UNIT 1

INTRODUCTION

Predictive analysis, also known as predictive analytics, is a branch of data analytics that involves the use of data, statistical algorithms, machine learning, and modeling techniques to make predictions about future events or outcomes. It aims to identify patterns, trends, and relationships in historical data and use this information to forecast future events or behaviours.

KEY ASPECTS OF PREDICTIVE ANALYSIS

- Data Collection: The process starts with collecting relevant data from various sources. This data can be structured (e.g., databases, spreadsheets) or unstructured (e.g., text, images), and it often includes historical information about the subject of interest.
- Data Preprocessing: Once data is collected, it needs to be cleaned and preprocessed. This involves handling missing values, outliers, and ensuring data quality. Data preprocessing also includes feature selection and engineering, where relevant variables are chosen or created to improve the predictive model's performance.
- Model Building: Predictive analysis typically involves building mathematical models or using machine learning algorithms. These models are trained on historical data, learning the relationships between input features (variables) and the target variable (the variable to be predicted). Common modeling techniques include linear regression, decision trees, random forests, support vector machines, and neural networks.
- Model Evaluation: After training the model, it's essential to assess its performance to ensure it can make accurate predictions. Common evaluation metrics include accuracy, precision, recall, F1 score, and mean squared error, among others.
- Deployment: Once a predictive model is deemed satisfactory, it can be deployed in real-world applications. This often involves integrating the model into business processes or software systems to make automated predictions.
- Continuous Improvement: Predictive models require ongoing monitoring and maintenance. As new data becomes available, models may need to be retrained to stay accurate. Feedback loops are crucial for keeping models up-to-date and relevant.

APPLICATIONS

Predictive analysis is widely used across various industries for a range of applications, including:

- Finance: Predicting stock prices, credit risk assessment, and fraud detection.
- Healthcare: Predicting disease outbreaks, patient readmission rates, and treatment outcomes.
- Marketing: Customer segmentation, churn prediction, and recommendation systems.
- Manufacturing: Predictive maintenance for machinery, quality control, and supply chain optimization.
- Retail: Inventory management, demand forecasting, and price optimization.
- Sports: Predicting game outcomes, player performance, and injury risk.

OVERVIEW OF SUPERVISED LEARNING

Supervised learning is a fundamental concept in machine learning, a subfield of artificial intelligence. It is one of the most common and widely used techniques for training machine learning models. In supervised learning, a machine learning algorithm learns to map input data to output labels based on a labeled dataset, where the correct answers (labels) are provided during the training process. The goal is for the algorithm to learn a mapping function that can make accurate predictions or classifications on new, unseen data.

For each there is a set of variables that might be denoted as inputs, which are measured or preset. These have some influence on one or more outputs. For each example the goal is to use the inputs to predict the values of the outputs. This exercise is called supervised learning. We have used the more modern language of machine learning. In the statistical literature the inputs are often called the predictors, a term we will use interchangeably with inputs, and more classically the independent variables. In the pattern recognition literature the term

features is preferred, which we use as well. The outputs are called the responses, or classically the dependent variables

Variable Types and Terminology

- Input Features (Independent Variables):

Input features, also known as independent variables or predictors, are the variables that are used as input to the machine learning model. These features are used to make predictions or classifications.

Inputs also vary in measurement type; we can have some of each of **qualitative and quantitative input variables.** These have also led to distinctions in the types of methods that are used for prediction: some methods are defined most naturally for quantitative inputs, some most naturally for qualitative and some for both.

A third variable type is **ordered categorical**, such as small, medium and large, where there is an ordering between the values, but no metric notion is appropriate (the difference between medium and small need not be the same as that between large and medium). These are discussed further in

- Output Labels (Dependent Variables):

Output labels, also known as dependent variables or target variables, are the variables that the model aims to predict or classify. The model's goal is to learn a mapping from input features to output labels.

The output is a quantitative measurement, where some measurements are bigger than others, and measurements close in value are close in nature.

There is no explicit ordering in the classes, and in fact often descriptive labels rather than numbers are used to denote the classes. Qualitative variables are also referred to as categorical or discrete variables as well as factors. For both types of outputs it makes sense to think of using the inputs to predict the output. Given some specific atmospheric measurements today and yesterday, we want to predict the ozone level tomorrow. Given the grayscale values for the pixels of the digitized image of the handwritten digit, we want to predict its class label.

- Training Data:

Training data is the portion of the dataset used to train the machine learning model. It consists of a set of input-feature-output-label pairs that the model uses to learn the underlying patterns and relationships.

- Test Data:

Test data, also called validation data or evaluation data, is a separate portion of the dataset that is not used during training. It is used to evaluate the model's performance and assess its ability to make accurate predictions on unseen data.

