Linear REGRESSION:

[**Linear regression**](https://www.geeksforgeeks.org/ml-linear-regression/) is a statistical method that is used to predict a continuous dependent variable(target variable) based on one or more independent variables(predictor variables). This technique assumes a linear relationship between the dependent and independent variables, which implies that the dependent variable changes proportionally with changes in the independent variables. In other words, linear regression is used to determine the extent to which one or more variables can predict the value of the dependent variable.

Github:

<https://github.com/niharraju4/linear_R>

https://github.com/niharraju4/Multiple\_LinearRegression/blob/main/california\_housing.ipynb

**Multiple Linear Regression**

Multiple linear regression attempts to model the relationship between **two or more features** and a response by fitting a linear equation to the observed data.  
Clearly, it is nothing but an extension of simple linear regression. Consider a dataset with **p** features(or independent variables) and one response(or dependent variable).

Github:

https://github.com/niharraju4/Multiple\_LinearRegression/blob/main/multipleLR.ipynb

**Polynomial Linear Regression**

**Polynomial Regression**is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modelled as an *nth-degree* polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y | x).

Github:

https://github.com/niharraju4/Machine-learning/blob/main/polynomial\_linearRegression/poly.ipynb

**Navie baye’s:**

Github:

<https://github.com/niharraju4/Machine-learning/blob/main/Naive_Bayes_algo/naive.ipynb>

pdf:

https://github.com/niharraju4/Machine-learning/blob/main/Naive\_Bayes\_algo/7-Naive%20Baye's.pdf

**Random Forest**

The [Random forest](https://www.geeksforgeeks.org/videos/random-forest-algorithm-in-machine-learning/) or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees. Random Forests are particularly well-suited for handling large and complex datasets, dealing with high-dimensional feature spaces, and providing insights into feature importance. This algorithm’s ability to maintain high predictive accuracy while minimizing overfitting makes it a popular choice across various domains, including finance, healthcare, and image analysis, among others.

**Random Forest Classifier**

The Random forest classifier creates a[set](https://www.geeksforgeeks.org/set-in-cpp-stl/) of[decision trees](https://www.geeksforgeeks.org/decision-tree/) from a randomly selected subset of the training set. It is a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

*Random Forest Classifier*

Additionally, the random forest classifier can handle both classification and regression tasks, and its ability to provide feature importance scores makes it a valuable tool for understanding the significance of different variables in the dataset.

Pdf:

<https://github.com/niharraju4/Random_Forest-ML-/blob/main/RandomForest%20Regression.pdf>

github:

<https://github.com/niharraju4/Random_Forest-ML->

**What is a Decision Tree?**

A **decision tree**is a flowchart-like structure used to make decisions or predictions. It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing final outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.

Github:

https://github.com/niharraju4/Machine-learning/tree/main/DecisionTree

**Logistic regression**

**Logistic regression** is a **supervised machine learning algorithm**used for **classification tasks** where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors.

Github:

https://github.com/niharraju4/Logestic\_Regression

A **Support Vector Machine (SVM)** is a powerful **machine learning algorithm** widely used for both **linear and nonlinear classification**, as well as **regression** and **outlier detection** tasks. SVMs are highly adaptable, making them suitable for various applications such as **text classification**, **image classification**, **spam detection**, **handwriting identification**, **gene expression analysis**, **face detection**, and **anomaly detection**.

Pdf:

<https://github.com/niharraju4/Machine-learning/blob/main/SVM/svm.pdf>

github: https://github.com/niharraju4/Machine-learning/tree/main/SVM

The **K-Nearest Neighbors (KNN) algorithm** is a supervised machine learning method employed to tackle classification and regression problems.  KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the [supervised learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning) domain and finds intense application in pattern recognition, [data mining](https://www.geeksforgeeks.org/data-mining), and intrusion detection.

**Gradient Boosting** is a popular boosting algorithm in machine learning used for classification and regression tasks. Boosting is one kind of ensemble Learning method which trains the model sequentially and each new model tries to correct the previous model. It combines several weak learners into strong learners.

**XGBoost** is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

One of the key features of XGBoost is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing.

**PCA**

As the number of features or dimensions in a dataset increases, the amount of data required to obtain a statistically significant result increases exponentially. This can lead to issues such as overfitting, increased computation time, and reduced accuracy of machine learning models this is known as the curse of dimensionality problems that arise while working with high-dimensional data.

Github: https://github.com/niharraju4/Principal-component-analysis-PCA-/blob/main/pca/Principal%20Component%20Analysis%20(PCA)%20Implementation.ipynb

AdaBoost:

AdaBoost is a boosting algorithm that also works on the principle of the stagewise addition method where multiple weak learners are used for getting strong learners. The value of the alpha parameter, in this case, will be indirectly proportional to the error of the weak learner, Unlike Gradient Boosting in XGBoost, the alpha parameter calculated is related to the errors of the weak learner, here the value of the alpha parameter will be indirectly proportional to the error of the weak learner.

Pdf:

<https://github.com/niharraju4/AdaBoost/blob/main/11.1-%20Adaboost%20MAth%20intuition.pdf>

Github:

<https://github.com/niharraju4/AdaBoost>

**NLP**

**Spacy:**

Spacy is a library that comes under NLP (Natural Language Processing). It is an object-oriented Library that is used to deal with pre-processing of text, and sentences, and to extract information from the text using modules and functions.

Github:

<https://github.com/niharraju4/Natural_language_processing/tree/main/Spacy>

**StopWords:**

Github:

<https://github.com/niharraju4/Natural_language_processing/blob/main/stop_words.ipynb>

**Bag of Words:**

Github:

<https://github.com/niharraju4/Natural_language_processing/blob/main/spam_classifier/bag_of_words.ipynb>

**stemming:**

Stemming is the process of reducing a word to its root form or base form, often by removing suffixes and prefixes. The goal of stemming is to group words with the same meaning but different forms together, so they can be analyzed as a single item. This technique is widely used in natural language processing (NLP) and information retrieval systems, such as search engines, to improve the consistency and efficiency of data processing.

Github:

<https://github.com/niharraju4/Natural_language_processing/blob/main/preprocessingstemmingusingNLTK.ipynb>

**Lemmatization**

WordNet Lemmatization refers to the process of converting a word to its base or root form using the WordNet lexical database. WordNet is a large lexical database for the English language that groups words into sets of synonyms called synsets and provides short definitions and usage examples. The WordNetLemmatizer from the nltk library uses WordNet to find the lemma of a word, which means it considers the part of speech (POS) and context when reducing a word to its base form. This makes lemmatization more accurate and meaningful than simple stemming, as it aims to produce a valid dictionary word.

Github:

https://github.com/niharraju4/Natural\_language\_processing/blob/main/nltk\_preprocessing\_lemmanization.ipynb

**Parts of Speech:**

Github:

<https://github.com/niharraju4/Natural_language_processing/blob/main/parts_of_Speech.ipynb>

**Named Entity Recognition**

Github:

https://github.com/niharraju4/Natural\_language\_processing/blob/main/named\_entity\_recognition.ipynb

TF-IDF:

Github:

<https://github.com/niharraju4/Natural_language_processing/blob/main/TF-IDF.ipynb>

Word2Vec:

Github:

<https://github.com/niharraju4/Natural_language_processing/blob/main/Word2Vec.ipynb>

Tokenization:

Github:

https://github.com/niharraju4/Natural\_language\_processing/blob/main/Word2Vec.ipynb