Machine Learning with python

SOMMARIO

1 SUMMARY

- **Machine learning** is the ability of computers to learn from data without explicit programming. It has various applications and techniques, such as:
    - Recommendation systems
    - Classification
    - Clustering
    - Anomaly detection
- **AI, machine learning, and deep learning** are related but different concepts. AI aims to make computers intelligent, covering various fields. Machine learning is the statistical part of AI, teaching computers to solve problems based on examples. Deep learning is a subset of machine learning where computers can learn and make decisions independently.
- **Supervised learning** is when the model is trained with labeled data and can perform tasks like classification and regression. **Unsupervised learning** is when the model is trained with unlabeled data and can discover hidden information, such as:
    - Clusters
    - Patterns
    - Associations
- **Regression** is the process of predicting a continuous value, such as CO2 emission, sales, or house prices, using one or more independent variables. There are different types of regression models, such as:
    - **Simple regression**: One independent variable to estimate a dependent variable (linear or non-linear).
    - **Multiple regression**: More than one independent variable (linear or non-linear).
- **Classification** is a supervised learning technique that assigns a discrete label to an input instance based on its features. Classification algorithms include logistic regression, naive Bayes, support vector machines, etc. Classification can be used for tasks like spam filtering, face recognition, customer churn prediction, loan approval, etc.
- **Clustering** is an unsupervised learning technique that groups similar instances based on their features. Clustering algorithms include k-means, fuzzy c-means, Gaussian mixture models, etc. Clustering can be used for tasks like dimensionality reduction, market basket analysis, density estimation, etc.

2 INTRODUCTION TO MACHINE LEARNING

1. **Example Scenario:**
   - Analysis of human cell samples to predict benign or malignant cells using a dataset of characteristics.
   - Importance of early cancer diagnosis.
2. **Machine Learning Definition:**
   - Definition: "Computers' ability to learn without being explicitly programmed."
   - Example: Recognizing animals in images without explicit programming.
3. **Learning Process:**
   - Feature interpretation and transformation into sets.
   - Traditional approach vs. machine learning.
   - Iterative learning from data to build models without explicit programming.
4. **Real-Life Examples:**
   - Netflix and Amazon recommendations.
   - Loan approval decisions by banks.
   - Telecommunication customer segmentation.
   - Various applications like chatbots and face recognition in games.
5. **Machine Learning Techniques:**
   - Regression/Estimation for continuous value prediction.
   - Classification for predicting categories.
   - Clustering for grouping similar cases.
   - Association for finding co-occurring items.
   - Anomaly detection for discovering abnormal cases.
   - Sequence mining for predicting the next event.
   - Dimension reduction for data size reduction.
   - Recommendation systems for suggesting items based on preferences.
6. **Difference Between AI, Machine Learning, and Deep Learning:**
   - AI aims to make computers intelligent, covering various fields.
   - Machine Learning is the statistical part of AI, teaching computers to solve problems based on examples.
   - Deep Learning is a subset of Machine Learning where computers can learn and make decisions independently.

2.1 Introduction to Supervised and Unsupervised Learning

1. **Supervised Learning:**
   - Definition: Teaching a machine learning model by observing and directing its tasks.
   - Training the model with labeled data from a dataset.
   - Example: Cancer dataset with historical patient data and class labels.
   - Attributes, features, observations, numerical, and categorical data explained.
   - Two types: Classification (discrete class labels) and Regression (predicting continuous values).
2. **Unsupervised Learning:**
   - Definition: Allowing the model to work independently to discover hidden information.
   - Training on unlabeled data, drawing conclusions without supervision.
   - More challenging algorithms due to limited information about data and outcomes.
   - Widely used techniques: Dimension reduction, density estimation, market basket analysis, and clustering.
3. **Unsupervised Techniques Explained:**
   - Dimensionality reduction and feature selection aid in classification.
   - Market basket analysis predicts item groups based on purchasing behavior.
   - Density estimation explores data structure.
   - Clustering groups similar data points or objects.
4. **Comparison:**
   - Supervised learning deals with labeled data; unsupervised learning deals with unlabeled data.
   - Supervised: Classification, Regression; Unsupervised: Clustering.
   - Unsupervised learning has fewer models and evaluation methods.

