrect assumptions) and variance (error from sensitivity to data changes).

9. What are discriminant functions used for in classification?

They separate data points into classes by defining decision boundaries.

10. Explain the key idea behind logistic regression.

Logistic regression models the probability of a class using the sigmoid function and optimizes it via maximum likelihood estimation.

11. What is entropy, and how is it used in decision tree construction?

Entropy measures impurity or disorder. Decision trees use entropy to choose splits that maximize information gain.

12. How can overfitting in decision trees be avoided?

Overfitting can be avoided using pruning, limiting tree depth, or using a validation set.

13. Describe the k-Nearest Neighbor algorithm.

k-NN classifies a data point based on the majority label of its kkk nearest neighbors in feature space.

14. What is locally weighted regression (LWR)?

LWR assigns weights to data points based on proximity to the query point, enabling localized model fitting.

15 . Describe Support Vector Machine (SVM)

A **Support Vector Machine (SVM)** is a supervised machine learning algorithm used for **classification** and **regression** tasks. It is particularly effective for high-dimensional datasets and problems where a clear margin of separation exists between classes.

16**. How does the kernel trick enable SVMs to handle non-linearly separable data?**

The kernel trick maps data to a higher-dimensional space where a linear boundary can separate classes.

17. Give an example of SVM application in unstructured data.

- **Answer**: SVMs can classify text documents by mapping word embeddings into feature space.

18. Differentiate between MLE and MAP hypotheses.

MLE [Maximun Likelihood Estimation] maximizes likelihood based only on data;

MAP [Maximum A Posteriori Estimation] incorporates a prior distribution, combining data and prior knowledge.

19**. What is the Naïve Bayes assumption?**

Naïve Bayes assumes that features are conditionally independent given the class label.

20. **A data scientist collects a dataset with missing values, outliers, and categorical variables. The dataset also has a target variable for prediction. What steps would the scientist follow to preprocess the data and build a machine learning model?**

1. **Handle Missing Values**: Use imputation (mean, median, or mode) or drop missing rows/columns.
2. **Remove Outliers**: Identify using techniques like the IQR rule or Z-scores, and handle them.
3. **Encode Categorical Variables**: Use one-hot encoding or label encoding for categorical data.
4. **Feature Scaling**: Normalize or standardize numeric features.
5. **Split the Dataset**: Create training, validation, and testing sets.
6. **Train and Evaluate the Model**: Use a suitable algorithm, fit the training data, and evaluate it using validation and testing sets.

21. In a binary classification problem, the dataset has 95% of examples in class A and 5% in class B. The model is biased towards predicting class A. **What sampling methods can be used to address this issue?**

1. **Oversampling**: Duplicate or synthesize examples of the minority class (e.g., SMOTE).
2. **Undersampling**: Randomly remove examples from the majority class to balance the dataset.
3. **Combine Both**: Use hybrid methods to oversample the minority class and undersample the majority class.

22. Given the dataset: x=[1,2,3] y=[2,4,6]

Calculate the slope (m) and intercept (b) using the direct method for linear regression.

1. Compute means:

$$\bar{x} = \frac{1+2+3}{3} = 2, \quad \bar{y} = \frac{2+4+6}{3} = 4$$

2. Calculate slope:

$$m = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{(1-2)(2-4) + (2-2)(4-4) + (3-2)(6-4)}{(1-2)^2 + (2-2)^2 + (3-2)^2} = 2$$

3. Calculate intercept:

$$b = \bar{y} - m\bar{x} = 4 - 2(2) = 0$$

Regression equation: $y = 2x + 0$

23. In a gradient descent process, the loss function is $L=(w-3)^2$. If $w_0=0$ and the learning rate $\eta=0.1$, compute the first two updates for w

1. Compute gradient $\frac{\partial L}{\partial w} = 2(w-3)$:
   At $w = 0$, $\frac{\partial L}{\partial w} = 2(0-3) = -6$.
2. Update w:
   $w_1 = w_0 - \eta \cdot \frac{\partial L}{\partial w} = 0 - 0.1 \cdot (-6) = 0.6$.
3. Repeat:
   At $w = 0.6$, $\frac{\partial L}{\partial w} = 2(0.6-3) = -4.8$.
   $w_2 = w_1 - \eta \cdot \frac{\partial L}{\partial w} = 0.6 - 0.1 \cdot (-4.8) = 1.08$.