- Supervised Learning Algorithm:

A supervised learning algorithm is a specific technique or method used to learn the mapping from input features to output labels. Common algorithms include linear regression, decision trees, support vector machines, neural networks, and more, depending on the nature of the problem.

- Model:

The model represents the learned relationship between input features and output labels. It's essentially a mathematical function or a set of rules that can make predictions or classifications.

- Training Phase:

The training phase is the process during which the model is fitted to the training data by adjusting its internal parameters. This phase involves optimization techniques to minimize the difference between predicted and actual labels.

- Prediction/Inference Phase:

After training, the model can be used to make predictions or classifications on new, unseen data. This is often referred to as the prediction or inference phase.

- Loss/Cost Function:

The loss or cost function measures the error between the model's predictions and the actual labels in the training data. The goal during training is to minimize this function. ● Hyperparameters:

Hyperparameters are configuration settings for the machine learning algorithm that are not learned from the data but are set prior to training. Examples include learning rates, regularization strengths, and the choice of algorithms.

- Overfitting:

Overfitting occurs when a model learns the training data too well but fails to generalize to new, unseen data. It happens when the model captures noise in the training data rather than the underlying patterns. ● Underfitting:

Underfitting occurs when a model is too simple to capture the underlying patterns in the data. It results in poor performance on both the training and test datasets. ● Bias and Variance:

Bias refers to the error due to overly simplistic assumptions in the learning algorithm, which can lead to underfitting. Variance refers to the error due to too much complexity in the

algorithm, leading to overfitting. Striking the right balance between bias and variance is essential for model performance.

REPRESENTATION

Qualitative variables are typically represented numerically by codes. The easiest case is when there are only two classes or categories, such as “success” or “failure,” “survived” or “died.” These are often represented by a single binary digit or bit as 0 or 1, or else by −1 and 1. For reasons that will become apparent, such numeric codes are sometimes referred to as targets. When there are more than two categories, several alternatives are available. The most useful and commonly used coding is via dummy variables. Here a K-level qualitative variable is represented by a vector of K binary variables or bits, only one of which is “on” at a time. Although more compact coding schemes are possible, dummy variables are symmetric in the levels of the factor.

We will typically denote an input variable by the symbol X. If X is a vector, its components can be accessed by subscripts Xj . Quantitative outputs will be denoted by Y , and qualitative outputs by G (for group). We use uppercase letters such as X, Y or G when referring to the generic aspects of a variable. Observed values are written in lowercase; hence the ith observed value of X is written as xi (where xi is again a scalar or vector). Matrices are represented by bold uppercase letters; for example, a set of N input p-vectors xi , i = 1,... ,N would be represented by the N ×p matrix X. In general, vectors will not be bold, except when they have N components; this convention distinguishes a p-vector of inputs xi for the ith observation from the N-vector xj consisting of all the observations on variable Xj . Since all vectors are assumed to be column vectors, the ith row of X is x T i , the vector transpose of xi . For the moment we can loosely state the learning task as follows: given the value of an input vector X, make a good prediction of the output Y, denoted by Yˆ (pronounced “y-hat”). If Y takes values in IR then so should Yˆ ; likewise for categorical outputs, Gˆ should take values in the same set G associated with G. For a two-class G, one approach is to denote the binary coded target as Y , and then treat it as a quantitative output. The predictions Yˆ will typically lie in [0, 1], and we can assign to Gˆ the class label according to whether ˆy > 0.5. This approach generalizes to K-level qualitative outputs as well. We need data to construct prediction rules, often a lot of it. We thus suppose we have available a set of measurements (xi ,yi) or (xi ,gi), i = 1,... ,N, known as the training data, with which to construct our prediction rule

These terms and concepts are foundational to understanding and working with supervised learning algorithms. They provide the framework for developing, training, and evaluating machine learning models for various tasks, from regression to classification and beyond.

REGRESSION AND CLASSIFICATION

This distinction in output type has led to a naming convention for the prediction tasks: **regression** when we predict quantitative outputs, and **classification** when we predict qualitative outputs. We will see that these two tasks have a lot in common, and in particular both can be viewed as a task in function approximation.

Classification and regression are two fundamental types of supervised learning problems in machine learning, and they differ in terms of their goals and the types of output they produce:

Classification:

- Goal: In classification, the objective is to assign input data points to one of several predefined categories or classes. The goal is to learn a mapping from input features to discrete class labels.
- Output: The output of a classification algorithm is categorical or nominal. It predicts which class or category an input belongs to. Examples include spam vs. non-spam email, disease vs. no disease diagnosis, or image classification (e.g., cat vs. dog).
- Examples of Algorithms: Some common classification algorithms include logistic regression, decision trees, random forests, support vector machines, and neural networks (for multi-class classification with softmax activation).
- Evaluation Metrics: Classification models are evaluated using metrics such as accuracy, precision, recall, F1-score, confusion matrix, ROC curve, and AUC-ROC.