3 Regression

Overview of regression, focusing on predicting continuous values.

1. **Example Dataset:**
   - CO2 emissions dataset with variables: engine size, number of cylinders, fuel consumption, and CO2 emission for various car models.
   - Question: Can we predict CO2 emission using other variables like engine size or cylinders?
2. **Regression Basics:**
   - Regression is the process of predicting a continuous value.
   - Two types of variables: dependent (Y) and independent (X).
   - Regression model relates Y to a function of X.
3. **Types of Regression Models:**
   - **Simple regression**: One independent variable to estimate a dependent variable (linear or non-linear).
   - **Multiple regression**: More than one independent variable (linear or non-linear).
4. **Applications of Regression:**
   - Sales forecasting: Predicting total yearly sales based on variables like age, education, and experience.
   - Psychology: Determining individual satisfaction using demographic and psychological factors.
   - Real estate: Predicting house prices based on size, number of bedrooms, etc.
   - Employment income prediction using variables like hours of work, education, occupation, etc.
5. **Diverse Applications:**
   - Regression analysis is widely used in various fields such as finance, healthcare, retail, and more.

3.1 Linear Regression

Linear regression is introduced as a method for predicting a continuous value, e.g. CO2 emission prediction in cars.

1. **Types of Linear Regression Models:**
   - Simple linear regression: One independent variable used to estimate a dependent variable.
   - Multiple linear regression: More than one independent variable.
2. **How Linear Regression Works:**
   - Explanation of plotting variables, showing their linear relation in a scatter plot.

- Linear regression fits a line through the data, modeling the relationship between variables.
- Linear regression helps predict the approximate emission of each car.

3. **Parameters of Linear Regression:**
    - Two parameters: Theta 0 (intercept) and Theta 1 (slope or gradient).
    - These parameters are coefficients of the linear equation.
4. **Fitting the Line and Minimizing Error:**
    - Mean Squared Error (MSE) is used to measure how well the line fits the data.
    - Objective: Minimize MSE by adjusting parameters Theta 0 and Theta 1.
5. **Calculating Parameters Theta 0 and Theta 1:**
    - Equations provided for estimating Theta 1 (slope) and Theta 0 (intercept) using data averages.
6. **Polynomial Representation of Linear Regression:**
    - Linear regression equation represented as a polynomial: $yhat=\vartheta0+\vartheta1x1$.
7. **Using the Linear Model for Prediction:**
    - Once parameters are found, predictions for new data points are made using the linear model equation.
8. **Benefits of Linear Regression:**
    - Linear regression is fast, requires no parameter tuning, and is easy to understand.
    - Highly interpretable and a fundamental regression method.

3.2 Model Evaluation in Regression Models

Model evaluation is discussed in the context of regression for predicting unknown cases accurately.

1. **Two Evaluation Approaches:**
    - **Train and Test on the Same Dataset:**
        - The model is trained on the entire dataset, then tested on a portion of the same dataset.
        - It results in high training accuracy but may have low out-of-sample accuracy, risking overfitting.
        - Training accuracy is the percentage of correct predictions on the test dataset.
    - **Train/Test Split:**
        - A portion of the dataset is used for training, and the rest for testing, providing more realistic out-of-sample accuracy.

        - A model is built on the training set, and its predictions are compared with the actual values of the testing set.
        - A better approach for real-world problems to ensure the model's performance on unknown data.

2. **Out-of-Sample Accuracy and Overfitting:**
    - Out-of-sample accuracy is crucial for making correct predictions on unknown data.
    - Overfitting occurs when a model is overly trained on the dataset, capturing noise and lacking generalization.
3. **Improving Out-of-Sample Accuracy:**
    - **Train/Test Split:**
        - Ensures that the testing dataset is not part of the dataset used for training.
        - Realistic out-of-sample testing, but dependent on specific datasets.
    - **K-Fold Cross-Validation:**
        - Addresses issues of dependency in train/test split.
        - The dataset is split into K folds, and the model is trained and tested on different folds.
        - Averages the results to produce a more consistent out-of-sample accuracy.
        - Provides a solution to high variation resulting from dependency in train/test split.
4. **K-Fold Cross-Validation:**
    - Represents the dataset using multiple folds, each used for testing in a distinct iteration.
    - The accuracy of each fold is calculated and then averaged to produce a more reliable out-of-sample accuracy.
    - Provides a more thorough evaluation of the model's performance.

