6. What is Statistical Learning? What is the objective of statistical learning?

Statistical learning:

- Statistical Learning theory is a framework for machine learning drawing from the fields of statistics and functional analysis.
- When the goal is to interpret the model and quantify the uncertainty in the data, this analysis is usually referred to as statistical learning.
- It focuses on developing methods and algorithms for making predictions or decisions based on data.
- Statistical learning theory deals with the statistical inference problem at finding predictive function based on data.
- Statistical theory has lead to successful applications in fields such as computer vision,speech recognition and Bioinformatics.
- From perspective of statistical learning, Supervised learning is best understood(from training set of data). we get to predict or estimate an outcome based on previously present output
- with unsupervised statistical learning, we find various patterns present within the data by clustering them into similar groups.
- Statistical Learning helps us understand why a system behaves the way it does.
- It reduces ambiguity and produces results that matter in the real world.
- Statistical Learning provides accurate results that can find medical, business, banking, and government applications.

Objectives of Statistical Learning:

- The primary objective of statistical learning is to understand the underlying patterns and relationships in data, allowing for accurate prediction, classification, or decision-making in new, unseen instances, minimizing the loss or empherical risk.
- Gaining knowledge, making predictions, making decisions or constructing models from a set of data.

Some of the objectives are:

1. **Prediction:**
   - **Objective:** Minimize the prediction error or loss function, ensuring that the model generalizes well to new data.
2. **Inference:**
   - **Objective:** Understand underlying data patterns and relationships between variables to gain insights into the factors that influence the outcomes.
3. **Decision Making:**
   - **Objective:** Develop models that aid decision-making processes, such as in business, healthcare, finance, and other fields.
4. **Pattern Recognition:**
   - **Objective**: Identify and leverage patterns, trends, and structures in the data.
5. **Classification and Regression:**
   - **Objective:** Categorize data (classification) or predict continuous variables (regression).
6. **Feature Selection and Dimensionality Reduction:**
   - **Objective:** Improve model efficiency by selecting relevant features and reducing data dimensionality.

7. **Model Interpretability:**
    - **Objective:** Enhance understanding by creating interpretable models for stakeholders.
8. **Scalability and Efficiency:**
    - **Objective:** Ensure models handle large datasets and are computationally efficient.
9. **Robustness and Generalization:**
    - **Objective:** Develop models that generalize well to new data and remain robust.
10. **Ethical Considerations:**
    - **Objective:** Address biases, fairness, and transparency in model predictions, considering ethical implications.

7.How Regression and Classification are different? Explain in detail with real time examples?

Regression and classification are two types of supervised learning techniques in machine learning, each addressing different types of prediction problems.

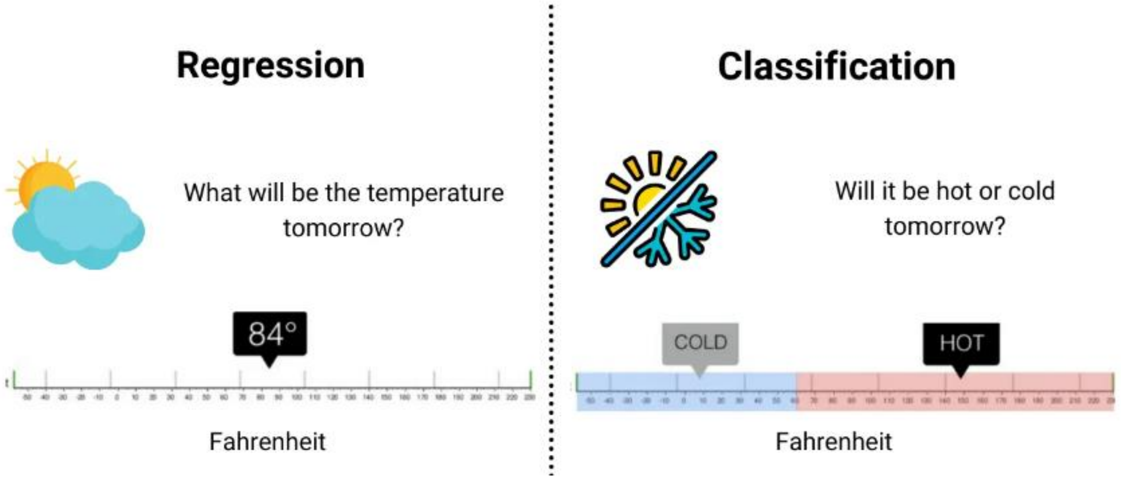
Regression:

