Mini Project On

ML Application Hub

By

Rashi Khandelwal (2021510028)

Under the guidance of

Internal Supervisor

Prof. Pallavi Thakur

Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University
2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Rashi Khandelwal (2021510028)

Have satisfactorily carried out work on the project entitled

“ML Application Hub”

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Prof. Pallavi Thakur

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Rashi Khandelwal (2021510028)

Have successfully completed the Project report on

“ML Application Hub”,

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER Prof. Pallavi Thakur	EXTERNAL EXAMINER
HEAD OF DEPARTMENT Prof. Dr. Pooja Raundale	PRINCIPAL Dr. Bhalchandra Chaudhari

Contents

Abstract

Machine Learning is one of the hottest research topics in computer science and engineering which is applicable in many disciplines. It provides a collection of algorithms, methods and tools capable of embodying some kind of intelligence to machines. Presently, many industrial projects have ML algorithms running at the back-end, focussing on one aspect only. The user is not aware of this implementation.

MLApplicationHub is a flask based, Machine Learning driven web application which clubs together different ML applications under one roof to demonstrate the wide scope of this technology in various domains like Healthcare, Commodity Pricing, Product Recommendations, Image Recognition to mention a few.

This flask web application demonstrates the use of ML in solving real-world problems in different domains under 6 main hoods :

- Classification - PIMA Indian Diabetes
- Regression - Car Price Prediction, Bengaluru House Price Estimation
- Recommendation Engine - Collaborative Movies Recommendation Engine
- Natural Language Processing - Duplicate Question Pair Identification, Email Spam/Ham Classification
- Chatbot - Currency Conversion
- Deep Learning - Fashion Image Search Engine, Stock Price Prediction, Body Posture Detection

Objectives

The flask based web application "MLApplicationHub" is used

- To put the theoretical ML concepts into application to solve real world problem like house price prediction, diagnose diabetes, classify mails as spam or ham and recommend movies based on your liking.
- To access various machine learning powered applications under one roof.
- To provide user-friendly and interactive web interface with view functionality of actual ML code running at the backend for the individuals interested in Data Science domain and eager to learn about ML techniques but perplexed about the roadmap to follow.

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

The purpose of Machine Learning Applications Hub system is to clubs together different ML driven applications under one roof to demonstrate the wide scope of this domain in medical diagnosis, email filtering, recommendation systems, future price prediction of commodities and reverse image search engines among others.

1.2 Objectives and Scope

1.2.1 Objectives

The flask based web application ”MLApplicationHub” is used

- To put the theoretical ML concepts into application to solve real world problem like house price prediction, diagnose diabetes, classify mails as spam or ham and recommend movies based on your liking.
- To access various machine learning powered applications under one roof.
- To provide user-friendly and interactive web interface with view functionality of actual ML code running at the backend for the individuals interested in Data Science domain and eager to learn about ML techniques but perplexed about the roadmap to follow.

1.2.2 Scope

Machine Learning is a vast field in itself with numerous algorithms for classification, recommendation and regression problems. However, for this project we limit ourselves to using certain models for 6 specific categories having below-mentioned applications :

- Classification - PIMA Indian Diabetes
- Regression - Car Price Prediction, Bengaluru House Price Estimation
- Recommendation Engine - Collaborative Movies Recommendation Engine
- Natural Language Processing - Duplicate Question Pair Identification, Email Spam/Ham Classification
- Chatbot - Currency Conversion
- Deep Learning - Fashion Image Search Engine, Stock Price Prediction, Body Posture Detection

1.3 Existing System

Presently, many industrial projects have machine learning algorithms running at the back-end, focussing on one aspect only. The user is not aware of this implementation. There is no such kind of application which demonstrates the utility of machine learning in various domains under one roof. A newbie, venturing into this domain of ML, finds himself perplexed and wonders where and how to begin from. Even after learning the mathematics and statics behind the complex algorithms, he searches how to realise them into practical utilities.

1.4 Proposed System

MLApplication Hub is a flask based, Machine Learning driven web application which clubs together different ML applications under one roof to demonstrate the wide scope of this technology in various domains like Healthcare, Commodity Pricing, Product Recommendations, Image Recognition to mention a few. It demonstrates the applications under 6 main hoods:

- **Classification Problem** - It demonstrates the binary classification of *Indian PIMA Diabetes* using Random Forest Classifier. The user had to input 8 parameters like age, glucose level, insulin level, no. of pregnancies, blood pressure and similar others. The model would predict whether the patient with given input parameters was either prone to diabetes or not.

- **Regression Problems** - Presents *laptop price prediction* using Ridge Model and *Bengaluru house price predictio*n using Random Forest Regressor. The user had to give input parameters like Company, ProductType, Inches, Screen Resolution, Cpu, Ram, HDD, SSD, Gpu, OpSys, Weight and model gave them the estimated price. In case of house price, user gave 4 input parameters like location, total sq ft, BHK and no. of bathrooms.

- **Recommendation Engine** - It deals with building a content-Based *Movie recommendation system* using Bag Of Words text vectorisation technique. The user had to select a movie of his choice from among 5000 available choices and the model gives back top 5 recommendations based on the similarity of the content.

- **Natural Language Processing** - It provides two sub-applications - *Email Spam/Ham Filtering* (using TF IDF text vectorisation technique and further Naive Baye's Model for classification) and *Duplicate Question Pairs Identification* (using Bag Of Words text vectorisation technique and further classified using XGBoast Model)

- **Chatbot** - It demonstrates a *Currency Conversion Chatbot* built using Google's Dialogflow, which can assist a user in converting any amount in x currency into y currency.

- **Deep Learning** - It will further depict 3 sub applications -
  - *Body Posture Detection* - It uses pre-trained Posnet Model developed by google and ML5.js to run ML on web browser. The model identified 17 body points in a single person video and drew a sketch for that.
  - *Stock Price Prediction* - It uses LSTM model for predicting the stock price of a company based on the historic close price of that stock.
  - *Fashion Reverse Image Search Engine* - There are about 44,441 images in the dataset, when the user uploaded an image, he was given back top 5 most similar images to the uploaded one. For this functionality, a pre-trained ResNET model which is trained by elite scientists on imageNET dataset is used and thus is very efficient and effective.

Some of the advantages of our system are as follows :

- **Numerous ML Application under one Roof**
  We club together different ML applications under one hood to demonstrate its wide scope in numerous domain.
- **Variety of Domains**
  We demonstrate the real-time utility of Machine Learning applications under various domains like health, fashion, movies, properties price, etc

1.5 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 20 GB Hard Disk Space for smooth run

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Device	Any Desktop or Laptop
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	Windows or Mac
Database	cvs files
Libraries and Packages	numpy, matplotlib, keras, pandas, tensorflow
Framework	Flask
Web Browser	Any browser that supports HTML, CSS, JS

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	OS Independent
Web Browser	Google Chrome, Safari

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

This project is intended To build an application that focuses on the realising the machine learning algorithms into practical utilities, along-with demonstrating the wide scope of this technology in various domains. The purpose of this document is to describe the functional requirements for Machine Learning Applications Hub system that clubs together different ML driven applications under one roof to demonstrate the wide scope of this domain in medical diagnosis, email filtering, recommendation systems, future price prediction of commodities and reverse image search engines among others.
It also provides a learning pathway for the individuals keen to venture in the world of data science and learn about the ML algorithms along with their applications.

2.2 Definition

1. ML - Machine Learning
2. SVM - Support Vector Machine
3. NB - Naive Baye's
4. CNN - Concurrent Neural Network
5. NLP - Natural Language Processing
6. TF - Tem Frequency
7. IDF - Inverse Term Frequency
8. CSV - Comma Separated
9. ERD - Entity Relationship Diagram
10. DB - Database

2.3 Overall Description

2.3.1 Functional Requirements

1. **Data Acquisition**: It is the process of gathering or collecting the dataset with correct and important features. When acquiring the data, we need to have enough features populated to train the learning model correctly. The dataset for various modules can be acquired from the below mentioned links:

- Diabetes Classification : Kaggle Indian PIMA Daibetes
- Movies Recommendation : Kaggle TMDB 5000 Movies
- House Price Prediction : Kaggle Bengaluru House Price Data
- Spam or Ham Message Identification : Kaggle Mails Data

2. **Data Cleaning and Pre-processing**: Data Cleaning is a critical process for the success of any machine learning function. It is a process used to determine inaccurate, incomplete or unreasonable data and then improve the quality through correcting detected errors and thus by reducing errors, improving the data quality. Data Cleaning can be a time consuming and tedious process but, it cannot be ignored. Data pre-processing is a technique which is used to transform the raw data in a useful and efficient format. The datasets may contain characters, strings and non-numeric values. These values cannot process by the programming hence, these values need to convert into numerical values. Vector mapping, binning and grouping are different approaches of processing the data.

3. **Exploratory Data Analysis**: It is an approach to analysing data sets to summarize their main characteristics often with visual or graphical methods. The Department of Computer Science, CHRIST (Deemed to be University) HealthDiagnosis Page 12 goal is to obtain confidence in data to a point where, it is ready to engage a machine learning algorithm.

4. **Feature Engineering**: It is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning. It increases the predictive power of machine learning algorithms by creating features from raw data that help facilitate the machine learning process.