Regression:

- Goal: In regression, the goal is to predict a continuous or numerical output based on input features. The objective is to learn a function that can map input data to a real-valued output.
- Output: The output of a regression algorithm is a continuous value. Examples include predicting house prices, stock prices, temperature, or a person's age based on various features.
- Examples of Algorithms: Some common regression algorithms include linear regression, polynomial regression, ridge regression, and support vector regression. Additionally, regression can be tackled using neural networks with appropriate output layers (e.g., a single neuron with linear activation for simple regression).

- Evaluation Metrics: Regression models are evaluated using metrics such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), R-squared (coefficient of determination), and others that measure the accuracy of the continuous predictions.

To summarize, classification is concerned with predicting discrete class labels, while regression is focused on predicting continuous numerical values. The choice between these two types of supervised learning tasks depends on the nature of the problem and the type of output you want to produce. Classification is suitable when the outcome is categorical, such as yes/no or multiple classes, while regression is appropriate when the outcome is numeric and continuous.

REGRESSION ANALYSIS

Regression Analysis is a statistical process for estimating the relationships between the dependent variables or criterion variables and one or more independent variables or predictors. Regression analysis is generally used when we deal with a dataset that has the target variable in the form of continuous data. Regression analysis explains the changes in criteria in relation to changes in select predictors. The conditional expectation of the criteria is based on predictors where the average value of the dependent variables is given when the independent variables are changed. Three major uses for regression analysis are determining the strength of predictors, forecasting an effect, and trend forecasting.

There are times when we would like to analyze the effect of different independent features on the target or what we say dependent features. This helps us make decisions that can affect the target variable in the desired direction. Regression analysis is heavily based on statistics and hence gives quite reliable results to this reason only regression models are used to find the linear as well as non-linear relation between the independent and the dependent or target variables.

TYPES OF REGRESSION MODELS

Linear Regression
Polynomial Regression
Stepwise Regression

Decision Tree Regression
Random Forest Regression
Support Vector Regression
Ridge Regression
Lasso Regression
ElasticNet Regression
Bayesian Linear Regression

LINEAR REGRESSION MODELS

A linear regression model assumes that the regression function E(Y |X) is linear in the inputs X1,... ,Xp. Linear models were largely developed in the precomputer age of statistics, but even in today's computer era there are still good reasons to study and use them. They are simple and often provide an adequate and interpretable description of how the inputs affect the output. For prediction purposes they can sometimes outperform fancier nonlinear models, especially in situations with small numbers of training cases, low signal-to-noise ratio or sparse data. Finally, linear methods can be applied to transformations of the inputs and this considerably expands their scope.

Linear Regression Models and Least Squares

we have an input vector **XT = (X1,X2,... ,Xp)**, and want to predict a real-valued output **Y** . The linear regression model has the form

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j.$$

The linear model either assumes that the regression function **E(Y |X)** is linear, or that the linear model is a reasonable approximation.

Here the βj 's are unknown parameters or coefficients, and the variables Xj can come from different sources:

- quantitative inputs;
- transformations of quantitative inputs, such as log, square-root or square;
- basis expansions, such as X2 = X2 1 , X3 = X3 1 , leading to a polynomial representation; • numeric or "dummy" coding of the levels of qualitative inputs. For example, if G is a five-level factor input, we might create Xj , j = 1,... , 5, such that Xj = I(G

= j). Together this group of Xj represents the effect of P G by a set of level-dependent constants, since in 5 j=1 Xjβj , one of the Xj s is one, and the others are zero.

- interactions between variables, for example, X3 = X1 · X2.

No matter the source of the Xj , the model is linear in the parameters. Typically we have a set of training data (x1,y1)...(xN ,yN) from which to estimate the parameters β. Each xi = (xi1,xi2,... ,xip) T is a vector of feature measurements for the ith case.

The **most popular estimation method is least square**s, in which we **pick the coefficients β = (β0,β1,... ,βp) T to minimize the residual sum of squares**

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{N}(y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N}\Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\Big)^2. \end{aligned}$$

From a statistical point of view, this criterion is reasonable if the training observations (xi ,yi) represent independent random draws from their population.

Even if the xi ’s were not drawn randomly, the criterion is still valid if the yi ’s are conditionally independent given the inputs xi .

Least squares method in detail

Least squares linear regression is a statistical method used for modeling the relationship between a dependent variable (also called the response variable) and one or more independent variables (also called predictor variables or features) by fitting a linear equation to observed data.

The goal is to find the best-fitting linear equation that minimizes the sum of the squared differences between the observed and predicted values.

This method is widely used for tasks like predicting values, estimating relationships between variables, and understanding the strength and direction of those relationships.

