Statistical learning can be defined as a set of tools for understanding and analysing data. It can be categorised into supervised (with a response variable) and unsupervised (without a response variable) learning.

**Types of Problems in Statistical Learning**

* **Regression vs. Classification:** To put it simply, regression is like predicting your future salary based on your current job, while classification is like predicting whether it will rain tomorrow.
* **Supervised vs Unsupervised Learning**: Supervised learning involves predicting outputs using labelled data, while unsupervised learning explores relationships and structure without labelled outputs.

#### **Why Estimate ƒ?**

The primary goal of statistical learning is to estimate the relationship between a set of predictors **X** and a response variable **Y**. This relationship is often expressed as:

𝐘=**ƒ**(**X**) + ϵ

* **Explanation**:
  + **ƒ**(**X**): Represents the systematic information that **X** provides about **Y**.
  + ϵ: Represents the error term, capturing all the other unpredictable variations not explained by **X**. It is usually assumed to have a zero mean and captures randomness or measurement errors.
* **Goals of Estimating ƒ**:
  + **Prediction**: When **ƒ** is used to predict **Y** based on a new set of predictors **X**, the accuracy of these predictions is crucial.
  + **Inference**: In many cases, we want to understand how the input variables **X** affect the output **Y**. This is useful for gaining insights into the relationship between predictors and responses.

#### **How Do We Estimate ƒ?**

There are several approaches to estimate **ƒ**, which broadly fall under two categories:

1. **Parametric Methods**:
   * These methods involve assuming the functional form or shape of **ƒ**. For instance, linear models assume **ƒ** is a linear combination of the input variables.
   * **Steps in Parametric Methods**:
     1. **Assume the form of ƒ**: For example, a linear model might assume **ƒ**(**X**)=β0 + β1**X**1 + β2**X**2 + … + βp**X**p.
     2. **Estimate the parameters**: Find the best-fit values of β0,β1,…,βp that minimise the error in predicting **Y**.
   * **Advantages**: Simpler to interpret and often computationally more straightforward.
   * **Disadvantages**: Rigid; may not capture complex relationships.
2. **Non-Parametric Methods**:

* These methods do not assume any particular form of **ƒ**. Instead, they seek to estimate **ƒ** in a more flexible manner that can adapt to the underlying structure of the data.
* **Examples**: Decision trees, K-nearest neighbours, and support vector machines.
* **Advantages**: Can capture more complex relationships in the data.
* **Disadvantages**: Requires more data and can be more computationally intensive. Interpretability may also be lower compared to parametric methods.

#### **Supervised Versus Unsupervised Learning**

* **Supervised Learning**:
  + In supervised learning, each dataset observation has input (predictors) and output (response) variables.
  + **Examples**: Linear regression, logistic regression, decision trees.
  + **Goal**: To learn a mapping from inputs to outputs and make accurate predictions on new data.
* **Unsupervised Learning**:
  + In unsupervised learning, the dataset only contains input variables with no corresponding output variable.
  + **Examples**: Clustering, principal component analysis.
  + **Goal**: To find patterns or structure within the data

#### **Regression Versus Classification Problems**

* **Regression Problems**:
  + The response **Y** is quantitative or continuous.
  + **Examples**: Predicting house prices, estimating income based on experience and education.
* **Classification Problems**:
  + The response **Y** is categorical or qualitative.
  + **Examples**: Predicting whether an email is spam or not, classifying tumour cells as benign or malignant.
* **Example of a Regression Problem**: Predicting a person’s wage based on years of experience and education level.
* **Example of a Classification Problem**: Predicting whether a person will default on a loan based on their credit history.

**Measuring the Quality of Fit**

Different metrics and techniques evaluate how well a model fits the training data. The focus is on quantifying the difference between predicted and actual values using various measures.

1. **Training MSE (Mean Squared Error)**:
   * **Definition**: The Mean Squared Error is a commonly used measure of the quality of fit. It is computed by taking the average of the squared differences between the predicted values (**Ŷ**i​) and the actual values (**Y**i​) for all observations in the training set.
   * **Formula**: MSETraining=(**Y**i - **Ŷ**i​)2
   * **Interpretation**: A smaller training MSE indicates a model that better fits the training data. However, a low training MSE does not guarantee good performance on new data.
2. **Test MSE**:
   * **Definition**: The Test Mean Squared Error measures how well the model performs on unseen or new data. It is computed similarly to training MSE, but using a separate test set.
   * **Formula**: MSETest=(**Y**i - **Ŷ**i​)2
   * **Interpretation**: The test MSE is often higher than the training MSE. A low test MSE is a good indicator that the model generalises well to new data.