- Regression involves predicting a continuous output or numeric value based on input features.
- The goal is to establish a relationship between the independent variables (features) and the dependent variable (output) to make accurate predictions.
- Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables.
- These are used to predict continuous output variables, such as market trends, weather prediction, etc.
- Some popular Regression algorithms are given below:
    - Simple Linear Regression Algorithm
    - Multivariate Regression Algorithm
    - Decision Tree Algorithm
    - Lasso Regression

Classification:

- Classification deals with predicting the category or class of an observation based on input features.
- It involves assigning instances to predefined classes or labels.
- Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as "Yes" or No, Male or Female, Red or Blue, etc.
- The classification algorithms predict the categories present in the dataset.
- Some real-world examples of classification algorithms are Spam Detection, Email filtering, etc.
- Some popular classification algorithms are given below:
    - Random Forest Algorithm
    - Decision Tree Algorithm
    - Logistic Regression Algorithm
    - Support Vector Machine Algorithm

Aspect	Regression	Classification
Task	The regression algorithm's task is mapping input value (x) with continuous output variable (y).	The classification algorithm's task mapping the input value of x with the discrete output variable of y.
Output Type	Continuous numerical values.	Discrete categorical labels or classes.
Nature of Output	Real number within a range	Class label or category
Evaluation Metrics	Uses metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE)	Uses metrics like Accuracy, Precision, Recall, F1 Score, etc.
Model Output	Numeric value	Probability scores or predicted class labels
Decision Boundary	No clear decision boundary in the output space	Decision boundary separates different classes
Division	We can further divide Regression algorithms into Linear and Non-linear Regression.	We can further divide Classification algorithms into Binary Classifiers and Multi-class Classifiers.
Use Cases	Financial forecasting, predicting sales, predicting temperature, etc.	Email spam detection, image recognition, sentiment analysis, etc.
Examples	Predicting house prices, stock prices, temperature, etc.	Spam detection, image recognition, sentiment analysis, etc.
Graph		


Regression
What will be the temperature tomorrow?
84°
Fahrenheit
Classification
Will it be hot or cold tomorrow?
COLD
HOT
Fahrenheit

Detailed Explanation with Real-Time Examples:

Regression:

1. **Housing Price Prediction:**
   - **Objective:** Predict the price of a house based on features like square footage, number of bedrooms, and location.
   - **Output:** Continuous numerical values representing the predicted price.
   - **Example:** Given a dataset of houses with known features and prices, a regression model can learn the relationship between the features and the house prices. For a new house with specified features, the model can predict its price.
2. **Temperature Forecast:**
   - **Objective:** Predict the temperature for the next day based on historical weather data and other relevant features.

- **Output:** Continuous numerical values representing the predicted temperature.
- **Example:** Using historical weather data (temperature, humidity, wind speed, etc.), a regression model can learn to predict the temperature for the following day.

Classification:

1. **Spam Email Detection:**
    - **Objective:** Classify emails as either spam or not spam based on their content and features.
    - **Output:** Discrete categorical labels (spam or not spam).
    - **Example:** A classification model trained on a dataset of labelled emails can learn to distinguish between spam and non-spam emails. It can then be used to classify new, unseen emails as spam or not spam.
2. **Image Classification (Cat vs. Dog):**
    - **Objective:** Identify whether an image contains a cat or a dog.
    - **Output:** Discrete categorical labels (cat or dog).
    - **Example:** Using a dataset of labelled images of cats and dogs, a classification model can learn to distinguish between the two. Given a new image, the model can predict whether it contains a cat or a dog.

8.How to estimate the loss function in statistical learning? Discuss how to assess model accuracy?

In statistical learning, estimating the loss function and assessing model accuracy are crucial steps in evaluating the performance of a predictive model.