3.3 Evaluation Metrics in Regression Models

Evaluation metrics are crucial for assessing the performance of a model and play a key role in identifying areas that require improvement.

1. **Metrics for Regression Models:**
    - In regression, accuracy is calculated by comparing actual values with predicted values.
2. **Error in Regression Models:**
    - Error in regression models is defined as the difference between data points and the trend line generated by the algorithm.
    - Multiple ways to determine errors due to multiple data points.
3. **Key Evaluation Metrics:**

- **Mean Absolute Error (MAE):**
    - Represents the average of the absolute values of errors.
    - Simple to understand, as it's the average error.
- **Mean Squared Error (MSE):**
    - Represents the mean of the squared errors.
    - More popular than MAE, focuses more on larger errors due to the squared term.
- **Root Mean Squared Error (RMSE):**
    - Represents the square root of the mean squared error.
    - Highly popular, interpretable in the same units as the response vector (Y units).
- **Relative Absolute Error (Residual Sum of Square):**
    - Normalizes the total absolute error by dividing it by the total absolute error of the simple predictor.
- **Relative Squared Error:**
    - Similar to relative absolute error, widely adopted for calculating R-squared.
- **R-squared:**
    - Not an error metric but a popular accuracy metric for the model.
    - Represents how close the data values are to the fitted regression line.
    - Higher R-squared values indicate a better fit of the model to the data.

4. **Choice of Metric:**
    - The selection of a specific metric depends on factors like the type of model, data type, and domain knowledge.
    - Each metric provides a quantifiable measure of prediction accuracy.

3.4 Multiple Linear Regression

Multiple linear regression involves multiple independent variables to predict the dependent variable.

1. **Applications of Multiple Linear Regression:**
    - Identifying the strength of the effect of independent variables on the dependent variable.
    - Predicting the impact of changes, understanding how the dependent variable changes with changes in independent variables.
2. **Model Representation:**
    - Multiple linear regression predicts a continuous variable (e.g., CO2 emission) using multiple independent variables.

   - Model is represented as $yhat=\vartheta0+\vartheta1x1+\vartheta2x2+...+\vartheta nxn$.
   - Vector form: ϑTx where ϑ is a vector of coefficients and x is a vector of feature sets.

3. **Objective of Multiple Linear Regression:**
   - Find the best fit hyperplane for the data by minimizing the Mean Squared Error (MSE).
4. **Parameter Estimation:**
   - Parameters (ϑ) can be estimated using methods like *Ordinary Least Squares* or optimization algorithms like *Gradient Descent*.
   - Ordinary Least Squares minimizes mean squared error through linear algebra operations.
   - Optimization algorithms iteratively minimize errors by adjusting coefficients.
5. **Prediction Phase:**
   - After finding the parameters, predictions are made by solving the linear regression equation for a specific set of inputs.
6. **Concerns and Considerations:**
   - Using too many independent variables without justification may lead to overfitting.
   - Overfit models are too complex for the dataset and lack generalization.
   - Categorical independent variables can be included by converting them into numerical variables.
   - There should be a linear relationship between the dependent variable and each independent variable.
7. **Linear Relationship Check:**
   - Scatter plots can be used to visually check for linearity.
   - If the relationship is not linear, non-linear regression may be needed.

4 CLASSIFICATION

Classification is a supervised learning approach in machine learning. It involves categorizing or classifying unknown items into discrete classes. The goal is to learn the relationship between feature variables and a categorical target variable.

1. **Working of Classification and Classifiers:**
    - Given a set of labeled training data, classification determines the class label for an unlabeled test case.
    - Example: Loan default prediction, where previous data is used to predict future defaults.
    - The classifier is trained on features like age, income, education, and labels data points as defaulters or non-defaulters.
2. **Binary and Multi-Class Classification:**
    - Classification can be binary (two classes) or multi-class (more than two classes).
    - Example: Predicting drug response with three medications is a multi-class classification scenario.
3. **Business Use Cases of Classification:**
    - Predicting customer categories, churn detection, and response to advertising campaigns are common business use cases.
    - Classification finds applications in various industries, solving problems with labeled data.
4. **Applications of Classification:**
    - Email filtering, speech recognition, handwriting recognition, biometric identification, and document classification are some applications.
5. **Types of Classification Algorithms:**
    - Decision trees, naive Bayes, linear discriminant analysis, k-nearest neighbor, logistic regression, neural networks, and support vector machines are examples of classification algorithms.