First two updates: $w_1 = 0.6, w_2 = 1.08$.

24. The logistic regression model predicts a probability of 0.8 for a class. The actual class is 1. Calculate the log-loss.

$$\text{Log-Loss} = -[y \cdot \log(p) + (1-y) \cdot \log(1-p)]$$

Substitute $y = 1$ and $p = 0.8$:

$$\text{Log-Loss} = -[1 \cdot \log(0.8) + 0 \cdot \log(1-0.8)] = -\log(0.8) \approx 0.22$$

25. A decision tree splits a dataset into two subsets with entropy values 0.4 and 0.6. Subset sizes are 30 and 70, respectively. Calculate the weighted entropy.

$$\text{Weighted Entropy} = \frac{30}{100} \cdot 0.4 + \frac{70}{100} \cdot 0.6 = 0.12 + 0.42 = 0.54$$

26. What kernel function would you use to classify non-linearly separable data where the decision boundary is circular?

The **Radial Basis Function (RBF)** kernel is the best choice for non-linearly separable data, especially when the decision boundary is circular or involves complex patterns.

27. Given a prior P(H)=0.3 likelihood P(D|H)=0.8, and evidence P(D)=0.5, calculate the posterior P(H|D).

Using Bayes' Rule:

$$P(H|D) = \frac{P(D|H) \cdot P(H)}{P(D)} = \frac{0.8 \cdot 0.3}{0.5} = 0.48$$

28. Sam is building a model to predict house prices. The dataset contains 80% numerical data and 20% categorical data (e.g., "City" and "Property Type"). What preprocessing steps would Sam apply to prepare the data for a linear regression model?

Given the confusion matrix:

[TP=50 FN=10
FP=15 TN=25]

Calculate the following metrics:

1. Accuracy
2. Precision
3. Recall
4. F1-score

1. **Accuracy**:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{50 + 25}{50 + 25 + 15 + 10} = \frac{75}{100} = 0.75$$

2. **Precision**:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{50}{50 + 15} = \frac{50}{65} \approx 0.769$$

3. **Recall**:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{50}{50 + 10} = \frac{50}{60} \approx 0.833$$

4. **F1-score**:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \cdot \frac{0.769 \cdot 0.833}{0.769 + 0.833} \approx 0.8$$

29. For a dataset for churn prediction, where only 5% of customers have churned. What specific method would you use to balance this dataset? Why?

Use **oversampling (e.g., SMOTE)** or **under sampling**:

- **Oversampling** generates synthetic examples for the minority class to balance the dataset.
- **Undersampling** reduces the number of majority class samples.

For churn prediction, **SMOTE** is preferred because it generates realistic synthetic samples instead of simply duplicating existing ones. SMOTE (Synthetic Minority Over-sampling Technique) is a technique used to address class imbalance in machine learning tasks. It is particularly useful when you have a dataset where one class (typically the minority class) is underrepresented compared to the other class (majority class). This imbalance can lead to poor performance of machine learning models, as the model tends to be biased towards predicting the majority class.

30. The dataset has outliers in one feature, X, with values: X=[1,2,3,4,100] Compute the IQR (Interquartile Range). Find outliers

1. Compute $Q1$ and $Q3$:
   - Sort: $[1, 2, 3, 4, 100]$.
   - Median = $Q2 = 3$.
   - $Q1 = 2$ (lower quartile), $Q3 = 4$ (upper quartile).
2. Compute IQR:

$$\text{IQR} = Q3 - Q1 = 4 - 2 = 2$$

3. Outlier bounds:

$$\text{Lower bound} = Q1 - 1.5 \times \text{IQR} = 2 - 3 = -1$$

$$\text{Upper bound} = Q3 + 1.5 \times \text{IQR} = 4 + 3 = 7$$

Since $100 > 7$, it is an outlier.

31.You trained a linear regression model to predict sales based on advertising budgets. The coefficient for the "TV Advertising" feature is 3.5 What does this coefficient represent?

The coefficient 3.5 means that for every additional dollar spent on TV advertising, the predicted sales increase by 3.5 units, assuming other factors remain constant.