Loss function:

- Loss is a value that represents the summation of errors in our model.
- To calculate the loss, a loss or cost function is used.
- The loss function, also referred to as the error function, is a crucial component in machine learning that quantifies the difference between the predicted outputs of a machine learning algorithm and the actual target values. If your predictions are totally off, your loss function will output a higher number. If they're pretty good, it'll output a lower number.
- For each prediction that we make, our loss function will simply measure the absolute difference between our prediction and the actual value.
- In mathematical notation, it might look something like

```
abs(y_predicted – y_actual).
```

Estimating the Loss Function:

The choice of a loss function depends on the nature of the problem (regression or classification). Common loss functions include:

1. **Mean Squared Error (MSE):**

    - MSE is commonly used to solve regression problems.
    - Mean squared error (MSE) is the workhorse of basic loss functions.
    - It's easy to understand and implement and generally works pretty well.
    - To calculate MSE, you take the difference between your predictions and the ground truth, square it, and average it out across the whole dataset.
    - Formula:

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

Where,

- o *n* is the number of observations.
- o Y_i is the actual value for the *i*-th observation.
- o $y^{\wedge}_i$ is the predicted value for the *i*-th observation.

    - A lower MSE indicates better performance.

2. **Likelihood loss:**

    - The likelihood function is also relatively simple, and is commonly used in classification problems.
    - The function takes the predicted probability for each input example and multiplies them.
    - And although the output isn't exactly human-interpretable, it's useful for comparing models.

3. **Cross-Entropy Loss (Log Loss):**

    - Log loss is a loss function also used frequently in classification problems,It's just a straightforward modification of the likelihood function with logarithms.
    - Measures the performance of a classification model whose output is a probability value.
    - Formula:

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\right]$$

    - Lower log loss indicates better performance.

<u>Accuracy:</u>

- Accuracy is a method for measuring a classification model's performance.
- It is typically expressed as a percentage.
- Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made.
- We calculate it by dividing the number of correct predictions by the total number of predictions.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

This formula provides an easy-to-understand definition that assumes a binary classification problem.

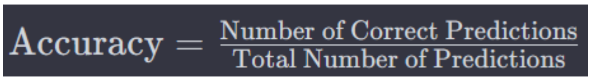
- Accuracy is often graphed and monitored during the training phase though the value is often associated with the overall or final model accuracy.
- Accuracy is easier to interpret than loss.

<u>Assess Model Accuracy:</u>

- Assessing model accuracy involves various metrics that provide insights into the model's overall performance.

Assessing Model Accuracy:

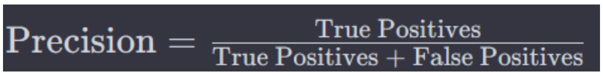
1. **Accuracy:**
   - **Formula:**



$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

   - **Interpretation:** The proportion of correctly predicted instances. Commonly used in classification problems.

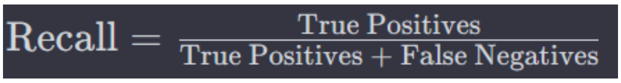
2. **Precision:**
   - precision is defined as the percentage of the correct predictions among the total prediction of a class between all of the classes present in the dataset.
   - **Formula:**



$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

   - **Interpretation:** The ability of the model to correctly identify positive instances among instances predicted as positive.

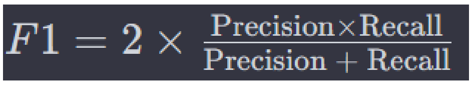
3. **Recall (Sensitivity or True Positive Rate):**
   - Recall is defined as the proportion of the correct predictions among the total prediction of a class between all of the classes present in the dataset.
   - **Formula:**



$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

   - **Interpretation:** The ability of the model to correctly identify all positive instances.

4. **F1 Score:**
   - F-score is defined as a metric combination of precision and also the recall.
   - **Formula:**



$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

   - **Interpretation:** The harmonic mean of precision and recall, providing a balance between the two metrics.

5. **Receiver Operating Characteristic (ROC) Curve:**
   - ROC is defined as the binary classification of the diagnostic plot of the curve.
   - Plots the true positive rate against the false positive rate at various threshold settings.