5. **Splitting of Dataset**: The whole database is split into two sets: the training set and the testing set. The 70% data is taken for training while remaining 30% data is used for testing.

6. **Modelling**: The process of modelling means training a machine learning algorithm to predict the labels from the features, tuning it for the business needs, and validating it on holdout data. Different types of classifiers such as K-Nearest Neighbours, Random Forest, Support Vector Machine, Logistic Regression, Decision Tree and Extreme Gradient Boost are used for modelling purpose and identifying the best suited classifier for the testing phase. The details regarding the functioning and theory of the mentioned classifiers are provided in the appendices section of the report.

7. **Cross Validation and Grid Search**: Cross-validation is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset

of the data. We use cross-validation to detect overfitting, i.e., failing to generalize a pattern. Grid search is an approach to hyperparameter tuning that will methodically build and evaluate a model for each combination of algorithm parameters specified in a grid.

8. **Training**: The process of training the best fit ML model involves providing a learning algorithm with training data to learn from. The term ML model refers to the model artefact that is created by the training process. The training data must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

9. **Testing:** Employing new data points to the trained algorithm and predicting the output results. A test dataset is a dataset that is independent of the training dataset but, that follows the same probability distribution as the training dataset. If a model fit to the training dataset also fits the test dataset well, minimal over-fitting has taken place. A better fitting of the training dataset as opposed to the test dataset usually points to over-fitting.

3 Project Analysis and Design

3.1 Methodologies Adapted

The nature of the implementation techniques adopted in the ML Application Hub depends on the scope of the modules, architecture of the system, application performance and network - flask connection. A sequential model called V-model is followed which includes the below mentioned levels of implementation. The work products in each of the phase are as follows:

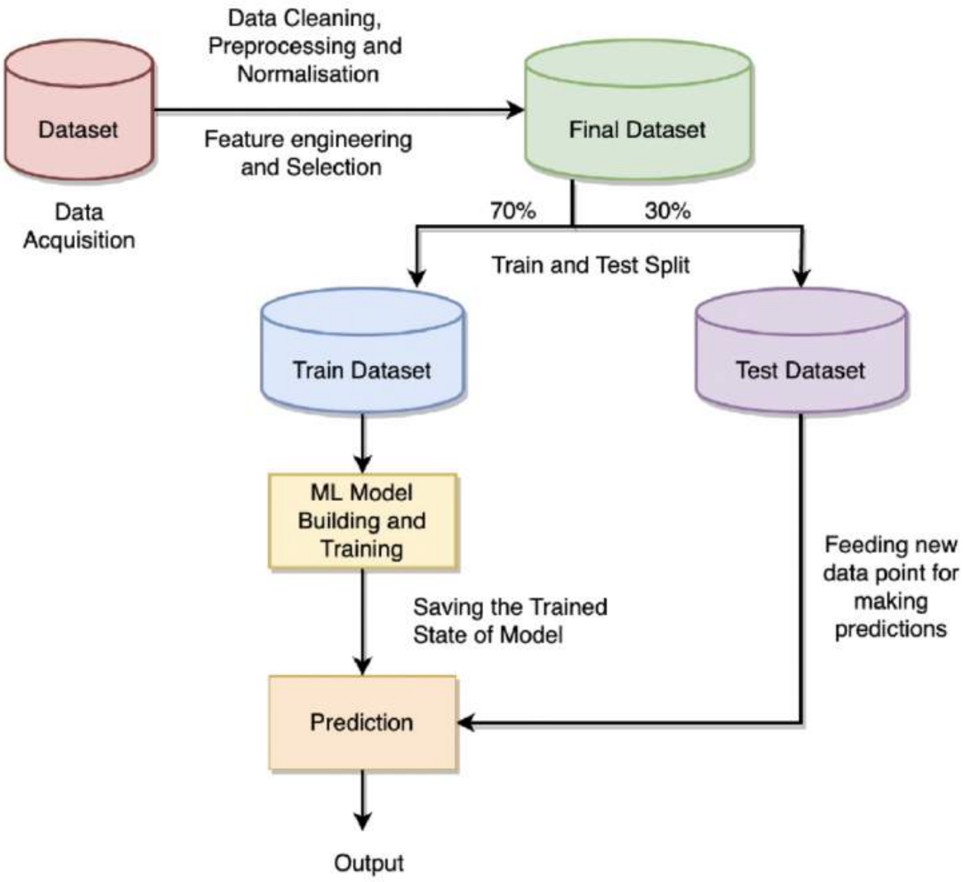
- **Requirement Analysis**: This is the first phase in the development cycle where the product requirements are understood from the end users perspective.

- **System Design**: The system design involves complete the understanding and detailing of the hardware and communication setup for the product under development.

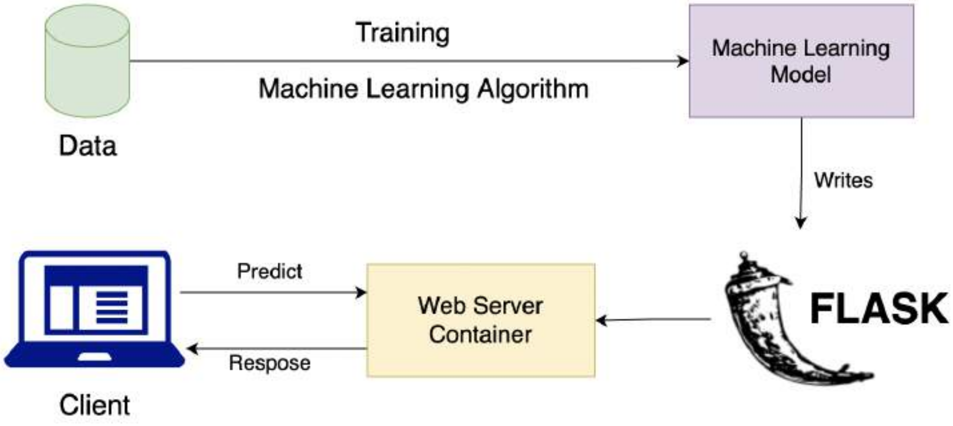
- **Architecture Design**: In this phase, the system design is broken down further into modules taking up different functionality. This is also referred to as High Level Design (HLD). The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage.

- **Module Design:** In this phase, the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems.

- **Coding Phase**: The actual coding of the system modules designed in the design phase is taken up in the Coding phase. ML Application Hub is built using python language, which is very suitable for the machine learning algorithms which form the core part of the system. The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance.


Dataset
Data Acquisition
Data Cleaning, Preprocessing and Normalisation
Feature engineering and Selection
Final Dataset
70%
30%
Train and Test Split
Train Dataset
Test Dataset
ML Model Building and Training
Saving the Trained State of Model
Feeding new data point for making predictions
Prediction
Output

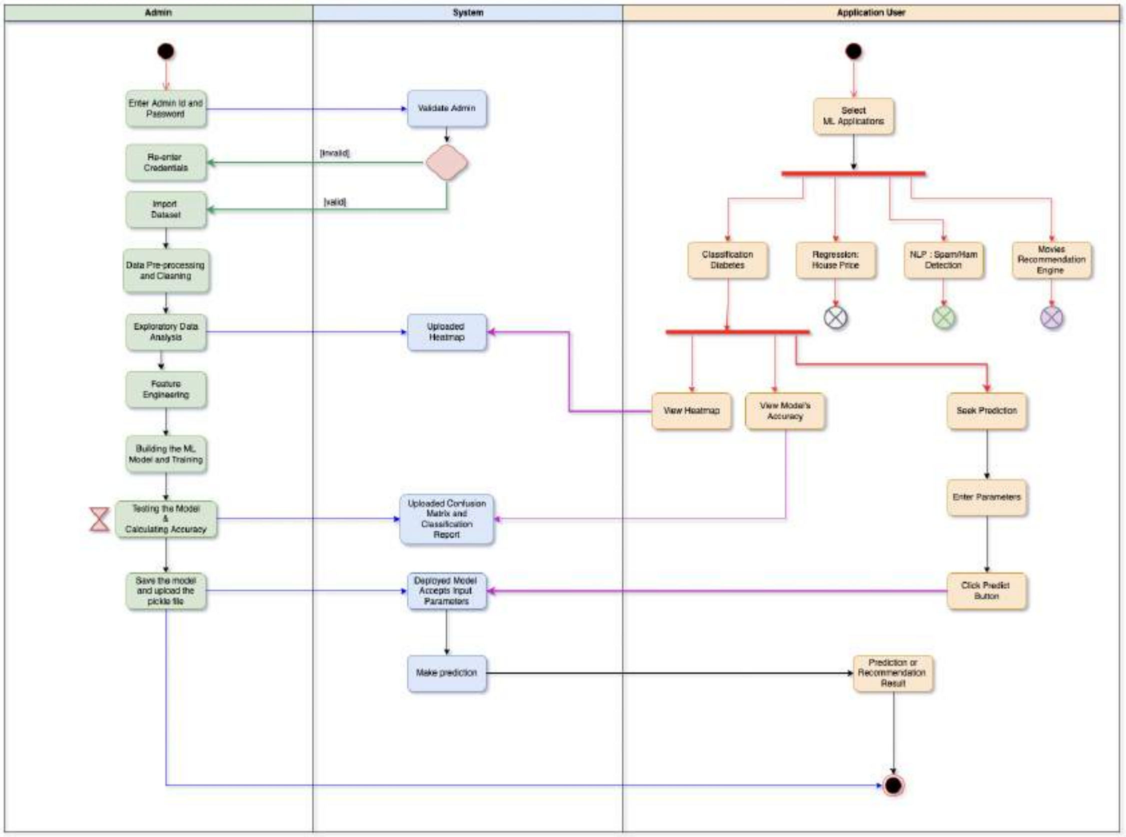
3.1.1: Diagrammatic Representation of Application Flow