32. Given the cost function for linear regression: Derive Gradient Descent

Given:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

Gradient:

$$\frac{\partial J}{\partial \theta} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) x_i$$

Steps:

1. Compute the predicted value $h_\theta(x_i) = \theta_0 + \theta_1 x_i$.
2. Substitute values into the gradient formula.
3. Update θ_0 and θ_1 using:

$$\theta_j = \theta_j - \eta \cdot \frac{\partial J}{\partial \theta_j}$$

where η is the learning rate.

33. A dataset has 1 million samples. **Which type of gradient descent (batch, stochastic, or mini-batch) would you use and why?**

$$J(w) = (w-5)^2$$

1. **Compute Gradient:**

$$\frac{\partial J}{\partial w} = 2(w-5)$$

2. **Gradient at** $w_0 = 0$:

$$\frac{\partial J}{\partial w} = 2(0-5) = -10$$

3. **Update Rule:**

$$w_1 = w_0 - \eta \cdot \frac{\partial J}{\partial w}$$

$$w_1 = 0 - 0.1 \cdot (-10) = 1$$

34 You are constructing a decision tree for a dataset with the target variable "Will Buy" (Yes/No). You find that the "Income" attribute splits the data into subsets with the following proportions:

- Subset 1: 50% Yes, 50% No.
- Subset 2: 80% Yes, 20% No.

Calculate the weighted Gini index for the split if Subset 1 contains 40 samples and Subset 2 contains 60 samples.

Step 1: Gini index formula

For a subset, the Gini index is:

$$G = 1 - \sum_i (P_i)^2$$

where P_i is the proportion of each class in the subset.

Step 2: Calculate the Gini index for each subset

Subset 1:

- Proportions:
  - $P_{\text{Yes}} = 0.5$
  - $P_{\text{No}} = 0.5$

$$G_1 = 1 - (0.5^2 + 0.5^2) = 1 - (0.25 + 0.25) = 1 - 0.5 = 0.5$$

Subset 2:

- Proportions:
  - $P_{\text{Yes}} = 0.8$
  - $P_{\text{No}} = 0.2$

$$G_2 = 1 - (0.8^2 + 0.2^2) = 1 - (0.64 + 0.04) = 1 - 0.68 = 0.32$$

Step 3: Weighted Gini index

The weighted Gini index is calculated as:

$$G_{\text{weighted}} = \frac{\text{Size of Subset 1}}{\text{Total Size}} \cdot G_1 + \frac{\text{Size of Subset 2}}{\text{Total Size}} \cdot G_2$$

- Size of Subset 1 = 40
- Size of Subset 2 = 60
- Total size = $40 + 60 = 100$

$$G_{\text{weighted}} = \frac{40}{100} \cdot 0.5 + \frac{60}{100} \cdot 0.32$$

$$G_{\text{weighted}} = 0.4 \cdot 0.5 + 0.6 \cdot 0.32$$

$$G_{\text{weighted}} = 0.2 + 0.192 = 0.392$$

35 A decision tree splits a dataset into two subsets with entropy values 0.4 and 0.6. Subset sizes are 30 and 70, respectively. Calculate the weighted entropy.

$$\text{Weighted Entropy} = \frac{30}{100} \cdot 0.4 + \frac{70}{100} \cdot 0.6 = 0.12 + 0.42 = 0.54$$

36. A logistic regression model predicts the probability of an event. If the model outputs P=0.65 and the threshold is 0.7,What class will the model assign to this data point? Why?

The model assigns a class based on whether the predicted probability P exceeds the threshold.

- If $P \geq 0.7$, the model typically assigns the positive class (class 1).
- If $P < 0.7$, the model typically assigns the negative class (class 0).

Since P=0.65 is less than the threshold of 0.7, the model will assign the **negative class (class 0)** to this data point.

37. **Formula**

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Given $z = -2$, calculate $\sigma(z)$.

Answer

Substitute $z = -2$ into the equation:

$$\sigma(-2) = \frac{1}{1 + e^{-(-2)}} = \frac{1}{1 + e^{2}}$$

Now, calculate e^2:

$$e^2 \approx 7.389$$

So the equation becomes:

$$\sigma(-2) = \frac{1}{1 + 7.389} = \frac{1}{8.389}$$

Finally, calculate the result:

$$\sigma(-2) \approx 0.1192$$

38. You are using an SVM to classify emails as spam or not spam. Which kernel function would you use if the email features are high-dimensional (e.g., bag-of-words)? Why?