Here's a basic overview of how least squares linear regression works:

- Model Representation: The linear regression model assumes that the relationship between the dependent variable (Y) and the independent variables (X) can be represented as:

$$\mathbf{Y = \beta 0 + \beta 1X1 + \beta 2X2 + ... + \beta n*Xn + \varepsilon}$$

Y is the dependent variable.
X1, X2, ..., Xn are the independent variables. β0 is the intercept (constant) term. β1, β2, ..., βn are the coefficients associated with each independent variable.
ε represents the error term, which accounts for the variability in Y that is not explained by the model.

- Fitting the Model: The goal is to estimate the coefficients (β0, β1, β2, ..., βn) that minimize the sum of the squared differences between the observed Y values and the predicted Y values from the model. This is often done using an optimization algorithm.
- Least Squares Criterion: The method minimizes the sum of squared residuals, which is the sum of the squares of the differences between the observed Y values and the predicted Y values:

$$\mathbf{\Sigma(yi - \hat{y}i)^2}$$

where yi is the observed value of the dependent variable, ŷi is the predicted value, and the summation is taken over all data points.

- Estimating Coefficients: The estimated coefficients (β0, β1, β2, ..., βn) are found such that they minimize the least squares criterion. There are closed-form solutions for finding these coefficients, but iterative optimization algorithms like gradient descent are often used in practice.
- Assessing the Model: Once you have the estimated coefficients, you can assess the quality of the model by examining measures like the coefficient of determination (R-squared), which tells you how well the model explains the variance in the dependent variable.
- Making Predictions: You can use the fitted model to make predictions on new, unseen data by plugging in values for the independent variables.

Example

LS EXAMPLE

Multiple Regression

Linear regression using multiple inputs, often referred to as multiple linear regression, is a statistical method used to model the relationship between a dependent variable (also called the target or output variable) and two or more independent variables (also called predictors or features). It is an extension of simple linear regression, which deals with only one independent variable.

The linear model with p > 1 inputs is called the multiple linear regression model. The least squares estimates for this model are best understood in terms of the estimates for the univariate (p = 1) linear model

The general form of multiple linear regression can be represented by the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$$

Where:

- Y is the dependent variable you want to predict.
- β_0 is the intercept (the value of Y when all X variables are 0).
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for the independent variables $X_1, X_2, \ldots, X_n$, respectively. These coefficients represent the change in Y for a one-unit change in the corresponding X variable, while holding all other variables constant.
- $X_1, X_2, \ldots, X_n$ are the independent variables (features).
- ϵ represents the error term, which accounts for the variability in Y that is not explained by the linear relationship with the X variables.

EXAMPLE

multiple regression example

Multiple Outputs

Linear regression with multiple outputs is a variation of traditional linear regression that is used when you want to predict multiple dependent variables simultaneously, rather than just a single dependent variable. This technique is also known as multivariate linear regression.

In standard linear regression, you have a single dependent variable (Y) and one or more independent variables (X) and aim to find a linear relationship that best describes the relationship between them. The general equation for simple linear regression is:

$Y = \beta 0 + \beta 1X + \varepsilon$

Where:

Y is the dependent variable. X is the independent variable. β0 is the intercept. β1 is the coefficient of the independent variable. ε represents the error term.

In multiple output linear regression, you have multiple dependent variables, denoted as Y1, Y2, Y3, ..., Yn. The goal is to find a set of coefficients and an intercept for each dependent variable to predict them using one or more independent variables, which can be shared among the dependent variables. The general equation for multiple output linear regression is:

$Y1 = \beta 10 + \beta 11X1 + \beta 12X2 + ... + \beta 1mXm + \varepsilon 1$
$Y2 = \beta 20 + \beta 21X1 + \beta 22X2 + ... + \beta 2mXm + \varepsilon 2$
...
$Yn = \beta n0 + \beta n1X1 + \beta n2X2 + ... + \beta nmXm + \varepsilon n$

Where:

Y1, Y2, ..., Yn are the multiple dependent variables.
X1, X2, ..., Xm are the independent variables.
β10, β11, β12, ..., β1m, β20, β21, β22, ..., β2m, ..., βn0, βn1, βn2, ..., βnm are the intercepts and coefficients for each dependent variable. ε1, ε2, ..., εn represent the error terms for each dependent variable.

The goal of multiple output linear regression is to find the values of β coefficients that minimize the sum of the squared errors across all the dependent variables. This can be done using various optimization techniques, such as ordinary least squares (OLS) or gradient descent.

Multiple output linear regression is commonly used in various fields, including economics, finance, and engineering, when you want to model and predict multiple related variables simultaneously. It's an extension of simple linear regression that allows for a more comprehensive analysis of multivariate relationships.

Challenges of Least squares method

There are two reasons why we are often not satisfied with the least squares estimates .

