ML Model Basics

Supervised Learning: Supervised learning is a machine learning approach where the algorithm learns from labeled training data. Labeled data means that each example in the dataset is associated with a corresponding target or output value. The goal of supervised learning is to train a model that can predict the correct output for new, unseen inputs.

Example: Predicting whether an email is spam or not. The model is trained on a dataset of emails, where each email is labeled as spam or not spam. It learns from these labeled examples and can then classify new, unseen emails as spam or not based on the patterns it has learned.

Unsupervised Learning: Unsupervised learning is a machine learning approach where the algorithm learns from unlabeled data, meaning there are no predefined target or output values. Instead, the algorithm explores the patterns, structures, and relationships within the data on its own

Example: Clustering customer data based on purchasing behavior. The algorithm analyzes a dataset of customer transactions without any labels or predefined groups. It discovers natural clusters of customers who exhibit similar purchasing behavior, allowing businesses to segment their customers for targeted marketing campaigns.

Supervised Learning Models

- Linear Regression
- Logistic Regression
- Decision Tree
- Random Forest
- Naïve Bayes
- KNN
- SVM

Unsupervised learning Models

- K-Means clustering
- DBSCAN
- PCA

In Supervised Learning the type of models we can use depends on the target/outcome variable.
Outcome/target variable

- Numerical outcome => regression problem => Linear Regression Model
- Categorical outcome => classification problem => Logistic regression or Naïve Bayes

Also, note that the below models can be used for **both** regression type and classification type problems:

- KNN (e.g., majority vote vs. averaging)
- Decision Tree (e.g., ID3 vs. CART)
- SVM (e.g., SVC vs. SVR)

Linear regression:

Example: Predicting Blood Pressure based on Age

Suppose a medical researcher wants to understand the relationship between a patient's age and their blood pressure. They collect data from a group of patients, recording their ages and corresponding blood pressure readings. The researcher can use linear regression to build a model that predicts a patient's blood pressure based on their age.

Advantages of Linear Regression:

- **Identifying Correlations**: Linear regression helps determine if there is a linear relationship between age and blood pressure. It can quantify the strength and direction of this relationship.
- **Interpretability**: The coefficient associated with age in the linear regression equation provides insights into how blood pressure changes with age.

Disadvantages of Linear Regression:

- **Linearity Assumption**: Linear regression assumes a linear relationship between age and blood pressure. If the relationship is not linear, the model may not accurately capture the pattern.
- **Outliers**: Extreme blood pressure values in the dataset can influence the model's fit and predictions.
- **Limited Complexity**: Linear regression may not capture more complex interactions between age and other factors influencing blood pressure.

Logistic Regression:

Suppose a medical researcher wants to predict whether a patient is at risk of developing a specific medical condition based on their age. The researcher collects data from a group of patients, including their ages and whether they have a medical condition or not. Using logistic regression, the researcher can build a model to predict the probability of a patient being at risk of a medical condition based on their age.

Advantages of Logistic Regression:

- **Probability Output**: Provides probabilities of outcomes, allowing for better risk assessment and decision-making.
- **Versatility**: Suitable for binary classification tasks and can be extended to handle multiple classes (multinomial logistic regression).
- **Interpretability**: Coefficients represent the influence of variables on the likelihood of the outcome.

Disadvantages of Logistic Regression:

- **Limited Complexity**: This may not capture complex non-linear relationships between variables.
- **Feature Engineering**: Requires careful selection and transformation of features to avoid multicollinearity and enhance model performance.
- **High Sensitivity to Outliers**: Outliers can significantly impact the model's performance.

Decision Trees

Suppose a group of doctors wants to diagnose respiratory disease in patients based on their symptoms and test results. They have collected data from patients, including symptoms like cough, shortness of breath, and chest pain, as well as the results of various diagnostic tests such as X-rays and blood tests.

The decision tree algorithm can be applied to build a model that helps doctors make accurate diagnoses for new patients based on their symptoms and test results.