When dealing with high-dimensional features, such as those in a bag-of-words model for classifying emails as spam or not spam, the most commonly used kernel function in Support Vector Machines (SVMs) is the Radial Basis Function (RBF) kernel.

Reason for RBF Kernel

1. **Non-linearity**:
    - The bag-of-words model typically results in a **sparse, high-dimensional feature space**, where the relationships between features are not necessarily linear. The RBF kernel is well-suited to handle these situations because it can map the data

into a higher-dimensional space, where a linear separation may be easier to achieve. Essentially, the RBF kernel can capture complex, non-linear decision boundaries.

2. **Handling High Dimensions**:
   - SVMs with linear kernels perform well when the data is linearly separable or nearly linearly separable. However, in high-dimensional spaces like those created by a bag-of-words model, linear decision boundaries may not be effective. The RBF kernel implicitly maps data to a higher-dimensional feature space where non-linear decision boundaries can be found.
3. **Flexibility**:
   - The RBF kernel is flexible because it depends on a **hyperparameter** (typically denoted as γ) that controls the "spread" of the kernel. This allows the model to adapt to varying complexities in the data and better capture the structure of high-dimensional spaces.

RBF Kernel Formula:

The RBF kernel function between two data points x and 'x′ is given by:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

Where:

- $\|x - x'\|^2$ is the squared Euclidean distance between the data points.
- σ is a parameter that controls the width of the kernel and helps regulate the decision boundary's flexibility.

39.

For an SVM with a linear kernel, the decision boundary is:

$$f(x) = w^T x + b$$

Given $w = [2, -1]$, $b = -4$, and $x = [3, 2]$, calculate $f(x)$ and determine the class label.

Machine learning

Revision Notes M1-M8

The decision function for an SVM with a linear kernel is given by:

$$f(x) = \mathbf{w}^T\mathbf{x} + b$$

Where:

- $\mathbf{w} = [2, -1]$ is the weight vector,
- $\mathbf{x} = [3, 2]$ is the input feature vector,
- $b = -4$ is the bias term.

We are asked to compute $f(x)$ and determine the class label.

Step 1: Compute $f(x)$

First, compute the dot product $\mathbf{w}^T\mathbf{x}$:

$$\mathbf{w}^T\mathbf{x} = (2)(3) + (-1)(2) = 6 - 2 = 4$$

Now, add the bias term b:

$$f(x) = 4 + (-4) = 0$$

Step 2: Determine the Class Label

The class label in SVMs is typically determined as follows:

- If $f(x) > 0$, the data point belongs to class +1.
- If $f(x) < 0$, the data point belongs to class -1.
- If $f(x) = 0$, the data point lies on the decision boundary.

In this case, $f(x) = 0$, which means the data point lies **exactly on the decision boundary**. This usually indicates an undecidable case or the need for further clarification, but in a practical SVM model, the data point is often considered as part of the class that the model was trained to classify as "positive" or "negative", depending on how the model was set up.

Final Answer:

- $f(x) = 0$.
- The data point lies on the decision boundary.

40. A doctor diagnoses a rare disease that occurs in 1 out of 1,000 people. The test has:

- **Sensitivity (True Positive Rate) = 99%.**
- **Specificity (True Negative Rate) = 95%.**

If a patient tests positive, calculate the posterior probability of the disease using Bayes' theorem.

Machine learning

Revision Notes M1-M8

To calculate the **posterior probability** of the disease given a positive test result, we use **Bayes' theorem**:

$$P(\text{Disease}|\text{Positive Test}) = \frac{P(\text{Positive Test}|\text{Disease})P(\text{Disease})}{P(\text{Positive Test})}$$

Where:

- $P(\text{Disease}|\text{Positive Test})$ is the **posterior probability** (the probability of having the disease given a positive test result),
- $P(\text{Positive Test}|\text{Disease})$ is the **sensitivity** (true positive rate),
- $P(\text{Disease})$ is the **prior probability** of the disease,
- $P(\text{Positive Test})$ is the **total probability of a positive test** (whether the person has the disease or not).

Given:

- Sensitivity (True Positive Rate) $= 99\% = 0.99$,
- Specificity (True Negative Rate) $= 95\% = 0.95$,
- Prevalence (Prior Probability) $P(\text{Disease}) = \frac{1}{1{,}000} = 0.001$,
- $P(\text{No Disease}) = 1 - P(\text{Disease}) = 0.999$.