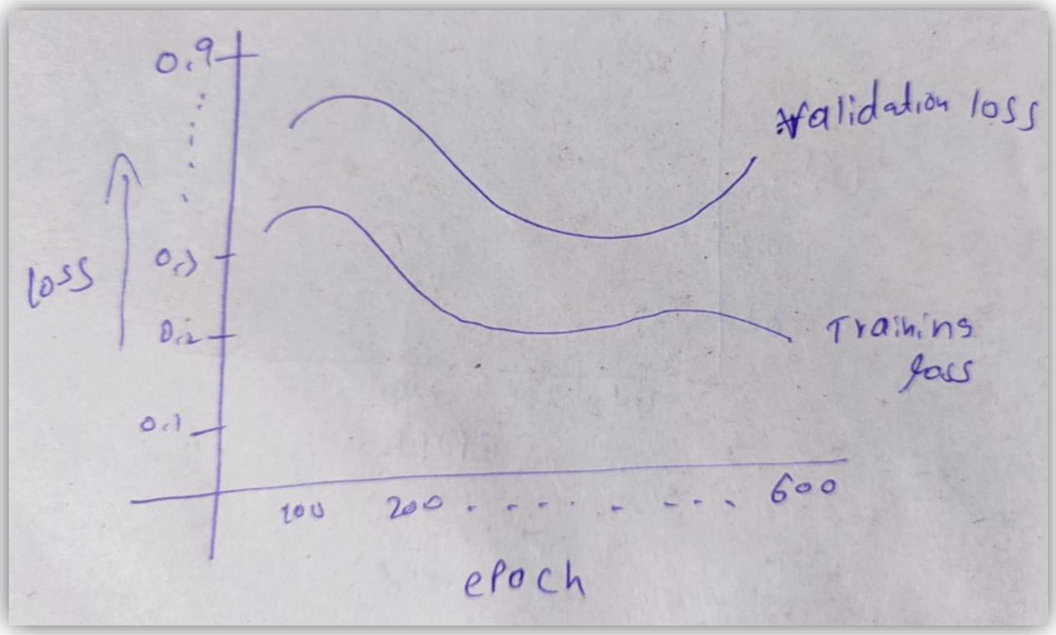
6. **Area Under the Curve (AUC):**
   - Measures the area under the ROC curve. AUC close to 1 indicates a good model.

9.Define the terms over fitting and under fitting? How can we resolve these issues?

Overfitting and underfitting are common issues in machine learning where the model's performance is affected negatively.

Overfitting:

- If the model performs well on the training data but poorly on the validation set, then this scenario is called Overfitting.



- Here,the validation loss decreases till it reaches a particular point then it increases whereas, training loss keeps decreasing.
- Overfitting happens when the model is too complex relative to the amount and noisiness of the training data or the model was trained for a long period.
- When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance.

Reasons for Overfitting:

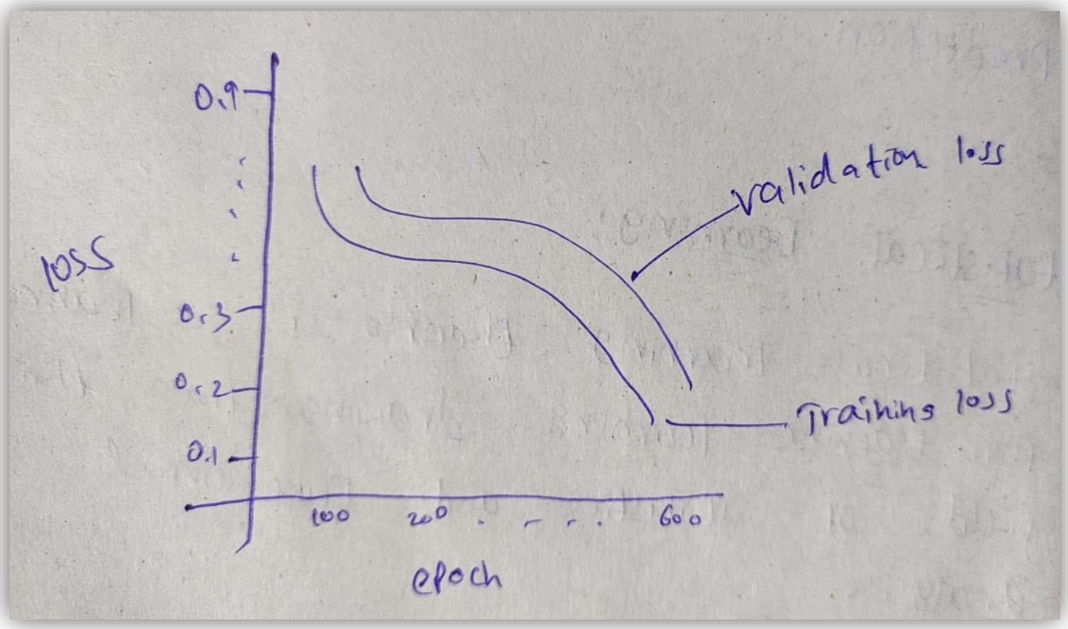
- High variance and low bias.
- The model is too complex.
- The size of the training data.