3.1.2: System Architecture

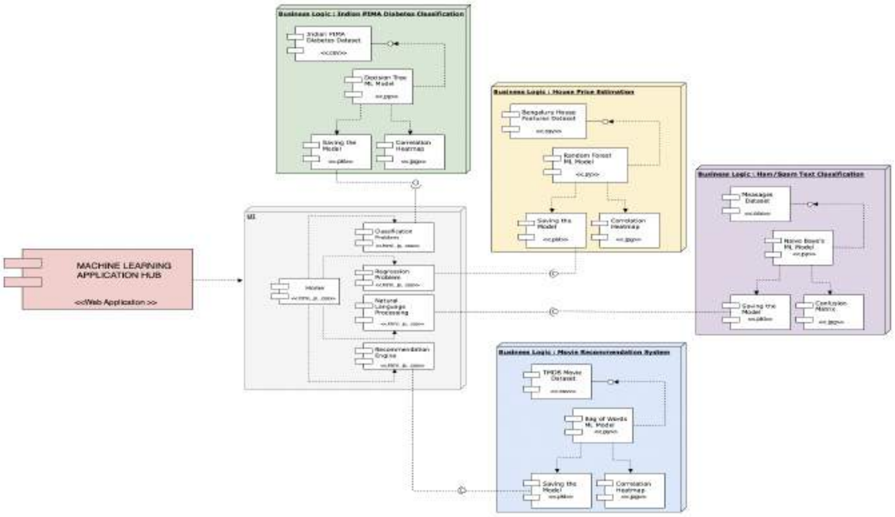
3.2 Modules

3.2.1 Activity diagram


Admin
System
Application User
Enter Admin Id and Password
Validate Admin
[invalid]
Re-enter Credentials
[valid]
Import Dataset
Data Pre-processing and Cleaning
Exploratory Data Analysis
Uploaded Heatmap
Feature Engineering
Building the ML Model and Training
Testing the Model & Calculating Accuracy
Uploaded Confusion Matrix and Classification Report
Save the model and upload the pickle file
Deployed Model Accepts Input Parameters
Make prediction
Select ML Applications
Classification Diabetes
Regression: House Price
NLP : Spam/Ham Detection
Movies Recommendation Engine
View Heatmap
View Model's Accuracy
Seek Prediction
Enter Parameters
Click Predict Button
Prediction or Recommendation Result

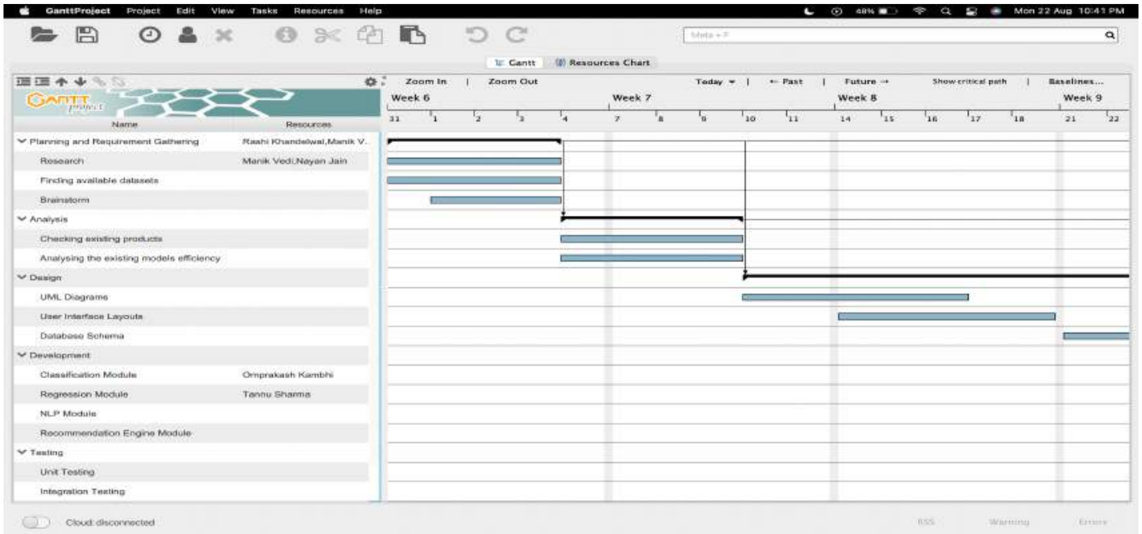
3.2.1: Activity Diagram

3.2.2 Component Diagram


Business Logic : Indian PIMA Diabetes Classification
Indian PIMA Diabetes Dataset <<csv>>
Decision Tree ML Model <<.py>>
Saving the Model <<.pkl>>
Correlation Heatmap <<.jpg>>
Business Logic : House Price Estimation
Bengaluru House Features Dataset <<.csv>>
Random Forest ML Model <<.py>>
Saving the Model <<.pkl>>
Correlation Heatmap <<.jpg>>
Business Logic : Ham/Spam Text Classification
Messages Dataset <<.tsv>>
Naive Baye's ML Model <<.py>>
Saving the Model <<.pkl>>
Confusion Matrix <<.jpg>>
MACHINE LEARNING APPLICATION HUB
<<Web Application >>
UI
Home <<.html, .js, .css>>
Classification Problem <<.html, .js, .css>>
Regression Problem <<.html, .js, .css>>
Natural Language Processing <<.html, .js, .css>>
Recommendation Engine <<.html, .js, .css>>
Business Logic : Movie Recommendation System
TMDB Movie Dataset <<.csv>>
Bag of Words ML Model <<.py>>
Saving the Model <<.pkl>>
Correlation Heatmap <<.jpg>>

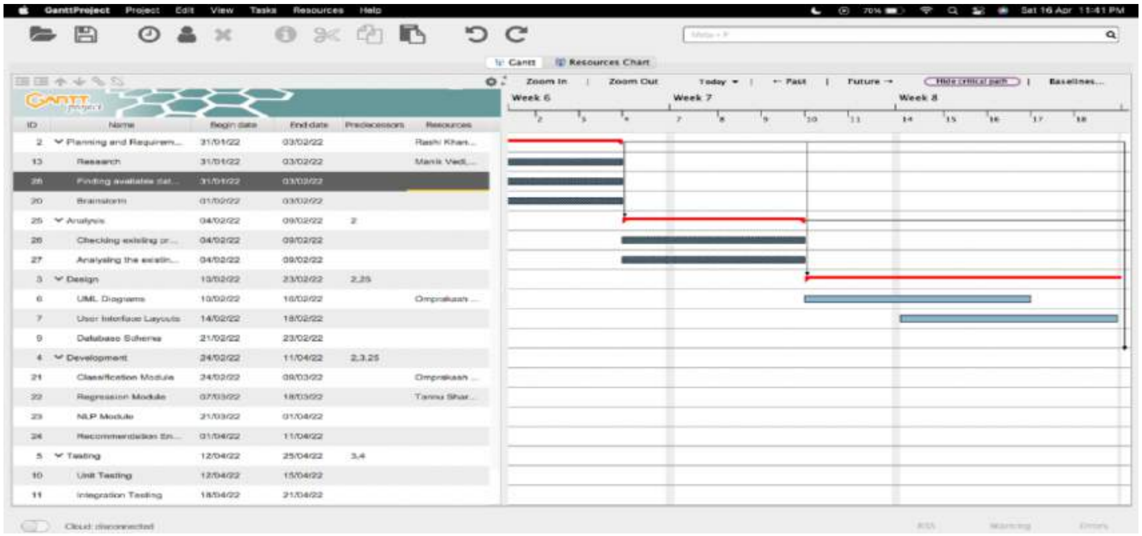
3.2.2: Component Diagram

3.2.3 Work Breakdown Structure and Gantt Chart



Name	Resources
Planning and Requirement Gathering	Rashi Khandelwal,Manik V.
Research	Manik Vedi,Nayan Jain
Finding available datasets	
Brainstorm	
Analysis	
Checking existing products	
Analysing the existing models efficiency	
Design	
UML Diagrams	
User Interface Layouts	
Database Schema	
Development	
Classification Module	Omprakash Kambhi
Regression Module	Tannu Sharma
NLP Module	
Recommendation Engine Module	
Testing	
Unit Testing	
Integration Testing	