Step 1: Calculate $P(\text{Positive Test})$

The total probability of a positive test, $P(\text{Positive Test})$, can be found using the law of total probability:

$$P(\text{Positive Test}) = P(\text{Positive Test}|\text{Disease})P(\text{Disease}) + P(\text{Positive Test}|\text{No Disease})P(\text{No Disease})$$

- $P(\text{Positive Test}|\text{Disease}) = 0.99$ (sensitivity),
- $P(\text{Positive Test}|\text{No Disease}) = 1 - \text{Specificity} = 1 - 0.95 = 0.05$ (false positive rate).

Step 1: Calculate $P(\text{Positive Test})$

The total probability of a positive test, $P(\text{Positive Test})$, can be found using the law of total probability:

$$P(\text{Positive Test}) = P(\text{Positive Test}|\text{Disease})P(\text{Disease}) + P(\text{Positive Test}|\text{No Disease})P(\text{No Disease})$$

- $P(\text{Positive Test}|\text{Disease}) = 0.99$ (sensitivity),
- $P(\text{Positive Test}|\text{No Disease}) = 1 - \text{Specificity} = 1 - 0.95 = 0.05$ (false positive rate).

So,

$$P(\text{Positive Test}) = (0.99)(0.001) + (0.05)(0.999)$$

$$P(\text{Positive Test}) = 0.00099 + 0.04995 = 0.05094$$

Step 2: Apply Bayes' Theorem

Now we can apply Bayes' theorem to find the posterior probability:

$$P(\text{Disease}|\text{Positive Test}) = \frac{(0.99)(0.001)}{0.05094}$$

$$P(\text{Disease}|\text{Positive Test}) = \frac{0.00099}{0.05094} \approx 0.0194$$

Approximately 1.94%

41. Consider the following hypothetical data concerning student characteristics and whether or not each student should be hired. Use a Naive Bayes classifier to determine whether or not someone with excellent attendance, poor GPA, and lots of effort should be hired.

Name	GPA	Effort	Hirable
Sarah	poor	lots	Yes
Dana	average	some	No
Alex	average	some	No
Annie	average	lots	Yes
Emily	excellent	lots	Yes
Pete	excellent	lots	No
John	excellent	lots	No
Kathy	poor	some	No

Calculate the Prior Probabilities

We need the prior probability for each outcome:

P(Hirable = Yes): Number of "Yes" outcomes / Total outcomes = 3/8

P(Hirable = No): Number of "No" outcomes / Total outcomes = 5/8

Calculate the probabilities for each feature (GPA, Effort) given the outcome "Hirable" and "Not Hirable".

- **For Hirable = Yes:**

  P(GPA = excellent | Hirable = Yes) = 1/3 (1 excellent GPA out of 3 hires)

  P(GPA = poor | Hirable = Yes) = 1/3 (1 poor GPA out of 3 hires)

  P(GPA = average | Hirable = Yes) = 1/3 (1 average GPA out of 3 hires)

  P(Effort = lots | Hirable = Yes) = 2/3 (2 lots of effort out of 3 hires)

  P(Effort = some | Hirable = Yes) = 1/3 (1 some effort out of 3 hires)

- **For Hirable = No:**

  P(GPA = excellent | Hirable = No) = 2/5 (2 excellent GPA out of 5 non-hires)

  P(GPA = poor | Hirable = No) = 1/5 (1 poor GPA out of 5 non-hires)

  P(GPA = average | Hirable = No) = 2/5 (2 average GPA out of 5 non-hires)

  P(Effort = lots | Hirable = No) = 3/5 (3 lots of effort out of 5 non-hires)

  P(Effort = some | Hirable = No) = 2/5 (2 some effort out of 5 non-hires)

  Apply Bayes' Theorem to calculate the probability of being hired given the student's characteristics:

  $$P(Hirable=Yes|GPA=poor,Effort=lots) \propto P(Hirable=Yes) \cdot P(GPA=poor|Hirable=Yes) \cdot P(Effort=lots|Hirable=Yes)$$

P(Hirable=No|GPA=poor,Effort=lots)∝P(Hirable=No)·P(GPA=poor|Hirable=No)·P(Effort=lots|Hirable=No)

