Python Basics

A quick introduction to,

- Python syntax
- Variable Assignment
- Numbers
- Data Types
- Variables & Operators
- Functions
- Flow Controls
- Conditional Constructs
- Working with External Libraries

Working with External Libraries

- Imports
- Operator overloading
- Survival tips for venturing into the world of external libraries

Python Packages

NUMPY:

- Array
- Array Manipulations
- Functions
- Numerical Operations
- Indexing & Slicing
- Append & Concatenate

Pandas:

Creating, Reading and Writing

Indexing, Selecting & Assigning

Summary Functions and Maps

Grouping and Sorting

Data Types and Missing Values

Renaming and Combining

Introduction to Data Visualization

Data Visualization with Matplotlib & Seaborn

Line Charts

Bar Charts

Heatmaps

Scatter Plots

Area Chart

Pie Chart

Distributions

Box Plot

Choosing Plot Types and Custom Styles

Data Wrangling Techniques

Introduction to Data preprocessing

Importing the Dataset

Character Encodings

Handling Missing Values

Inconsistent Data Entry

Parsing Dates

Working with categorical Data

Splitting the data into Train and Test set

Outlier Analysis

Feature Scaling

Supervised Learning – Regression

Introduction to Regression

Linear Regression

Multi Linear Regression

Polynomial Regression

Decision Tree Regressor

Random Forest Regressor

Supervised Learning - Classification

Introduction to Classification

Logistic Regression

Decision Tree Classification

Random Forest Classification

K-nearest Neighbors

Naïve-Bayes

Support Vector Machine

XGboost

ANN Using Tensorflow & Keras

Introduction to Tensoflow & Keras

Introduction to ANN

Perfom Regression & Classification using ANN

Model Evaluation Metrics

Regression Evaluation Metrics:

- MAE
- MSE
- R Squared
- RMSE
- Classification metrics
- Confusion Metrics
- Accuracy
- Precision
- Recall F1 Score
- AUC ROC Curves

Hyper-parameter Optimization

Handling Imbalanced Data

Oversampling

Undersampling

Ensembling Techniques

SMOTE

Hyper-parameter tuning

Grid Search

Randomize Search

Unsupervised Learning

Introduction to Clustering

K-Means Clustering

Hierarchical Clustering

Clustering use cases

Build & Deploy ML Application

Introduction to different modes of deployment

Working with Flask Framework

Building application with flask framework

Integrating Machine Learning model with web application

Project Description

- The Thyroid gland is a vascular gland and one of the most important organs of the human body. This gland secretes two hormones which help in controlling the metabolism of the body.

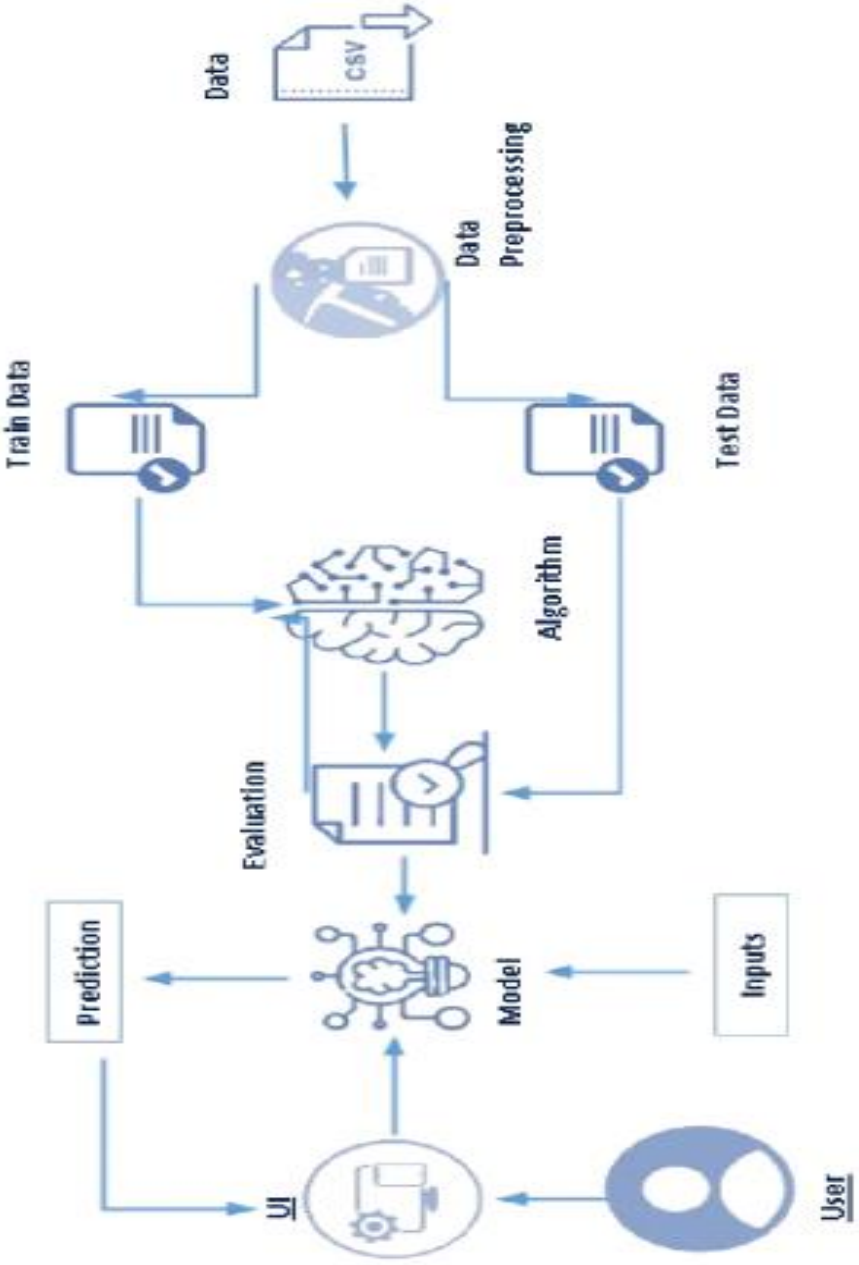
- The two types of Thyroid disorders are,
    1. Hyperthyroidism
    2. Hypothyroidism.