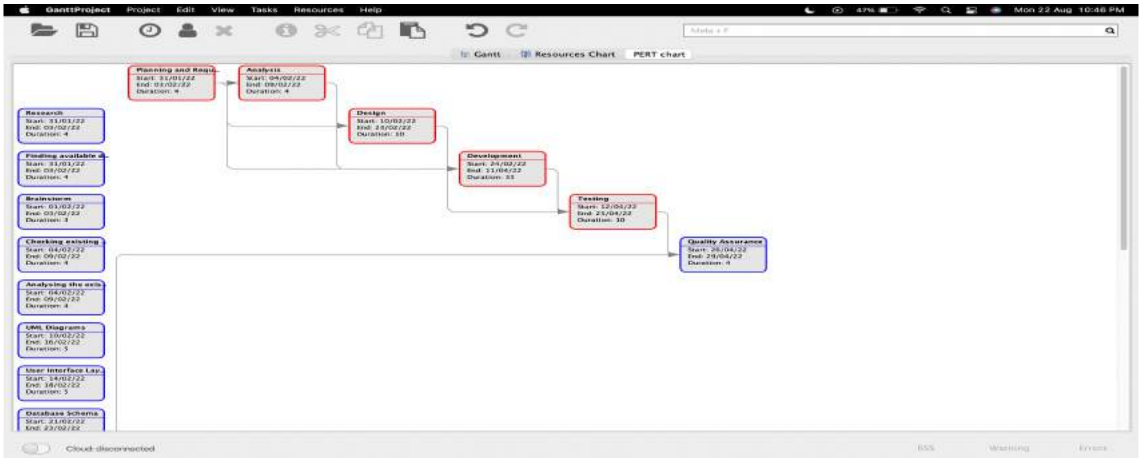
3.2.3: Gantt Chart



ID	Name	Begin date	End date	Predecessors	Resources
2	Planning and Requirem...	31/01/22	03/02/22		Rashi Khan...
13	Research	31/01/22	03/02/22		Mansi Vedi...
28	Finding available dat...	31/01/22	03/02/22		
20	Brainstorm	01/02/22	03/02/22		
25	Analysis	04/02/22	09/02/22	2	
26	Checking existing pr...	04/02/22	09/02/22		
27	Analysing the existin...	04/02/22	09/02/22		
3	Design	10/02/22	23/02/22	2,25	
6	UML Diagrams	10/02/22	16/02/22		Omprakash ...
7	User Interface Layouts	14/02/22	18/02/22		
9	Database Schema	21/02/22	23/02/22		
4	Development	24/02/22	11/04/22	2,3,25	
21	Classification Module	24/02/22	09/03/22		Omprakash ...
22	Regression Module	07/03/22	18/03/22		Tannu Shar...
23	NLP Module	21/03/22	01/04/22		
24	Recommendation En...	01/04/22	11/04/22		
5	Testing	12/04/22	25/04/22	3,4	
10	Unit Testing	12/04/22	15/04/22		
11	Integration Testing	18/04/22	21/04/22		

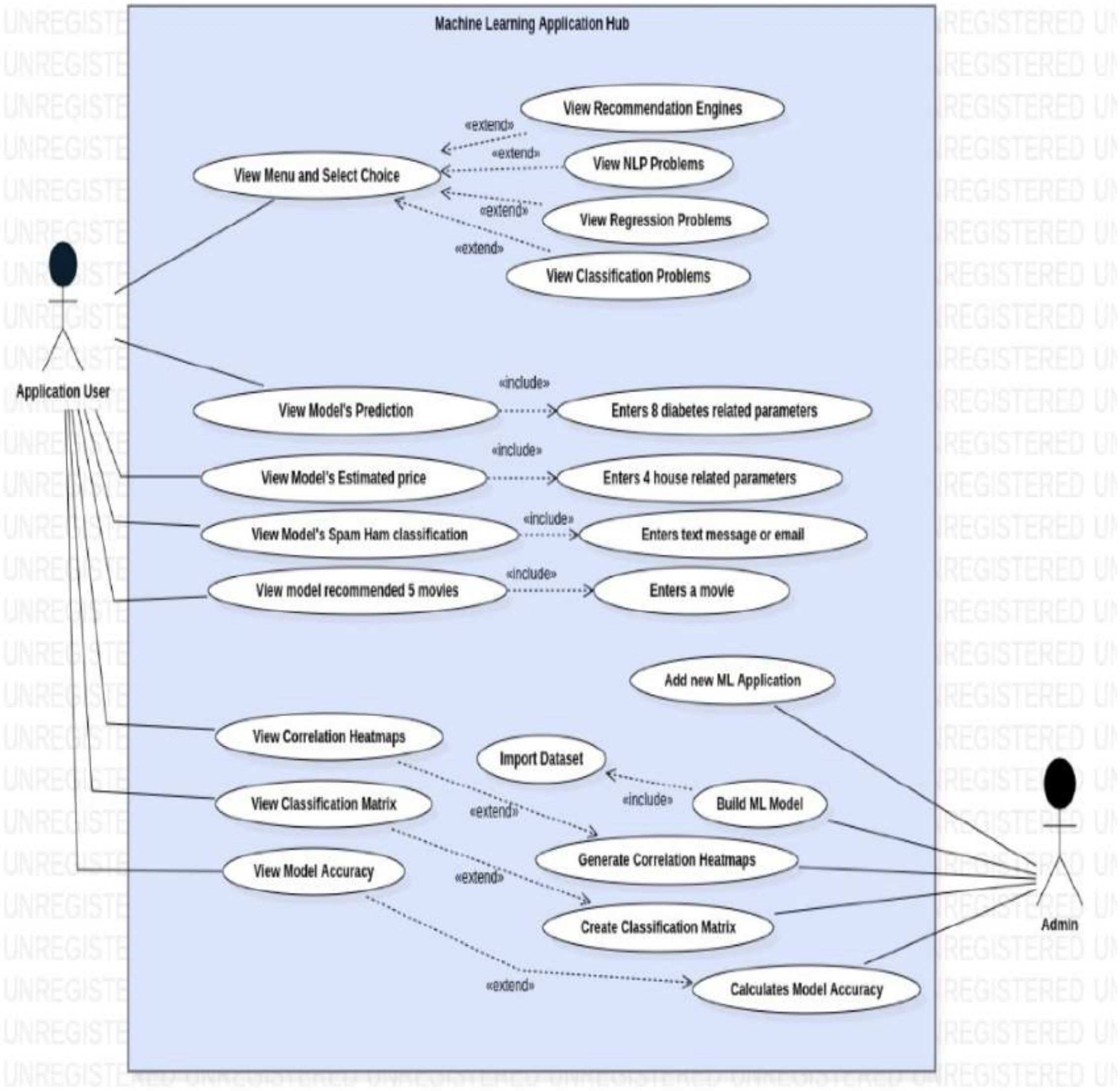
3.2.4: Gantt Chart

3.2.4 PERT Chart



3.2.5: PERT Chart

3.2.5 Use-Case


Machine Learning Application Hub
View Recommendation Engines
«extend»
«extend»
View NLP Problems
View Menu and Select Choice
«extend»
View Regression Problems
«extend»
View Classification Problems
Application User
«include»
View Model's Prediction
Enters 8 diabetes related parameters
«include»
View Model's Estimated price
Enters 4 house related parameters
«include»
View Model's Spam Ham classification
Enters text message or email
«include»
View model recommended 5 movies
Enters a movie
Add new ML Application
View Correlation Heatmaps
Import Dataset
«include»
Build ML Model
View Classification Matrix
«extend»
Generate Correlation Heatmaps
View Model Accuracy
«extend»
Create Classification Matrix
Admin
«extend»
Calculates Model Accuracy

3.2.6: Use-Case Diagram

Table 4.2.1: Use Case Specification

Use Case ID	MLAPPHUB1
Use Case Name	Machine Learning Application Hub
Created By	Rashi Khandelwal
Date Created	August 17, 2022

Table 4.2.2: Use Case Table - Description

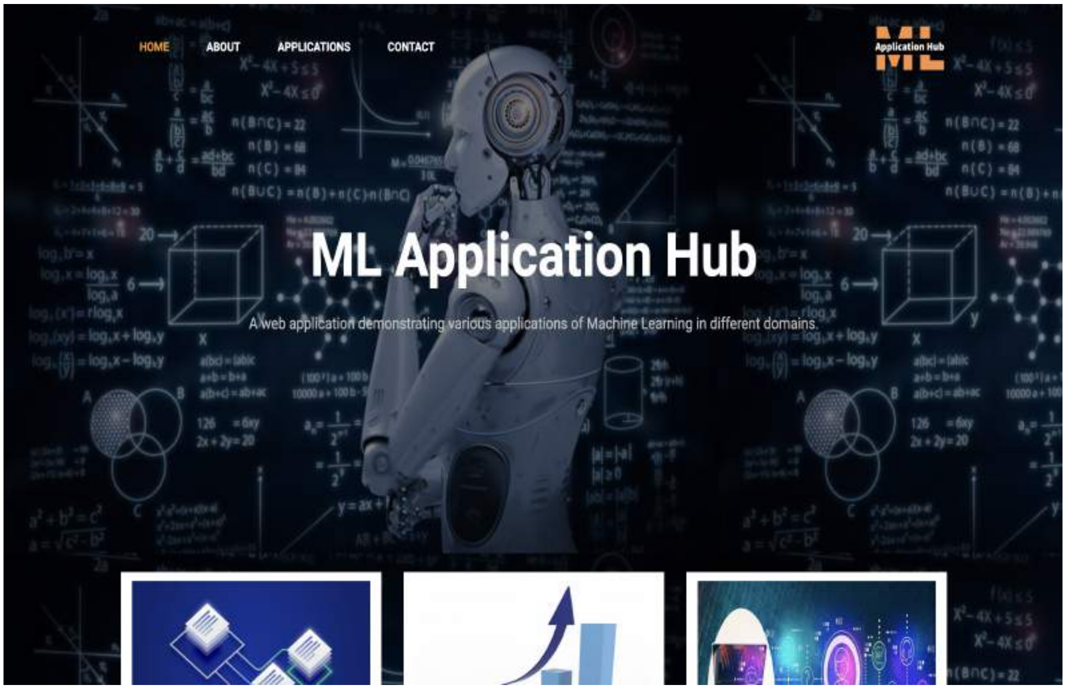
Actor	1. Admin 2. Application User
Description of use cases:	1. Add new ML Application - The admin adds a new ML application. 2. Import Dataset - The admin imports dataset used for model training. 3. Build ML Model - The admin builds the ML model for the required task. 4. Generate Correlation Heatmaps - The admin generates correlation heatmaps of the attributes in the dataset. 5. Create Classification Matrix - 6. Calculates Model Accuracy 7. View Model's Prediction 8. View Model's Estimated price 9. View Model's Spam Ham classification 10. View model recommended 5 movies 11. View Menu and Select Choice 12. View Classification Problems 13. View Regression Problems 14. View NLP Problems 15. View Recommendation Engines 16. Enters 8 diabetes related parameters 17. Enters 4 house related parameters 18. Enters text message or email 19. Enters a movie 20. View Correlation Heatmap 21. View Classification Matrix 22. View Model Accuracy