4.1 K-NEAREST NEIGHBOURS

KNN is a classification algorithm used for predicting the class of unknown cases based on their similarity to labeled cases. E.g. a telecommunications provider predicting customer groups based on demographic data.

1. **Basic Working of KNN:**
    - Given a dataset with predefined labels, the algorithm classifies new or unknown cases by finding their nearest neighbors in the dataset.
    - The example uses demographic features like age and income to predict customer groups.
2. **Selection of Neighbors (K):**
    - The value of K determines the number of nearest neighbors to consider.

- Choosing a low K may result in capturing noise or anomalies, leading to overfitting.
- Choosing a high K results in an overly generalized model.

3. **Selecting the Right Value for K:**
    - The general solution involves reserving part of the data for testing the model's accuracy.
    - Start with K=1, use the training part for modeling, and calculate accuracy on the test set.
    - Repeat the process, increasing K, and choose the K that gives the best accuracy.
4. **Dissimilarity Measures:**
    - Dissimilarity between two data points, e.g., customers, can be calculated using measures like Euclidean distance.
    - Normalizing features is important for accurate dissimilarity measures.
    - The choice of dissimilarity measure depends on data type and the classification domain.
5. **Continuous Target Prediction with KNN:**
    - KNN can also be used for predicting continuous target values.
    - For example, predicting the price of a home based on features involves finding nearest neighbors and predicting the price as the median of neighbors.
6. **Avoiding Overfitting:**
    - Overfitting is undesirable as it results in a model that is too specific to the training data and may not generalize well to new, unseen data.
    - The goal is to find a balanced K that provides a general and accurate model.

4.2 Classifier Evaluation Metrics:

Evaluation metrics are crucial for assessing the performance of a model.

1. **Jaccard Index:**
    - The Jaccard index, or Jaccard similarity coefficient, measures accuracy.
    - It calculates the size of the intersection divided by the size of the union of true and predicted label sets.
    - Subset accuracy is 1.0 if the entire predicted label set strictly matches the true set.
2. **Confusion Matrix:**
    - A confusion matrix provides a detailed breakdown of correct and wrong predictions.
    - Rows represent actual/true labels, and columns represent predicted labels.
    - True positives, false negatives, true negatives, and false positives can be derived from the matrix.
3. **Precision and Recall:**
    - Precision measures accuracy when a class label has been predicted.

- Recall is the true positive rate.
- Precision = True Positive / (True Positive + False Positive).
- Recall = True Positive / (True Positive + False Negative).

4. **F1 Score:**
    - F1 score is the harmonic average of precision and recall.
    - F1 score reaches its best value at 1 and worst at 0.
    - It provides a good balance between precision and recall.
    - Average accuracy for the classifier can be calculated as the average F1 score across labels.
5. **Log Loss:**
    - Logarithmic loss (Log loss) measures classifier performance when the predicted output is a probability value between 0 and 1.
    - For example, predicting a low probability (e.g., 0.13) when the actual label is 1 results in high log loss.
    - Log loss is calculated for each row and averaged across the test set.
    - Ideal classifiers have progressively smaller log loss values.

4.3 Decision Trees

Decision trees are a powerful tool for classification tasks. They help make decisions by splitting the dataset into distinct nodes, with each node containing or representing a specific category of data.

1. **Medical Research Example:**
    - Imagine a medical researcher compiling data for a study on patients who suffered from the same illness.
    - Each patient responded to either drug A or drug B during their treatment.
    - Features in the dataset include age, gender, blood pressure, and cholesterol, and the target is the drug each patient responded to.
2. **Decision Tree Construction:**
    - Decision trees are built by splitting the training set into nodes based on attributes.
    - Internal nodes correspond to tests on attributes, branches represent test results, and leaf nodes assign patients to classes (in this case, drug A or drug B).
3. **Decision-Making Process:**
    - The decision-making process involves testing attributes and branching based on the results.
    - Each decision node corresponds to a test, and each branch corresponds to a result of the test.
    - Leaf nodes assign patients to a specific class or category.