- The first is prediction accuracy: the least squares estimates often have low bias but large variance. Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.

- The second reason is interpretation. With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture," we are willing to sacrifice some of the small details.

SUBSET SELECTION

With subset selection we retain only a subset of the variables, and eliminate the rest from the model. Least squares regression is used to estimate the coefficients of the inputs that are retained. There are a number of different strategies for choosing the subset.

Best Subset Selection

Best Subset Selection is a technique used in statistical modeling and regression analysis to select the best combination of predictor variables from a larger set of potential predictors. Its goal is to find the subset of predictors that results in the best-performing model based on a specific criterion, such as minimizing the residual sum of squares (RSS), maximizing the coefficient of determination (R-squared), or using a different metric depending on the problem.

Here are the general steps involved in Best Subset Selection:

Generate Subsets: Create all possible subsets of predictor variables from the full set of potential predictors. For a set of 'p' predictors, this results in 2^p possible subsets.
Fit Models: Fit a regression model for each subset of predictors. This typically involves performing multiple linear regressions (or another regression method) for each subset.
Evaluate Models: Evaluate the performance of each model using a chosen criterion or metric. Common metrics include RSS, R-squared, or adjusted R-squared.
Select the Best Model: Choose the model that performs the best according to the chosen criterion. This model will be the one with the smallest RSS or the highest R-squared, for example.
Validate the Model: After selecting the best subset, it's crucial to validate the model's performance using a separate dataset or cross-validation to ensure it generalizes well to new data.

Forward Backward Stepwise Selection

Forward and backward stepwise selection are two commonly used methods for variable selection in statistical modeling and regression analysis. These methods are often used when dealing with a large number of predictor variables to choose a subset of predictors that are most relevant to the model. Both methods are iterative and seek to strike a balance between model complexity and predictive performance.

Forward Stepwise Selection:

Forward stepwise selection starts with an empty model and adds predictors one at a time based on a specific criterion (usually statistical significance or a decrease in some information criterion like AIC or BIC).

The steps involved in forward stepwise selection are as follows:

Start with an empty model.
Fit simple linear regression models for each predictor individually and choose the one that provides the best fit based on the chosen criterion (e.g., lowest p-value).
Add this predictor to the model.
Repeatedly fit models by adding one predictor at a time, choosing the predictor that improves the model fit the most until a stopping criterion is met (e.g., a predetermined number of predictors or no more predictors meet the significance threshold). The final model consists of the selected predictors.

Forward stepwise selection is generally less computationally intensive than best subset selection and is more efficient for large sets of potential predictors.

Backward Stepwise Selection:

Backward stepwise selection starts with the full model (all predictors) and iteratively removes predictors based on a specific criterion.

The steps involved in backward stepwise selection are as follows:

Start with the full model containing all predictors.
Fit the full model and assess the significance or usefulness of each predictor, typically through p-values.
Remove the predictor that is the least significant (highest p-value).
Repeatedly fit models by removing one predictor at a time until a stopping criterion is met (e.g., a predetermined significance threshold for removal).
The final model consists of the selected predictors.
Backward stepwise selection can be computationally more intensive than forward selection, especially when there are many predictors.

Both forward and backward stepwise selection have their advantages and disadvantages. The choice between them often depends on the specific problem, the number of predictors, and the computational resources available. Additionally, it's essential to validate the selected models and assess their generalization performance on independent data to ensure they provide accurate predictions. Stepwise methods can be sensitive to the order in which predictors are considered, and their performance can vary depending on the dataset and the criteria used for inclusion or exclusion.

Forward Stagewise Regression

Forward stage-wise regression, also known as forward stagewise regression, is a variable selection technique used in linear regression and related modeling methods. It is similar to forward stepwise regression, but it differs in how it updates the coefficients of the selected predictors. Forward stage-wise regression is often used when there are many potential predictor variables and you want to select a subset of the most relevant predictors for your model.

Here are the key characteristics and steps of forward stage-wise regression:

Initialization: Start with a model that includes only the intercept (no predictors).

Iteration:
In each iteration, select the predictor that has the highest correlation with the current model's residuals.
Update the coefficient of the selected predictor by taking a small step (learning rate) in the direction that minimizes the residual sum of squares (RSS).
Repeat the above steps until a predefined stopping criterion is met, such as a maximum number of iterations or when the improvement in RSS becomes very small.

Stopping Criterion: The algorithm stops when it reaches the desired number of iterations or when the change in RSS (or another chosen criterion) is below a certain threshold.

Output: The final model consists of the selected predictors and their corresponding coefficients.