Table 4.2.1: Use Case Specification

Preconditions	The admin has to be authenticated.
Postconditions	1. The admin preprocesses and cleans the imported dataset. 2. When the admin adds a new application, it must be reflected to the user.
Extends	1. View Recommendation Engines may nor may not be invoked by view menu. 2. View NLP Problems may nor may not be invoked by view menu. 3. View Regression Problems may nor may not be invoked by view menu. 4. View Classification Problems may nor may not be invoked by view menu. 5. View Correlation Heatmaps may nor may not be invoked by Generate Correlation Heatmaps. 6. View Classification Matrix may nor may not be invoked by Create Classification Matrix. 7. View Model Accuracy may nor may not be invoked by Calculate Model Accuracy.
Includes	1. Build ML Model invokes Import Dataset. 2. View Model's Prediction requires user to Enter 8 diabetes related parameters. 3. View Model's Estimated price requires user to Enter 4 house related parameters. 4. View Model's Spam Ham classification requires user to Enters text message or email. 5. View model recommended 5 movies requires user to Enters a movie.
Assumptions	Application user is assumed to be any end-user or ML enthusiast.

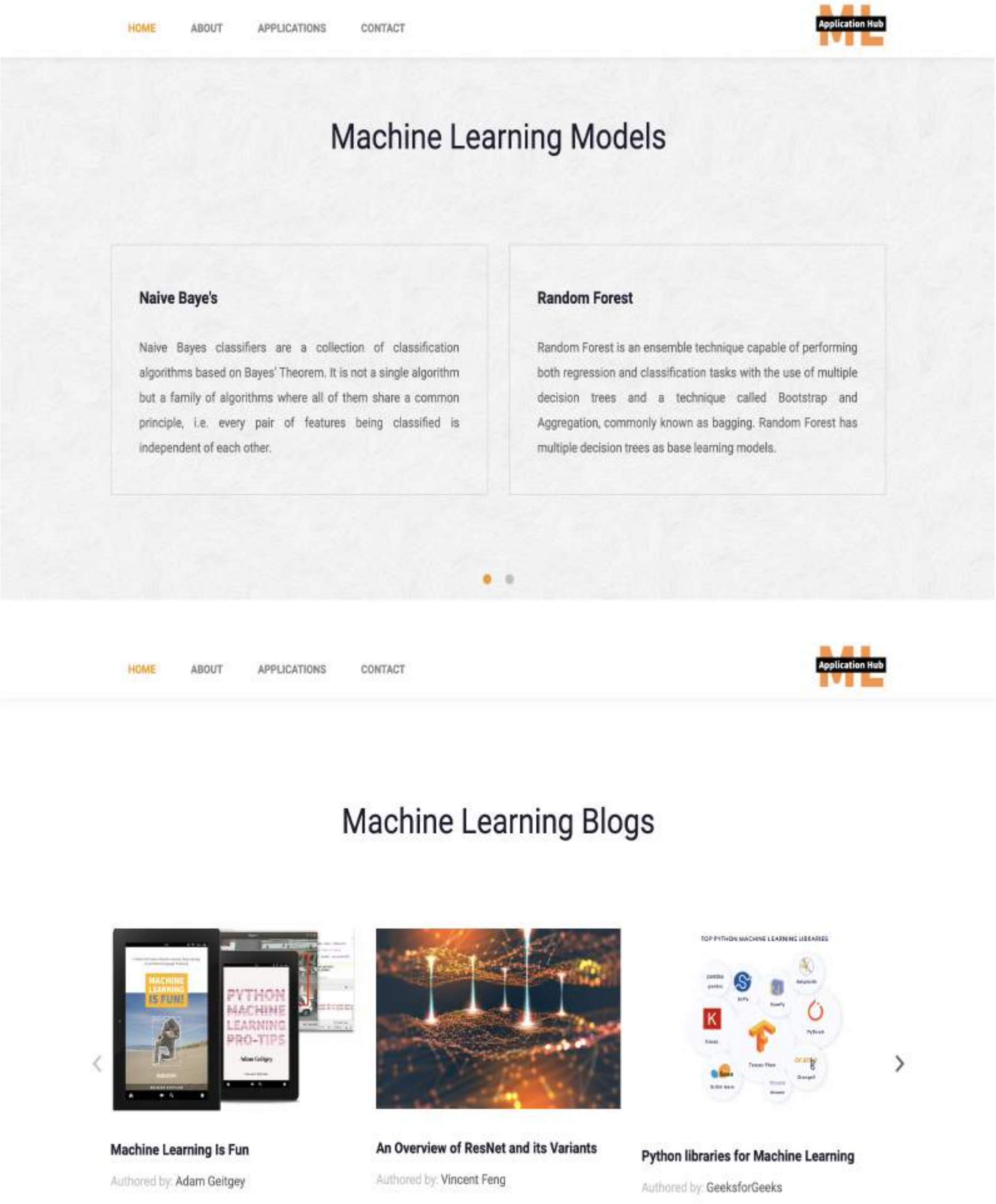
4 Project Implementation and Testing

4.1 Home Page



ML Application Hub

A web application demonstrating various applications of Machine Learning in different domains.



Machine Learning Models

Naive Baye's

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Random Forest

Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. Random Forest has multiple decision trees as base learning models.