Advantages of Decision Trees :

- **Interpretability**: The decision tree's structure allows doctors to trace the diagnosis process step-by-step, providing clear insights into how certain symptoms and test results influence the final diagnosis.
- **Rule Extraction**: Decision trees can be converted into a set of rules that can be used as guidelines for diagnosis.
- **Non-linear Patterns**: Decision trees can handle non-linear relationships between symptoms and diseases, capturing complex patterns that may not be apparent with simple linear models.

Disadvantages of Decision Trees:

- **Overfitting**: If the decision tree becomes too complex, it may overfit the training data and provide less accurate predictions for new patients.
- **Limited Generalization**: Decision trees may not generalize well to patients with unique combinations of symptoms or test results that were not present in the training dataset.
- **Data Quality**: The accuracy of decision trees heavily depends on the quality and completeness of the data. Incomplete or noisy data can lead to inaccurate diagnoses.

Random Forest

Suppose a team of medical researchers wants to diagnose breast cancer in patients based on various features extracted from mammograms and patient characteristics. The features may include tumor size, texture, cell shape, and patient age, among others.

The researchers can use Random Forest to build a model that predicts whether a patient's breast tumor is cancerous or non-cancerous based on the combination of these features.

Advantages of Random Forest:

- **High Accuracy**: Random Forest tends to achieve higher accuracy compared to individual decision trees, as it aggregates predictions from multiple trees, reducing the risk of overfitting and improving generalization.
- **Robustness**: Random Forest is less sensitive to outliers and noise in the data, making it more robust and reliable, especially in medical datasets where data quality can be challenging.

- **Feature Importance**: Random Forest provides a measure of feature importance, allowing researchers to identify the most relevant features contributing to the diagnosis.
- **Handling High-Dimensional Data**: Random Forest performs well even in datasets with a large number of features, making it suitable for medical applications with complex data.

Disadvantages of Random Forest:

- **Interpretability**: Although Random Forest provides valuable feature importance, the overall model's decision-making process can be more challenging to interpret compared to individual decision trees.
- **Computation Time**: Building and training multiple decision trees in Random Forests can be computationally expensive, especially for large datasets with numerous trees.
- **Overfitting with Noise**: Although Random Forest is less prone to overfitting if the dataset contains significant noise, it may still affect the model's performance

.**K-Nearest Neighbors (KNN)**:

Suppose we have a dataset of various fruits, each described by two features: sweetness and acidity. We want to classify new fruits into one of two categories: "apple" or "orange" based on their sweetness and acidity.

Advantages of KNN:

- **Simplicity**: KNN is easy to understand and implement, making it a great starting point for classification tasks.
- **No Training Phase**: KNN does not have an explicit training phase; the model uses the entire dataset for predictions.
- **Non-linearity**: KNN can capture non-linear decision boundaries, making it suitable for datasets with complex relationships.

Disadvantages of KNN:

- **Computationally Expensive**: As the dataset grows, KNN's prediction time increases significantly since it involves calculating distances to all data points.
- **Sensitivity to Irrelevant Features**: KNN considers all features equally, making it sensitive to irrelevant or noisy features.
- **Need for Proper Scaling**: Features should be scaled to have similar magnitudes; otherwise, features with larger scales can dominate the distance calculations.

Support Vector Machine (SVM):

Suppose we have a dataset with two classes: "positive" and "negative." The classes are not linearly separable in the feature space. SVM can be used to find the optimal hyperplane that best separates the two classes.
Advantages of SVM:

- **Effective in High-Dimensional Spaces**: SVM performs well even in high-dimensional feature spaces, making it suitable for complex datasets.
- **Robust to Overfitting**: SVM aims to maximize the margin between classes, which helps to avoid overfitting and improves generalization to unseen data.
- **Flexibility in Kernel Choices**: SVM allows the use of different kernel functions (e.g., linear, polynomial, radial basis function) to capture complex relationships between data points.

Disadvantages of SVM:

- **Computationally Intensive**: Training SVM models can be computationally expensive, especially for large datasets.
- **Model Selection**: Selecting the appropriate kernel and tuning hyperparameters can be challenging and require cross-validation.
- **Memory Usage**: SVM memory requirements increase with the number of support vectors, which can be an issue for large datasets.