4. **Building Decision Trees:**
    - Decision trees are constructed by considering attributes one by one.
    - The process involves choosing an attribute, calculating its significance in splitting the data, and then splitting the data based on the value of the best attribute.
    - The tree is built recursively by repeating the process for each branch.
5. **Predictive Power:**
    - Once the decision tree is built, it can be used to predict the class of unknown cases.
    - In the medical research example, the decision tree can help determine the appropriate drug for a new patient based on their characteristics.

4.3.1 Decision Tree Building Process

Decision trees are constructed using recursive partitioning to classify data. The algorithm chooses the most predictive feature to split the data on.

1. **Attribute Selection:**
    - The goal is to determine which attribute is the best or most predictive to split the data.
    - The choice of attribute is crucial for the purity of leaves after the split.
2. **Example Attribute Selection:**
    - Using a drug dataset example, attributes like cholesterol and sex are considered.
    - For each attribute, the dataset is split, and the impurity (entropy) of resulting nodes is evaluated.
3. **Entropy Calculation:**
    - Entropy is the amount of information disorder or randomness in the data.
    - Nodes with zero entropy are completely homogeneous, while nodes with equal division have entropy of one.
    - Entropy is calculated for each node based on the distribution of target categories.
4. **Information Gain:**
    - Information gain measures the increase in certainty after splitting.
    - It is the entropy of the tree before the split minus the weighted entropy after the split by an attribute.
5. **Attribute Comparison:**
    - Attributes are compared based on information gain.
    - The attribute resulting in higher information gain is considered a better choice for splitting.
6. **Recursive Process:**
    - The process is repeated for each branch, testing attributes to achieve the most pure leaves.

- Decision trees are built iteratively, selecting attributes at each level to optimize information gain.

7. **Decision Tree Construction:**

    - The construction involves selecting attributes that maximize information gain, recursively branching the tree until leaves are pure.

4.4 Regression Trees

Decision trees can be used not only for classification but also for regression, known as regression trees. The basic idea is to split data based on features and return a prediction that is the average across the data in each group.

1. **Example Using Housing Data:**

    - Consider a housing dataset where 'Age' is used to predict 'Price'

    - The dataset is divided into groups based on age ranges, and the average price is calculated for each group.

2. **Criterion for Regression Trees:**

    - Instead of using the entropy criterion as in classification trees, regression trees use criteria to minimize error.

    - Mean Absolute Error (MAE) is a popular criterion for regression trees.

3. **How Regression Trees are Built:**

    - Start by deciding the first decision by checking every feature in the dataset to see which one produces the minimal error.

    - Categorical features involve calculating the average price of houses in each category to determine the average error.

    - Numerical features require finding a boundary between data points and calculating the error on each side of the boundary.

4. **Stopping Criteria:**

    - Common conditions to stop growing regression trees include tree depth, the number of remaining samples on a branch, and the number of samples on each branch if another decision is made.

5. **Adding Decisions:**

    - The process of adding decisions involves calculating MAE for both categorical and numerical features and selecting the feature that results in the lowest MAE.

6. **Final Result:**

    - The regression tree is built by adding decisions based on features that minimize the error.

    - The depth of the tree and the number of samples on each branch are considerations for determining when to stop growing the tree.

5 Linear Classification

5.1 Logistic Regression

Logistic regression is a statistical and machine learning technique used for classifying records in a dataset based on the values of input fields.

It is commonly used for binary classification problems, predicting outcomes such as yes/no, true/false, or successful/not successful.