Machine Learning Blogs

Machine Learning Is Fun

Authored by: Adam Geitgey

An Overview of ResNet and its Variants

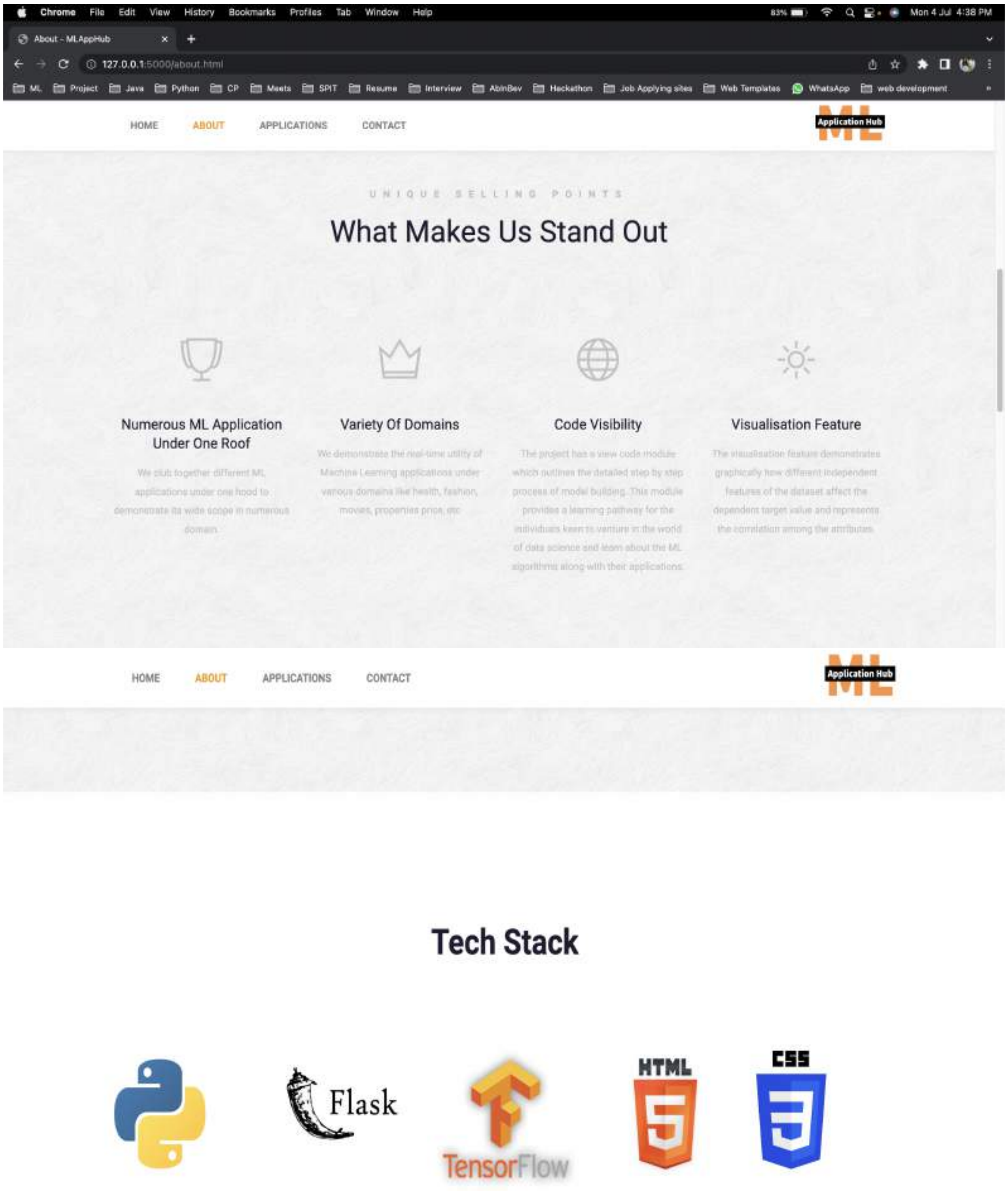
Authored by: Vincent Feng

Python libraries for Machine Learning

Authored by: GeeksforGeeks

4.1.1: Landing Page

4.2 About Page



UNIQUE SELLING POINTS

What Makes Us Stand Out

Numerous ML Application Under One Roof

We club together different ML applications under one hood to demonstrate its wide scope in numerous domain.

Variety Of Domains

We demonstrate the real-time utility of Machine Learning applications under various domains like health, fashion, movies, properties price, etc

Code Visibility

The project has a view code module which outlines the detailed step by step process of model building. This module provides a learning pathway for the individuals keen to venture in the world of data science and learn about the ML algorithms along with their applications.

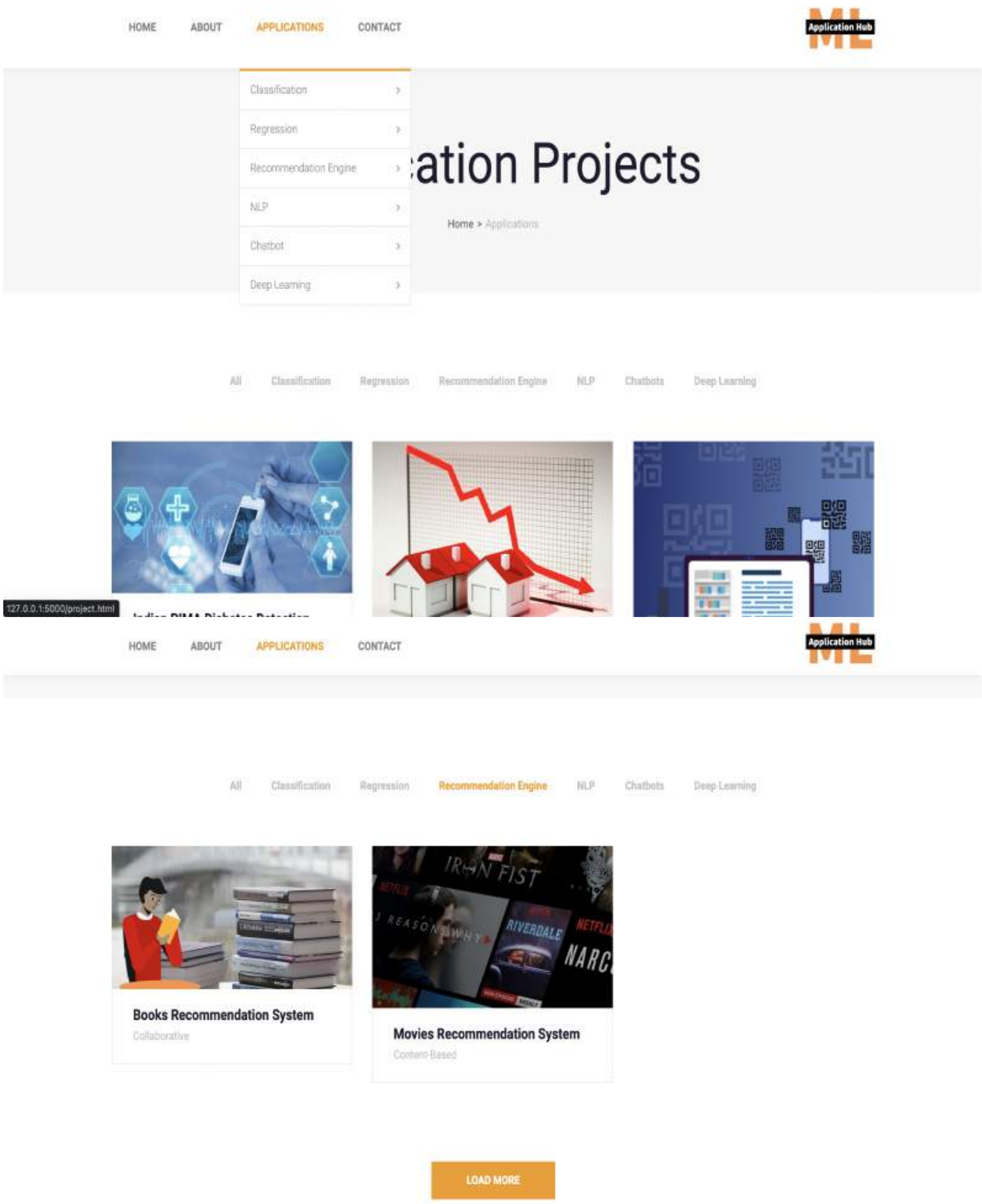
Visualisation Feature

The visualisation feature demonstrates graphically how different independent features of the dataset affect the dependent target value and represents the correlation among the attributes.

Tech Stack

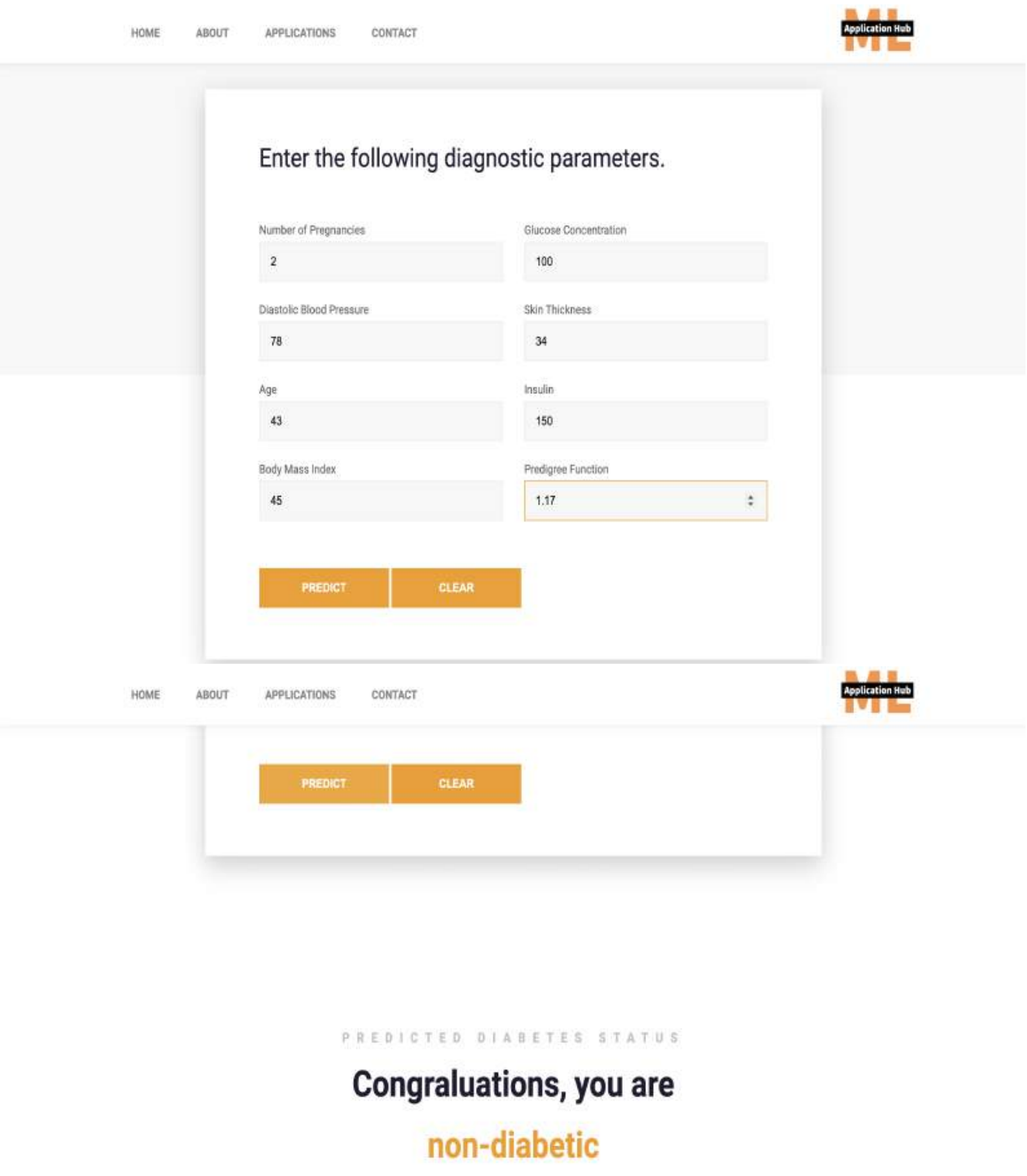
4.2.1: About Us

4.3 Applications Page



4.3.1: Applications

4.4 Classification - Indian PIMA Diabetes Prediction



Enter the following diagnostic parameters.