RIDGE REGRESSION

Ridge regression shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum

Ridge regression extends linear regression by adding a regularization term to the OLS cost function. The cost function for Ridge regression is:

$$\text{Cost}(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{p} \beta_j^2$$

Where:

- β represents the coefficients (weights) we want to find.
- n is the number of data points.
- y_i is the actual target value for the i-th data point.
- $\hat{y}_i$ is the predicted target value for the i-th data point using the linear regression model.

Key Points about Ridge Regression:

1. **Regularization Term**: The second term in the Ridge cost function, $\alpha \sum_{j=1}^{p} \beta_j^2$, is the regularization term. It penalizes large values of the coefficients. As a result, Ridge regression encourages the coefficients to be small, which helps prevent overfitting.
2. **Bias-Variance Tradeoff**: Ridge regression introduces a bias in the model (by shrinking coefficients), but it reduces the variance of the model. This tradeoff is useful when dealing with multicollinearity, where features are highly correlated. Ridge regression can handle multicollinearity better than simple linear regression.
3. **Choosing the Alpha Parameter**: The choice of the α parameter is crucial. A small α makes Ridge regression similar to OLS regression, while a large α increases the strength of regularization. Cross-validation is often used to tune the α parameter to find the best value for a specific dataset.
4. **Closed-Form Solution**: Ridge regression has a closed-form solution, which means you can directly compute the optimal coefficients using a mathematical formula. This is in contrast to some other regularization techniques like Lasso regression, which use optimization algorithms.

LASSO REGRESSION

Lasso regression, short for "Least Absolute Shrinkage and Selection Operator" regression, is a linear regression technique used for feature selection and regularization in statistical modeling and machine learning. It is particularly useful when dealing with datasets that have a large number of features, as it helps prevent overfitting and simplifies the model by automatically selecting a subset of the most important features.

Objective: Lasso regression aims to minimize the sum of squared residuals, like ordinary least squares (OLS) linear regression, but it adds a penalty term to the linear regression equation. The penalty term is the absolute sum of the coefficients of the features, multiplied by a regularization parameter (λ or alpha).

Regularization: Lasso introduces L1 regularization, which penalizes the absolute values of the regression coefficients. This regularization term encourages some coefficients to become exactly zero, effectively eliminating some features from the model. This feature selection property makes Lasso useful for automatic feature selection.

3. **Mathematical Formulation:** The objective function of Lasso regression is typically expressed as:
   Minimize: $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |w_j|$
   Here:
   - y_i is the observed value of the dependent variable for the i-th observation.
   - $\hat{y}_i$ is the predicted value of the dependent variable for the i-th observation.
   - w_j is the coefficient of the j-th feature.
   - p is the total number of features.
   - λ is the regularization parameter that controls the strength of the penalty. A higher value of λ results in stronger regularization and more feature shrinkage or elimination.

Feature Selection: Lasso regression can automatically select a subset of the most relevant features by driving the coefficients of irrelevant or less important features to zero. This can be particularly valuable when dealing with high-dimensional datasets where feature selection is crucial.

Relation to Ridge Regression: Lasso regression is closely related to Ridge regression, another regularization technique. The main difference is in the type of regularization they apply. Lasso uses L1 regularization, which tends to produce sparse models (with some coefficients being exactly zero), while Ridge uses L2 regularization, which shrinks the coefficients towards zero but doesn't force them to be exactly zero.

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a statistical and machine learning technique used for dimensionality reduction and classification. It is primarily employed in the field of pattern recognition and machine learning for tasks like face recognition, image classification, and data compression. LDA differs from other dimensionality reduction techniques, such as Principal Component Analysis (PCA), because it takes into account the class labels of the data points.

LDA is a supervised technique, meaning it relies on class labels during training, and its primary goal is to maximize class separability while minimizing the variance within classes.

- Data Preprocessing: LDA begins with a labeled dataset, where each data point is associated with a class label. This dataset is used for both dimensionality reduction and classification tasks.

- Calculate Class Means: For each class in the dataset, LDA calculates the mean (average) of the feature vectors belonging to that class. This results in as many class means as there are classes in the data.

- Calculate Scatter Matrices: LDA then computes two scatter matrices:

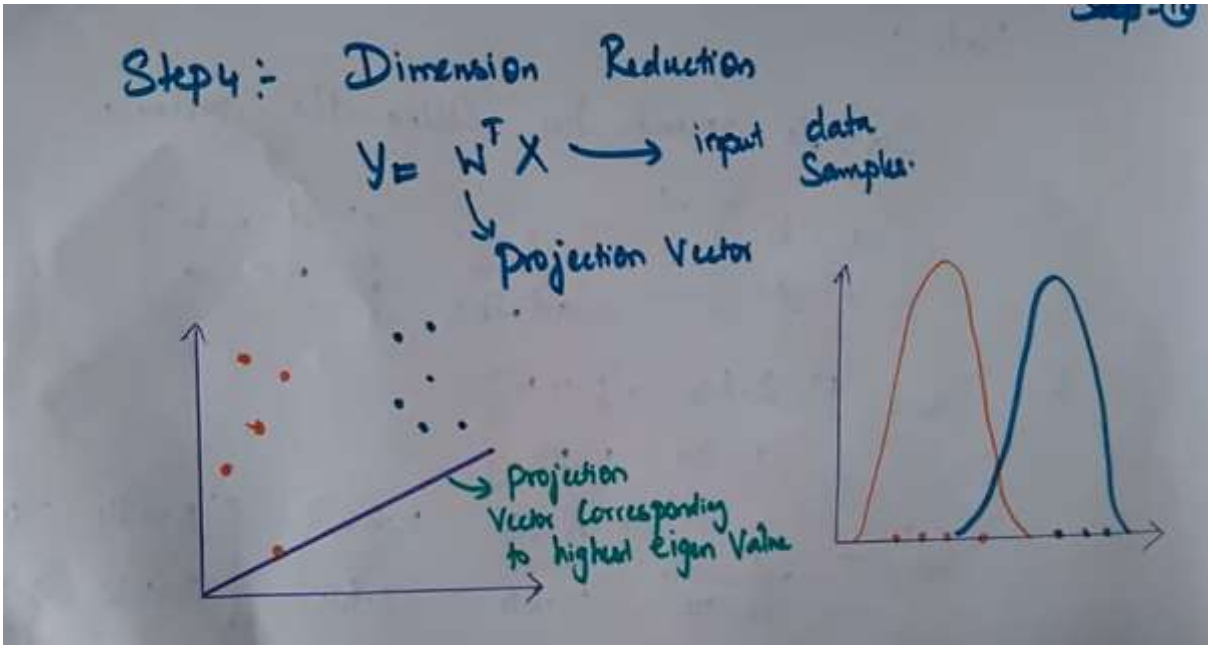
- Within-class scatter matrix (Sw): This measures the variance of data points within each class. It is calculated by summing up the covariance matrices of individual classes.
- Between-class scatter matrix (Sb): This measures the variance between class means. It is calculated by finding the covariance between the class means and then scaling it by the number of data points in each class.
- Eigenvalue Decomposition: The next step involves calculating the eigenvectors and eigenvalues of the matrix Sw^-1 * Sb. These eigenvectors represent the directions in the feature space along which the classes are best separated.

- Selecting Discriminant Vectors: The eigenvectors with the highest eigenvalues are selected as the discriminant vectors. These vectors capture the most important information for class discrimination.

- Projecting Data: To reduce the dimensionality of the data, you can project the original data onto the discriminant vectors. The number of discriminant vectors chosen typically depends on the desired dimensionality reduction.

- Classification: LDA can also be used for classification tasks. After reducing the dimensionality of the data using the discriminant vectors, you can apply a classifier (e.g., linear discriminant analysis, logistic regression) to classify new data points.

Key benefits of LDA include its ability to preserve class-specific information, making it particularly useful when you have labeled data and want to reduce dimensionality while maintaining discrimination power. It's worth noting that LDA assumes that the data follows a Gaussian distribution with equal covariance matrices for each class, which might not always hold in practice.



EXAMPLE

LDA EXAMPLE

LOGISTIC REGRESSION

Logistic regression is a statistical method used for binary classification, which means it is used to predict the probability that an input belongs to one of two possible classes (usually denoted as 0 and 1).

Despite its name, logistic regression is a classification algorithm, not a regression algorithm. It is widely used in various fields, including machine learning, medical research, economics, and social sciences.