1. **Example Scenario:**
    - The scenario presented involves a telecommunication dataset where the goal is to predict customer churn (whether customers will leave) based on historical customer data.
    - Independent variables (features) like tenure, age, and income are used to predict the dependent variable (churn).
2. **Comparison with Linear Regression:**
    - Logistic regression is analogous to linear regression but is used for predicting categorical or discrete target fields instead of numeric ones.
    - It predicts binary outcomes and requires continuous independent variables; categorical variables should be transformed into continuous values.
3. **Applications of Logistic Regression:**
    - Logistic regression is used for various classification problems, including predicting:
        - Probability of a person having a heart attack.
        - Chance of mortality in an injured patient.
        - Likelihood of a patient having a specific disease (e.g., diabetes).
        - Probability of a customer purchasing a product or canceling a subscription.
        - Likelihood of failure in a process, system, or product.
        - Likelihood of a homeowner defaulting on a mortgage.
4. **When to Use Logistic Regression:**
    - Logistic regression is suitable in four main situations:
        - When the target field is categorical or binary.
        - When the probability of prediction is needed.
        - When the data is linearly separable, allowing for a clear decision boundary.
        - When understanding the impact of features is important, as logistic regression coefficients indicate the feature impact.
5. **Formalization of the Problem:**

- The dataset is represented as X in the space of real numbers (m dimensions by n records), and Y is the binary class to be predicted (0 or 1).
- The logistic regression model, denoted as Y hat, predicts the class of a sample and calculates the probability of a sample belonging to a particular class.

5.1.1 Logistic regression vs Linear regression

1. **Linear Regression Recap:**
    - Linear regression is demonstrated with an example of predicting income based on customer age.
    - It involves fitting a line through the data and using the equation *y=a+bx* to make predictions.
2. **Linear Regression for Binary Classification:**
    - The text explores the use of linear regression for predicting a categorical field like churn (yes/no).
    - A polynomial is fitted through the data, and a threshold (e.g., 0.5) is used to classify instances into categories.
    - The problem arises as linear regression produces continuous values, and a step function is needed to convert them into binary classes.
3. **Introduction of Sigmoid Function:**
    - The sigmoid function is introduced as a solution to the shortcomings of linear regression in classification.
    - The sigmoid function, also called the logistic function, maps any real-valued number to the range of 0 and 1.
4. **Role of Sigmoid in Logistic Regression:**
    - Logistic regression uses the sigmoid function to transform the linear combination of features and parameters into probabilities.
    - The sigmoid function outputs probabilities between 0 and 1, making it suitable for binary classification.
5. **Training Process for Logistic Regression:**
    - The training process involves initializing parameters (Theta) with random values.
    - The sigmoid of the linear combination *ϑTx* is used to calculate the model's output (probability).
    - The model's error is computed by comparing the predicted probability with the actual label for each instance.
    - The total error (cost) for the model is calculated using a cost function, representing the overall accuracy.
    - Gradient descent is employed to iteratively update Theta to minimize the cost.

- The training process continues until the cost is sufficiently low or a predefined number of iterations is reached.

6. **Challenges and Optimization:**
    - The challenges addressed include the need for probabilistic outputs and the limitations of linear regression for classification.
    - Gradient descent is highlighted as a popular method for updating parameters and minimizing the cost function.
7. **Stopping Iterations:**
    - the training process should stop when the model's accuracy is satisfactory.

5.1.2 Logistic regression Training

The main goal is to change the parameters of the logistic regression model to best estimate the labels of the samples in the dataset (e.g., customer churn).

1. **Cost Function Formulation:**
    - The cost function represents the difference between the actual labels (y) and the model's predictions (y hat).
    - The cost function is defined as the average sum of the squared differences between predicted and actual values.
2. **Introduction of Minus Log Function:**
    - To simplify the cost function and make it easier to find its minimum point, the minus log function is introduced.
    - The minus log function penalizes situations where the model output is far from the actual label, providing a suitable cost function for logistic regression.
3. **Logistic Regression Cost Function:**
    - The logistic regression cost function is derived using the minus log function.
    - It penalizes cases where the predicted probability is close to the actual label (0 or 1) and encourages the model to converge to the optimal parameters.
4. **Optimization Approach:**
    - The objective is to minimize the cost function by finding the global minimum.
    - An optimization approach is needed to update parameters iteratively and reach the minimum point.
5. **Gradient Descent:**
    - Gradient descent is introduced as an iterative approach to finding the minimum of a function.
    - In logistic regression, it involves changing the parameter values (weights) based on the derivative of the cost function.