Number of Pregnancies	Glucose Concentration
2	100
Diastolic Blood Pressure	Skin Thickness
78	34
Age	Insulin
43	150
Body Mass Index	Predigree Function
45	1.17

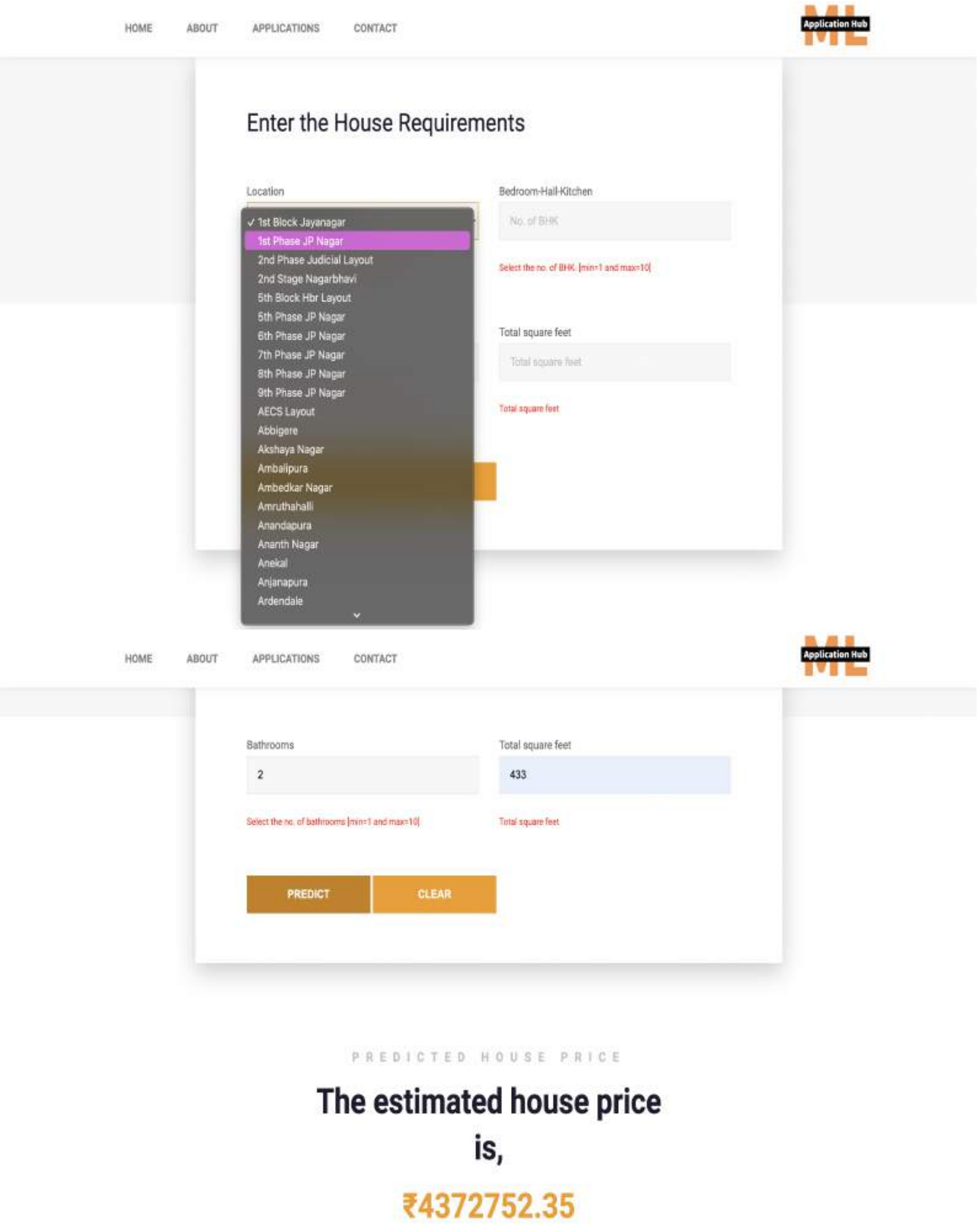
PREDICTED DIABETES STATUS

Congraluations, you are

non-diabetic

4.4.1: Indian PIMA Diabetes Prediction

4.5 Regression - Bengaluru House Price Prediction



Enter the House Requirements

Location

✓ 1st Block Jayanagar

Bedroom-Hall-Kitchen

No. of BHK

Select the no. of BHK. [min=1 and max=10]

Total square feet

Total square feet

Total square feet

Bathrooms: 2

Select the no. of bathrooms [min=1 and max=10]