Techniques to Reduce Overfitting

- Increase training data.
- Reduce model complexity.
- Early stopping during the training phase (training can be halted when the loss is low and stable).
- By reducing the number of attributes in the training data or by constraining the model
- Ridge Regularization and Lasso Regularization.
- Use dropout for neural networks to tackle overfitting.

Underfitting:

- At times, the validation loss is greater than training loss. This may indicate that the model is underfitting.



- Here, both validation loss and training loss decreases.
- Underfitting occurs when the model is unable to accurately model the training data and hence generate errors.
- Underfitting is the opposite of overfitting.
- It occurs when your model is too simple to learn the underlying structure of the data.
- It represents the inability of the model to learn the training data effectively result in poor performance both on the training and testing data.
- In simple terms, an underfit model's are inaccurate, especially when applied to new, unseen examples.

Reasons for Underfitting

- The model is too simple, So it may be not capable to represent the complexities in the data.
- The input features which is used to train the model is not the adequate representations of underlying factors influencing the target variable.
- The size of the training dataset used is not enough.
- Excessive regularization are used to prevent the overfitting, which constraint the model to capture the data well.
- Features are not scaled.

Techniques to Reduce Underfitting

- Increase model complexity.
- Increase the number of features, performing feature engineering.
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.
- Increasing the training data.
- Furthur training.

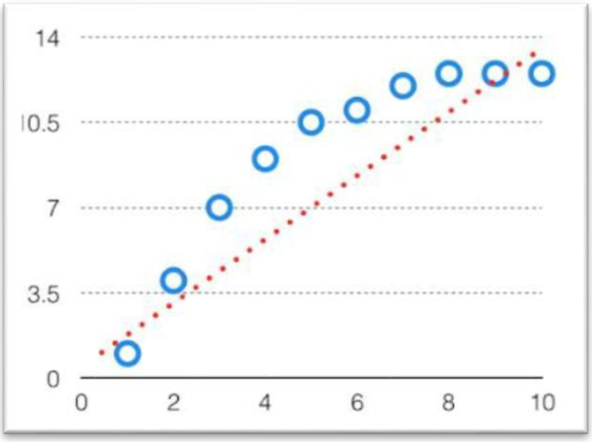
10.Describe bias and variance? Explain Bias variance trade-offs in statistical learning?

It is important to understand prediction errors (bias and variance) when it comes to accuracy in any machine learning algorithm. There is a trade-off between a model's ability to minimize bias and variance which is referred to as the best solution for selecting a value of **Regularization**(Constraining a model to make it simpler and reduce the risk of overfitting is called regularization.) constant. Proper understanding of these errors would help to avoid the overfitting and underfitting of a data set while training the algorithm.

Bias

The bias is known as the difference between the prediction of the values by the ML model and the correct value. Being high in biasing gives a large error in training as well as testing data. Its recommended that an algorithm should always be low biased to avoid the problem of underfitting.

By high bias, the data predicted is in a straight line format, thus not fitting accurately in the data in the data set. Such fitting is known as **Underfitting of Data**. This happens when the hypothesis is too simple or linear in nature. Refer to the graph given below for an example of such a situation.



High Bias

In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

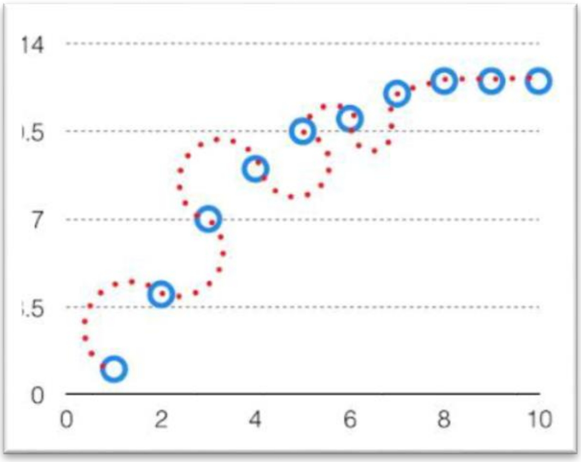
Variance

The variability of model prediction for a given data point which tells us spread of our data is called the variance of the model. The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

When a model is high on variance, it is then said to as **Overfitting of Data**. Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high.

While training a data model variance should be kept low.

The high variance data looks like follows.