#### **The Bias-Variance Trade-Off**

The concept of bias-variance trade-off helps explain why a model may have a low training error but a high test error, which can affect its performance.

1. **Understanding the Trade-Off**:
   * The **expected test MSE** can be broken down into three components:

=

* + - : Measures how far the average model prediction is from the true function **ƒ**(). High bias indicates underfitting.
    - : Measures how much varies when we use different training datasets. High variance indicates overfitting.
    - : The irreducible error, representing noise in **Y**.

1. **Bias-Variance Trade-Off**:
   * **High-Bias Models**: Simple models (e.g., linear regression) have high bias but low variance. They may underfit the data, leading to poor predictive performance.
   * **High-Variance Models**: Complex models (e.g., high-degree polynomials or deep neural networks) have low bias but high variance. They may overfit the data, capturing noise along with the underlying pattern.
   * The goal is to find a balance between bias and variance to minimise the overall test MSE.
2. **Graphical Representation**:
   * A plot of **training error** and **test error** against model complexity is typically U-shaped:
     + As model complexity increases, training error decreases.
     + However, test error initially decreases and then increases, indicating overfitting.

#### **The Classification Setting**

The primary goal is to assign observations to predefined categories or classes correctly.

1. **Training Error Rate**
   * The **training error rate** measures the proportion of observations in the training set that the model incorrectly classifies.
   * **Formula**: Training Error Rate=**I**(**y**i **ŷ**i​)
     + n = Total number of observations in the training set.
     + **I**(**y**i **ŷ**i​) is an indicator function that equals 1 if the true class **y**i does not match the predicted class **ŷ**i, and 0 otherwise.
   * A low training error rate indicates a good fit to the training data, but just like with regression, it does not guarantee good performance on new data (i.e., it does not guarantee a low test error rate).
2. **Test Error Rate and Bayes Classifier**
   * **Test Error Rate**: The test error rate is calculated similarly to the training error rate, but on a separate test dataset. It measures how well the model generalises to new, unseen data.
   * **Bayes Classifier**:
     + The Bayes classifier is the **optimal classifier** that minimises the test error rate. It assigns each observation to the most probable class based on its predictor values.
     + **Bayes Decision Boundary**: The boundary that separates the regions where the Bayes classifier assigns different classes. The Bayes decision boundary may be non-linear or complex depending on the true relationship between **X** and **Y**.
     + The Bayes classifier is not feasible in most cases because it requires knowing the true distribution of **Y** given **X**, denoted as .
3. **Bayes Error Rate**
   * Even with the optimal Bayes classifier, there will still be some misclassification error due to the **irreducible error** or **Bayes error rate**.
   * The Bayes error rate represents the lowest possible error rate achievable and is given by: Bayes Error Rate=1− . Even if we know the true distribution , there may still be overlap between classes, leading to unavoidable misclassification.
4. **K-Nearest Neighbors (KNN) Classifier**
   * The K-Nearest Neighbors (KNN) classifier is introduced as a **non-parametric** alternative to the Bayes classifier.
   * **How KNN Works**:
     + Given a new observation, KNN identifies the k-nearest points in the training data and assigns the most common class among these neighbours as the predicted class.
     + **Choice of k**:
       - A small k (e.g., k=1) means that the decision boundary will be highly flexible, potentially leading to overfitting and a low bias but high variance.
       - A large k results in a smoother decision boundary, reducing variance but increasing bias.
   * **Trade-Off**:
     + KNN can approximate the Bayes classifier if k is chosen appropriately and there is enough data.
     + However, KNN can become computationally expensive for large datasets, as it requires calculating the distance from the test observation to every point in the training set.
5. **Comparison of Classifiers**
   * + The Bayes classifier is optimal but often unattainable.
     + KNN can be a good approximation to the Bayes classifier but is sensitive to the choice of k.
     + Simpler models, such as linear classifiers, have higher bias but are easier to interpret and often perform well on smaller datasets.