- The derivative of the cost function represents the slope of the surface at a given point, and moving in the opposite direction of the slope minimizes the cost.
- The learning rate is introduced to control the size of the steps taken in the parameter space during each iteration.

6. **Training Algorithm Steps:**
    - Initialization: Parameters are initialized with random values.
    - Cost Calculation: The cost function is calculated for the training set, expecting a high error initially.
    - Gradient Calculation: The gradient of the cost function is computed using partial derivatives.
    - Parameter Update: The parameters are updated based on the gradient and learning rate.
    - Iteration: Steps 2-4 are repeated until the cost reaches an acceptable minimum or a set number of iterations is reached.
    - Model Readiness: The trained parameters are used for predicting probabilities (e.g., customer staying or leaving).
7. **Conclusion:**
    - The iterative nature of the algorithm ensures that the parameters are updated to minimize the cost, resulting in an optimized logistic regression model.

5.2 SVM – Support Vector Machine

SVM is a machine learning method used for classification.

1. **Formal Definition of SVM:**
    - SVM is a supervised algorithm that classifies cases by finding a separator.
    - It works by mapping data to a high-dimensional feature space, even when the data are not linearly separable.
    - The algorithm estimates a separator, typically represented as a hyperplane, in the transformed space.
2. **Data Transformation and Kernels:**
    - Data points that are not linearly separable can be transformed into higher-dimensional spaces using a kernel function.
    - The kernel function can be linear, polynomial, Radial Basis Function (RBF), or sigmoid.
    - Kernelling is the process of mapping data into a higher-dimensional space to achieve linear separability.
3. **Choosing the Right Separator:**
    - SVM aims to find a hyperplane that best divides a dataset into two classes.
    - The best hyperplane is the one with the largest separation or margin between the classes.

  - Support vectors, which are examples closest to the hyperplane, play a crucial role in determining the optimal hyperplane.

4. **Optimization and Gradient Descent:**
    - Learning the hyperplane involves an optimization procedure that maximizes the margin.
    - The optimization problem can be solved using gradient descent.
    - The output of the algorithm includes values w and b for the line, which can be used for classifications.

5. **Advantages and Disadvantages of SVMs:**
    - Advantages: Accuracy in high-dimensional spaces and memory efficiency (uses a subset of training points).
    - Disadvantages: Prone to overfitting with a high number of features, does not provide direct probability estimates, and computationally inefficient for large datasets.
6. **Applications of SVM:**
    - SVM is suitable for image analysis tasks (classification, digit recognition), text mining (spam detection, sentiment analysis), gene expression data classification, and other machine learning problems like regression, outlier detection, and clustering.

5.3 MULTICLASS PREDICTION

SoftMax Regression:

- **Purpose:** Used for multi-class classification, extending logistic regression.
- **Function:** Converts distances (dot products) of input features with each class's parameters into probabilities.
- **SoftMax Function:** $softmax(x,i) = \frac{k^{-\theta_i^T x}}{\sum_{j=1}^{k} e-\theta_i^T x}$
- **Training Procedure:** Similar to logistic regression using cross-entropy.
- **Prediction:** Use the argmaxargmax function to select the class with the highest probability.

One-vs-All (One-vs-Rest):

- **Approach:** Create *K* two-class classifier models for *K* classes.
- **Dummy Class:** Introduce a "dummy" class for each classifier.
- **Training:** Train each classifier to distinguish one class from the rest.
- **Classification:** Use majority vote or select the classifier with the highest probability, ignoring dummy classes.
- **Ambiguous Regions:** May lead to ambiguous regions where multiple classes are predicted.

One-vs-One:

- **Approach:** Train $K(K-1)/2$ classifiers, each distinguishing between two specific classes.
- **Number of Classifiers:** For K classes, $K(K-1)/2$ classifiers are trained.
- **Classification:** Perform a majority vote among all classifiers.
- **Ambiguous Regions:** Smaller compared to One-vs-All, but still present.

Geometric Interpretation:

- **SoftMax Regression:** Hyperplanes represent class boundaries.
- **One-vs-All:** Each class is assigned a region, and a majority vote is used.
- **One-vs-One:** Multiple hyperplanes for each pair of classes, and majority vote determines the final class.