- When this disorder occurs in the body, they release certain types of hormones into the body which imbalances the body's metabolism. A thyroid-related Blood test is used to detect this disease but it is often blurred and noise will be present.

- Data cleansing methods were used to make the data primitive enough for the analytics to show the risk of patients getting this disease. Machine Learning plays a very deciding role in disease prediction.

- Machine Learning algorithms, SVM - support vector machine, Random Forest Classifier, XGB Classifier and ANN - Artificial Neural Networks are used to predict the patient's risk of getting thyroid disease. The web app is created to get data from users to predict the type of disease.

Technical Architecture:



Project flow:

- User interacts with the UI to enter the input.
- Entered input is analyzed by the model which is integrated.
- Once model analyzes the input the prediction is showcased on the UI.

To accomplish this, we have to complete all the activities listed below,

Data collection

- Collect the dataset or create the dataset

Visualizing and analyzing data

- Univariate analysis
- Bivariate analysis
- Multivariate analysis
- Descriptive analysis

Data pre-processing

- Checking for null values
- Handling outlier
- Handling categorical data
- Splitting data into train and test

Model building

- Import the model building libraries
- Initializing the model
- Training and testing the model
- Evaluating performance of model
- Save the model

Application Building

- Create an HTML file
- Build python code

Project Flow

Milestone 1: Define Problem / Problem Understanding

- **Activity 1: Specify the business problem**

  Refer Project Description

- **Activity 2: Business requirements**

  The business requirements for a project aimed at "Identifying Patterns and Trends in Campus Placement Data using Machine Learning" would likelyinclude the following:

- Access to campus placement data: The project would requireaccess to data on student performance, qualifications, and job placement outcomes. This data would need to be collected,
  cleaned, and prepared for analysis.

- Machine learning expertise: The project would require individuals with expertise in machine learning, data science andstatistical analysis to develop and implement the algorithms andmodels needed to analyze the data.

- Data storage and management: The project would require a robust and secure data storage and management system to storeand organize the large amounts of data used in the analysis.

- Infrastructure for model deployment: The project would require infrastructure for deploying the models and algorithms developed, including hardware, software, and cloud-based resources.

- **Activity 3: Literature Survey**

  - There have been several studies that have used machine learningtechniques to identify patterns and trends in campus placement data.
  - One study by authors P. K. Rajesh and Dr. G. R. Suresh, published in the International Journal of Computer Science and Mobile Computing in 2015, used k-means clustering and decision trees to analyze campus placement data and identify patterns that could be used to predict placement outcomes.
  - Another study by authors V.V. Kulkarni and K.S. Patil, publishedin the International Journal of Engineering Research and Technology in 2012, used decision tree and neural network algorithms to analyze campus placement data and identify factorsthat influence student placement.

  - A study by authors S.S. Bhosale, S.S. Raut, and D.S. Kulkarni, published in the International Journal of Emerging Research inManagement & Technology in 2013, used decision tree and Naive Bayes algorithms to analyze campus placement data andpredict student placement outcomes.

- A study by authors S.S. Bhosale, S.S. Raut, and D.S. Kulkarni, published in the International Journal of Emerging Research inManagement & Technology in 2013, used decision tree and Naive Bayes algorithms to analyze campus placement data andpredict student placement outcomes.
- In general, these studies found that machine learning techniques were effective at identifying patterns and trends in campus placement data, and could be used to predict student placementoutcomes with high accuracy.
- It's important to note that all these studies are quite old now and you might find more recent studies and new techniques whichcan be useful for your project.

Conclusion

Thyroid disease is one of the diseases that afflict the world's population, and the number of cases of this disease is increasing. Because of medical reports that show serious imbalances in thyroid diseases, our study deals with the classification of thyroid disease between hyperthyroidism and hypothyroidism. This disease was classified using algorithms. Machine learning showed us good results using several algorithms and was built in the form of two models. In the first model, all the characteristics consisting of 16 inputs and one output were taken, and the result of the accuracy of the random forest algorithm was 98.93, which is the highest accuracy among the other algorithms. In the second embodiment, the following characteristics were omitted based on a previous study. The removed attributes were 1- query_thyroxine 2- query_hypothyorid 3- query_hyperthyroid. Here we have included the increased accuracy of some algorithms, as well as the retention of the accuracy of others. It was observed that the accuracy of Naive Bayes algorithm increased the accuracy by 90.67. The highest precision of the MLP algorithm was 96.4 accuracy.