Compute the Probabilities

For **Hirable = Yes**:

$$P(Hirable = Yes|GPA = poor, Effort = lots) \propto \frac{3}{8} \cdot \frac{1}{3} \cdot \frac{2}{3} = \frac{6}{72} = \frac{1}{12}$$

For **Hirable = No**:

$$P(Hirable = No|GPA = poor, Effort = lots) \propto \frac{5}{8} \cdot \frac{1}{5} \cdot \frac{3}{5} = \frac{15}{200} = \frac{3}{40}$$

Compare the probabilities

- **P(Hirable = Yes | GPA = poor, Effort = lots) = 1/12**
- **P(Hirable = No | GPA = poor, Effort = lots) = 3/40**

Since **1/12** is greater than **3/40**, we predict that the student should be **hired**.

Conclusion:

Based on the Naive Bayes classifier, the student with poor GPA, lots of effort, and excellent attendance is predicted to be hired.

42. Differentiate between classification and clustering algorithms with the help of suitable examples.

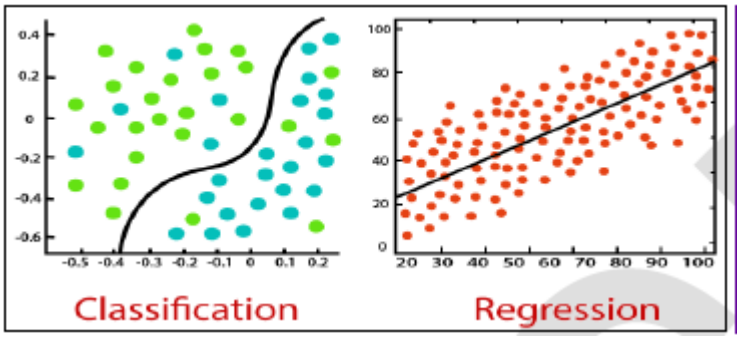
Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters. In Classification, a computer program is trained on the training dataset and based on that training; it categorizes the data into different classes. The task of the classification algorithm is to find the mapping function to map the input(x) to the discrete output(y).

Example: The best example to understand the Classification problem is Email Spam Detection. The model is trained on the basis of millions of emails on different parameters,

and whenever it receives a new email, it identifies whether the email is spam or not. If the email is spam, then it is moved to the Spam folder.

Regression is a process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables such as prediction of Market Trends, prediction of House prices, etc. The task of the Regression algorithm is to find the mapping function to map the input variable(x) to the continuous output variable(y).

Example: Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.



43. Calculate Accuracy, Precision, Recall and F1 Score for the following Confusion Matrix on Heart Attack Risk. Also suggest which metric would not be a good evaluation parameter here and why?

The Confusion Matrix	Reality: 1	Reality: 0
Prediction: 1	50	20
Prediction: 0	10	20

The Confusion Matrix	Reality: 1	Reality: 0	
Prediction: 1	50	20	70
Prediction: 0	10	20	30
	60	40	100

Here **TOTAL no of tests conducted** are sum of all cases recorded in matrix.

Here it is 50 + 20 + 10 + 20 = 100 tests conducted.

Total correct results = TP + TN = 50 + 20 = 70

Total wrong results = FP + FN = 20 + 10 = 30

Accuracy = (TP + TN) / (TP + TN + FP + FN)

= (50 + 20) / (50 + 20 + 20 + 10)

= 70 / 100

= 0.7

Precision = TP / (TP + FP)

= 50 / (50 + 20)

= 50 / 70

= 0.714

Recall = TP / (TP + FN)

= 50 / (50 + 10)

= 50/60

= 0.833

F1 Score = 2 * Precision * Recall / (Precision + Recall)

= 2 * 0.714 * 0.833/(0.714 + 0.833)

= 2 * 0.595/1.547

= 0.769

Therefore,

Accuracy= 0.7, Precision=0.714, Recall=0.833 F1 Score = 0.769

Here within the test there is a tradeoff. But Recall is not a good Evaluation metric. Recall metric needs to improve more.

Because, False Positive (impacts Precision): A person is predicted as high risk but does not have heart attack.

False Negative (impacts Recall): A person is predicted as low risk but has heart attack.

Therefore, False Negatives miss actual heart patients, hence recall metric need more improvement.

False Negatives are more dangerous than False Positives.