Issues:

- **One-vs-All:** Ambiguous regions may lead to multiple class predictions.
- **Solutions:** Fusion rules, using class probabilities to reduce ambiguity.

Conclusion:

- **SoftMax Regression:** Suitable for multi-class classification, converting distances to probabilities.
- **One-vs-All:** Builds multiple classifiers, one for each class.
- **One-vs-One:** Trains classifiers for all class pairs, more classifiers but potentially smaller ambiguous regions.

Note:

- **Probability Usage:** Probability scores can be used for decision-making, helping reduce ambiguity.
- **Package Note:** Some packages like Scikit-learn can output probabilities for SVM.

6 CLUSTERING

Identifying clusters in a dataset where data points in a cluster are similar to each other.

Cluster vs. Classification: Clustering is unsupervised, classification is supervised.

- **Examples of Clustering:**
    - Retail: Identify buying patterns and associations among customers.
    - Recommendation Systems: Group similar items or users for collaborative filtering.
    - Banking: Detect patterns of fraudulent credit card usage, identify customer clusters.
    - Insurance: Fraud detection in claims analysis, evaluate insurance risk based on segments.
    - Media: Auto categorize and tag news articles, recommend similar articles.
    - Medicine: Characterize patient behavior for identifying successful therapies.
    - Biology: Group genes with similar expression patterns, cluster genetic markers for family ties.

Purposes of Clustering:

- **Exploratory Data Analysis:** Understand the structure of data.
- **Summary Generation:** Summarize and interpret data.
- **Outlier Detection:** Identify anomalies, useful for fraud detection or noise removal.
- **Finding Duplicates:** Identify and handle duplicate records.
- **Pre-processing:** Used as a step before prediction, data mining, or in complex systems.

Clustering Algorithms and Characteristics:

- **Partition-based Clustering:** Produces sphere-like clusters (e.g., K-Means, K-Medians, Fuzzy c-Means). Efficient for medium and large databases.
- **Hierarchical Clustering:** Produces trees of clusters (e.g., agglomerative, divisive). Intuitive and suitable for small datasets.
- **Density-based Clustering:** Produces arbitrary shaped clusters (e.g., DB scan). Suitable for spatial clusters or datasets with noise.

6.1 K-MEANS

Introduction to K-Means Clustering:

- **Scenario:** Customer dataset for customer segmentation.
- **Customer Segmentation:** Partitioning customer base into groups with similar characteristics.
- **Algorithm:** K-Means clustering is introduced as an unsupervised algorithm for this purpose.

Defining K-Means Clustering:

- **Types of Clustering Algorithms:** Partitioning, hierarchical, density-based.

- **K-Means:** A type of partitioning clustering, dividing data into K non-overlapping clusters.
- **Objective:** Minimize intra-cluster distances, maximize inter-cluster distances.
- **Similarity Metric:** Dissimilarity metrics, commonly using distance (e.g., Euclidean distance).

K-Means Clustering Process:

- **Dataset Features:** Consider a dataset with features like age and income.
- **Initialization:** Randomly initialize K centroids, representing cluster centers.
- **Assignment:** Assign each data point to the closest centroid based on distance.
- **Error Calculation:** Calculate the error as the total distance of each point from its centroid.
- **Centroid Update:** Move centroids to the mean of data points in their cluster.
- **Iteration:** Repeat steps until centroids no longer move (convergence).
- **Iterative Nature:** K-Means is an iterative algorithm, requiring multiple repetitions.

Handling Initial Centroids:

- **Heuristic Algorithm:** No guarantee of converging to the global optimum.
- **Random Initialization:** Run the process multiple times with different starting conditions.

Evaluating K-Means Clustering:

- **Goodness Evaluation:** Assessing the quality of clusters formed.
- **Ground Truth Comparison:** Compare clusters with ground truth if available (unsupervised).
- **Error Metrics:** Average distance between data points within a cluster or from their centroids.
- **Determining K:** Choosing the right number of clusters (K) is challenging.
- **Elbow Method:** Plotting the error metric for different K values, identifying the elbow point.

Recap of K-Means Clustering:

- **Efficiency:** Relatively efficient for medium and large datasets.
- **Cluster Shape:** Produces sphere-like clusters.
- **Drawback:** Requires pre-specifying the number of clusters (challenging task).