High Variance

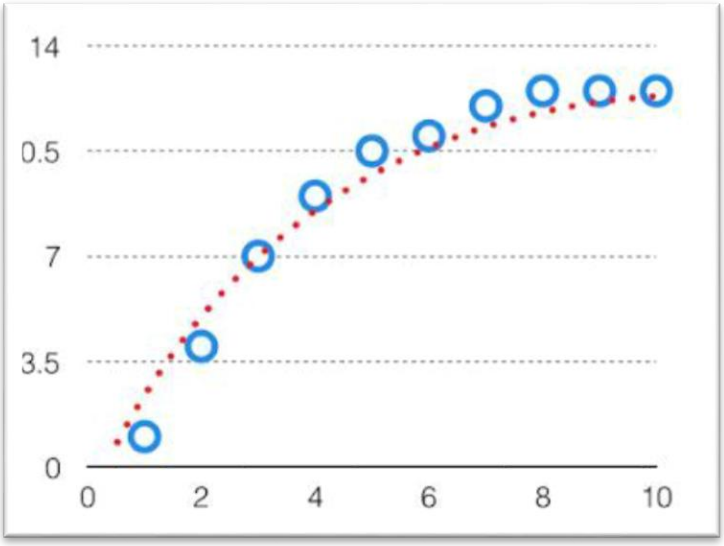
In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Bias Variance Tradeoff

If the algorithm is too simple (hypothesis with linear eq.) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree eq.) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as Trade-off or Bias Variance Trade-off.

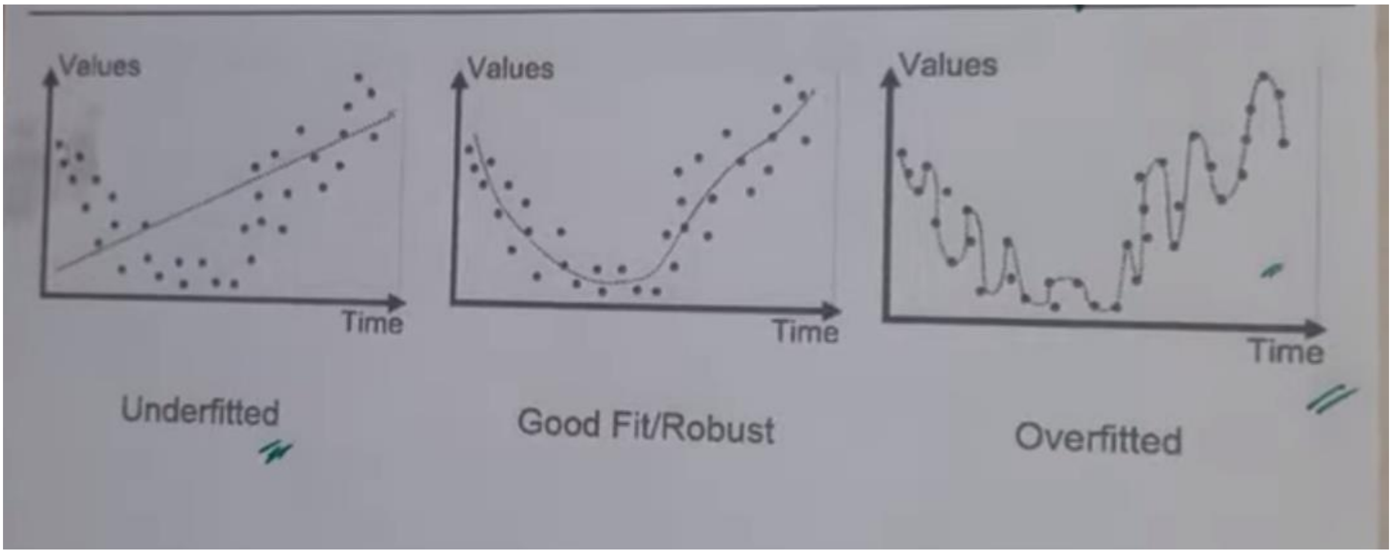
This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like.



We try to optimize the value of the total error for the model by using the Bias-Variance Tradeoff.

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

The best fit will be given by the hypothesis on the tradeoff point.
This is referred to as the best point chosen for the training of the algorithm which gives low error in training as well as testing data.


Values
Time
Underfitted
Values
Time
Good Fit/Robust
Values
Time
Overfitted