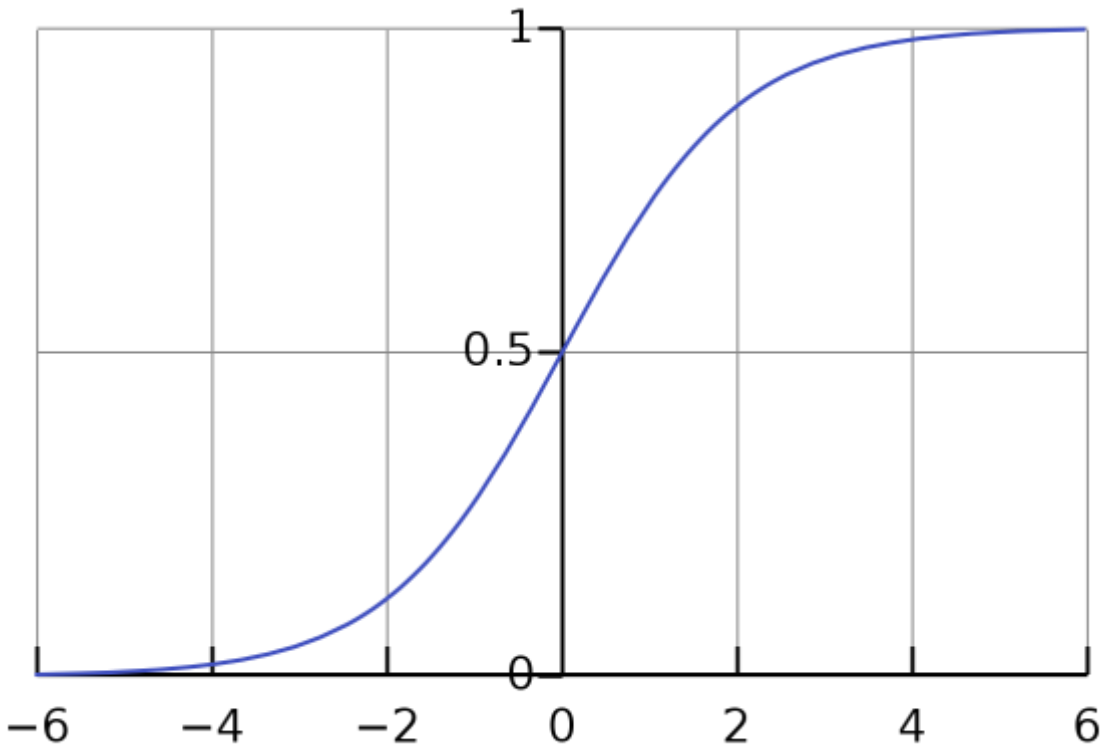
Logistic regression is a simple yet powerful algorithm for binary classification tasks, and its interpretability and efficiency make it a popular choice in various applications. It's important to note that logistic regression can be extended to handle multiclass classification problems using techniques like one-vs-all (OvA) or softmax regression.

Sigmoid function

$$S(x) = \frac{1}{1 + e^{-x}}$$

where

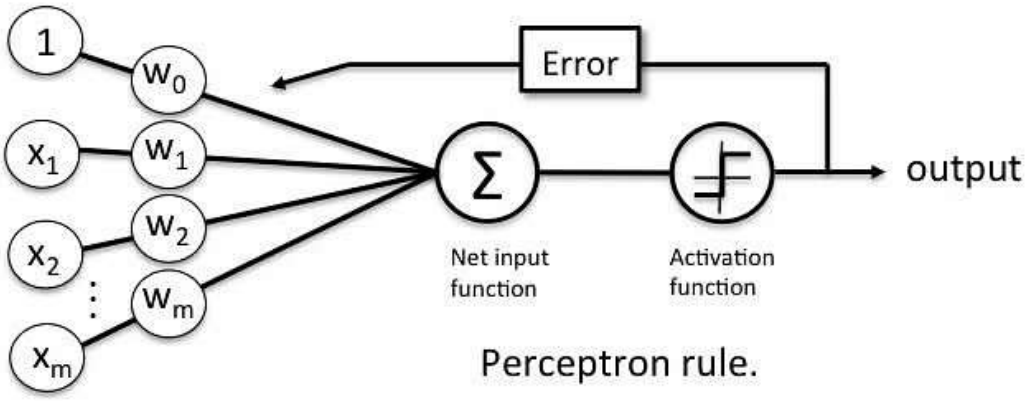
e is the base of the natural logarithm (approximately equal to 2.71828). The curve of this function starts close to 0 and ends close to 1 as x becomes very negative or very positive, respectively.



PERCEPTRON LEARNING ALGORITHM

Perceptron is a basic linear classification method suitable for simple tasks with linearly separable data, while logistic regression offers more flexibility and sophistication, making it a common choice for various classification problems. More advanced neural networks are typically preferred for handling complex, nonlinear classification tasks.
Perceptron learning algorithm is a linear classification method, and it's important to understand it in the context of other linear classification methods. Linear classification

methods are used for separating data points into different classes using linear decision boundaries, such as lines or hyperplanes.



This step function or Activation function is vital in ensuring that output is mapped between (0,1) or (-1,1). Take note that the weight of input indicates a node's strength. Similarly, an input value gives the ability the shift the activation function curve up or down.

Step 1: Multiply all input values with corresponding weight values and then add to calculate the weighted sum. The following is the mathematical expression of it:

$\sum wi*xi = x1*w1 + x2*w2 + x3*w3+……..x4*w4$

Add a term called bias 'b' to this weighted sum to improve the model's performance.

Step 2: An activation function is applied with the above-mentioned weighted sum giving us an output either in binary form or a continuous value as follows:

$Y=f(\sum wi*xi + b)$

The Perceptron learning algorithm is guaranteed to converge if the training data is linearly separable, meaning that it is possible to draw a straight line (in 2D) or a hyperplane (in higher dimensions) to separate the two classes. However, if the data is not linearly separable, the algorithm may not converge.