Total square feet: 433

Total square feet

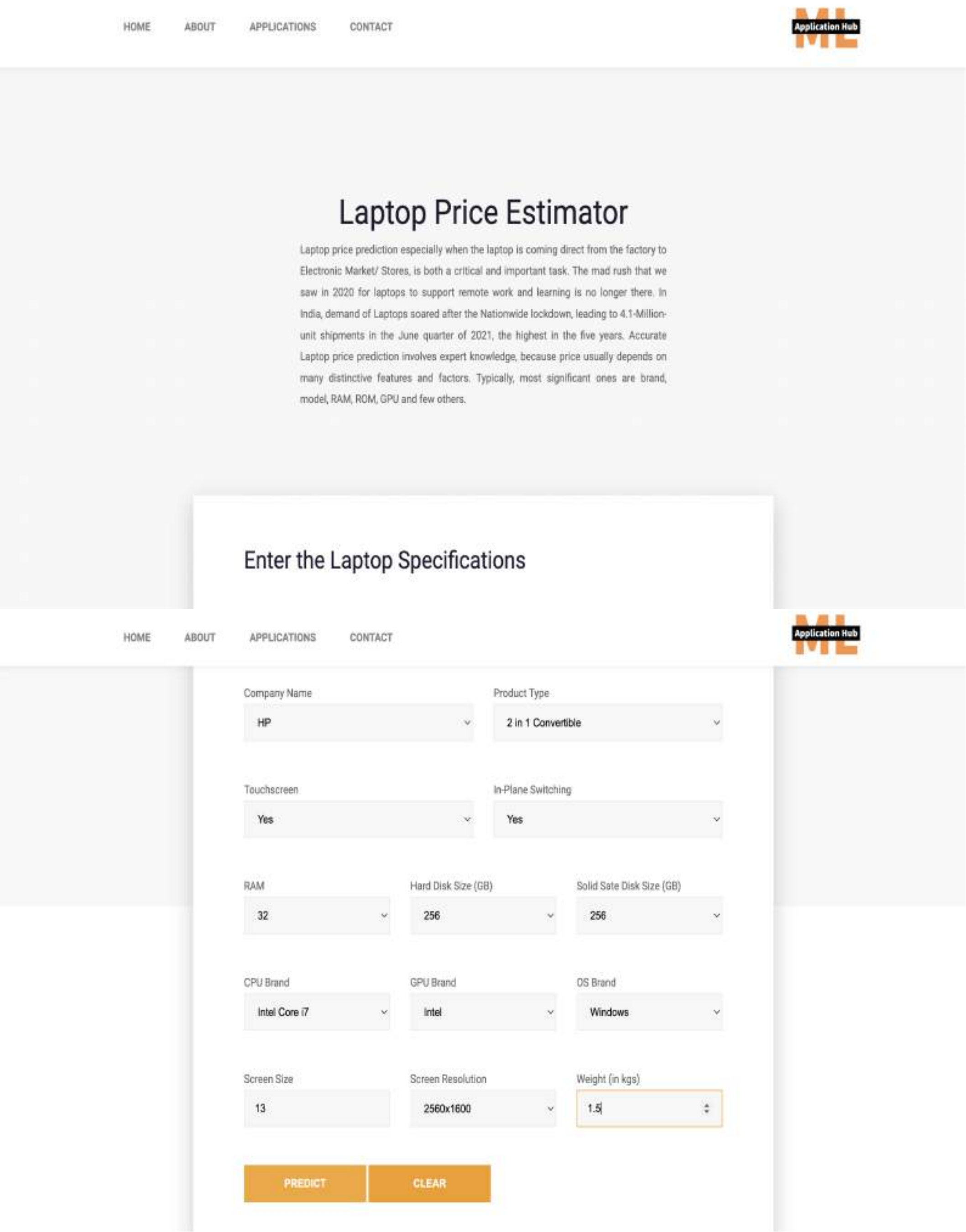
PREDICTED HOUSE PRICE

The estimated house price is,

₹4372752.35

4.5.1: Bengaluru House Price Prediction

4.6 Regression - Laptop Price Prediction



Laptop Price Estimator

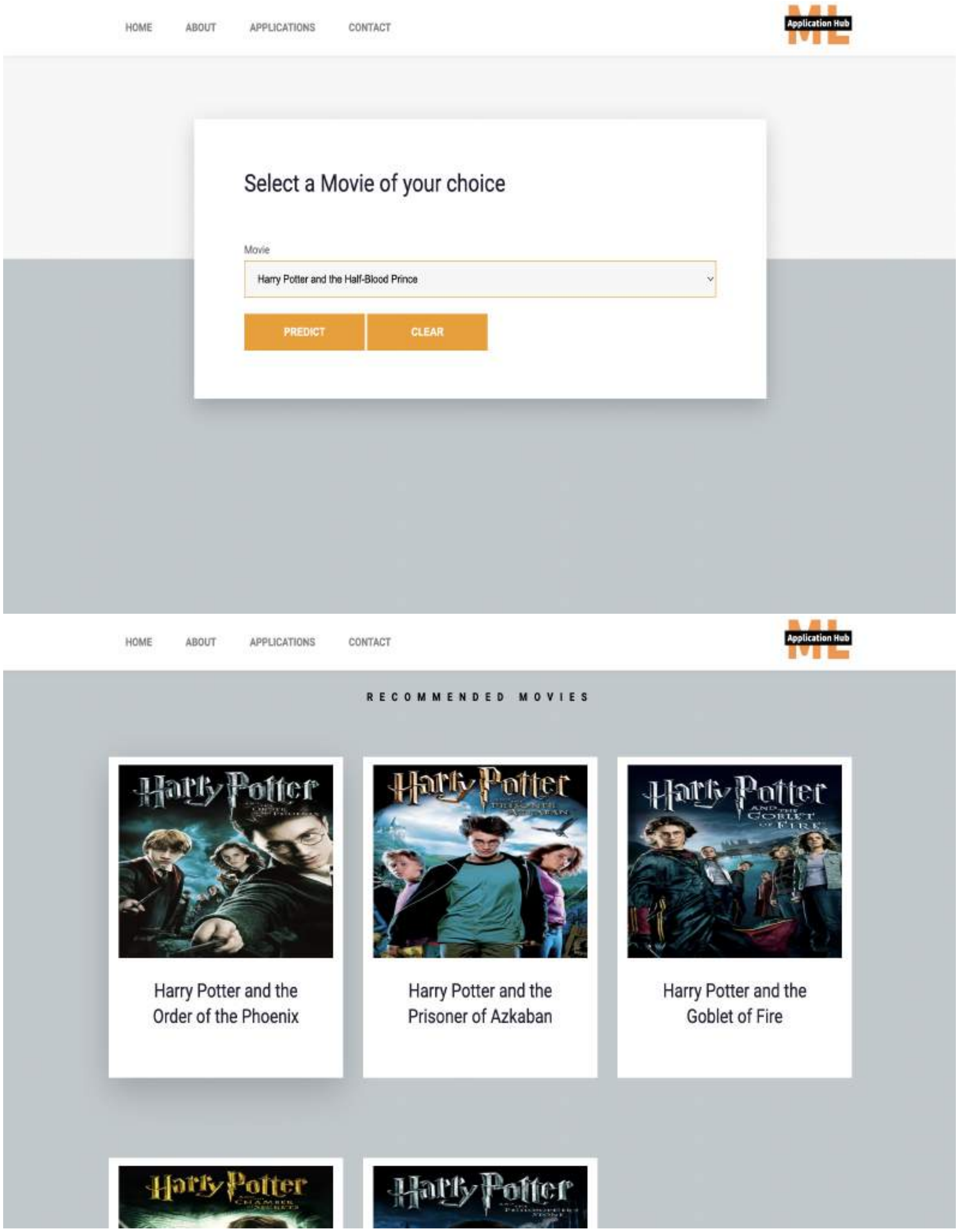
Laptop price prediction especially when the laptop is coming direct from the factory to Electronic Market/ Stores, is both a critical and important task. The mad rush that we saw in 2020 for laptops to support remote work and learning is no longer there. In India, demand of Laptops soared after the Nationwide lockdown, leading to 4.1-Million-unit shipments in the June quarter of 2021, the highest in the five years. Accurate Laptop price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand, model, RAM, ROM, GPU and few others.

Enter the Laptop Specifications

Company Name	HP
Product Type	2 in 1 Convertible
Touchscreen	Yes
In-Plane Switching	Yes
RAM	32
Hard Disk Size (GB)	256
Solid Sate Disk Size (GB)	256
CPU Brand	Intel Core i7
GPU Brand	Intel
OS Brand	Windows
Screen Size	13
Screen Resolution	2560x1600
Weight (in kgs)	1.5

4.6.1: Laptop Price Prediction

4.7 Recommendation Engine - Content Based : Movies



Select a Movie of your choice

Movie: Harry Potter and the Half-Blood Prince

RECOMMENDED MOVIES

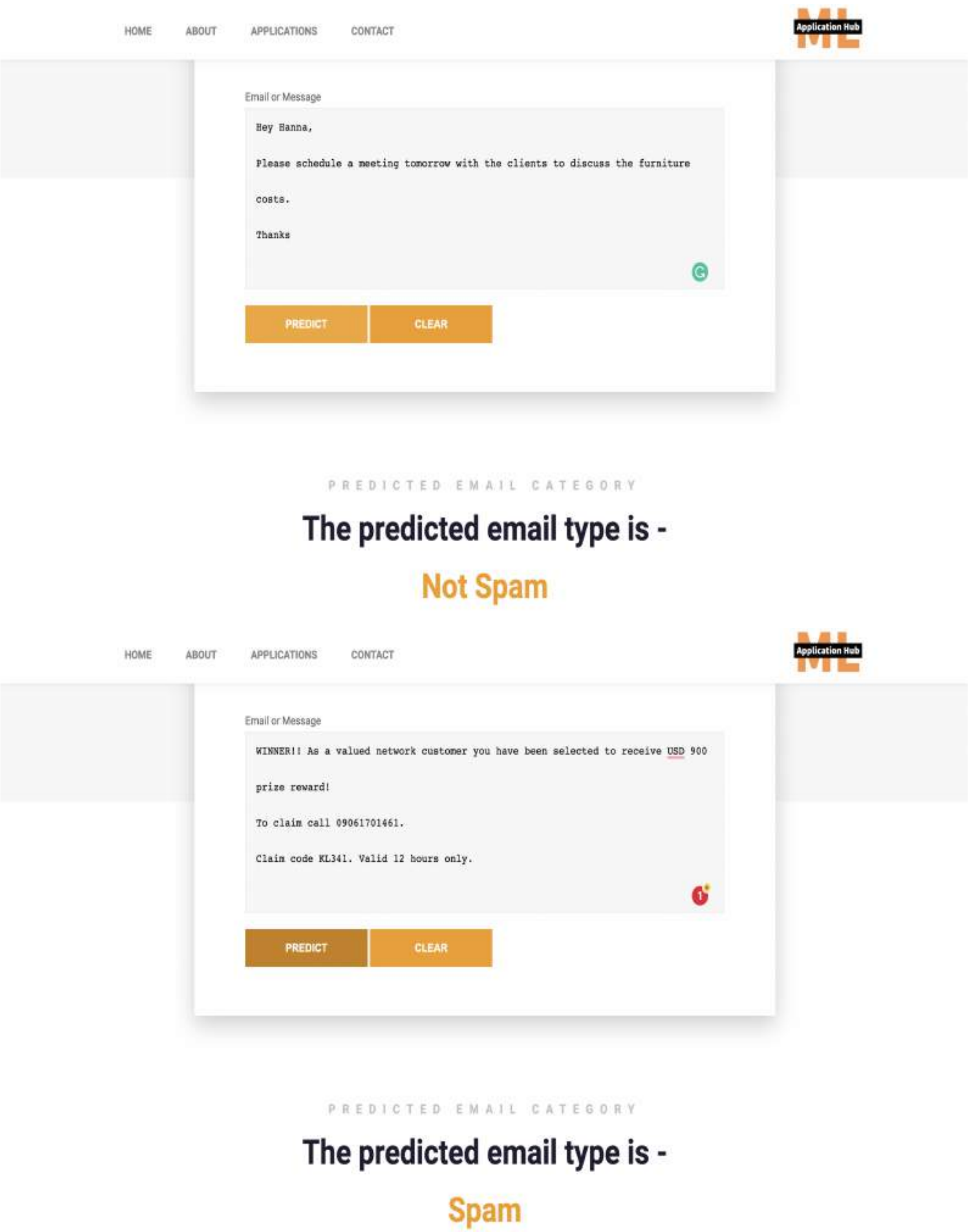
Harry Potter and the Order of the Phoenix

Harry Potter and the Prisoner of Azkaban

Harry Potter and the Goblet of Fire

4.7.1: Movies Recommendation Engine

4.8 NLP - Email Spam/Ham Classification



```
Hey Hanna,

Please schedule a meeting tomorrow with the clients to discuss the furniture

costs.

Thanks
```

PREDICTED EMAIL CATEGORY

The predicted email type is -

Not Spam

```
WINNER!! As a valued network customer you have been selected to receive USD 900

prize reward!

To claim call 09061701461.

Claim code KL341. Valid 12 hours only.
```

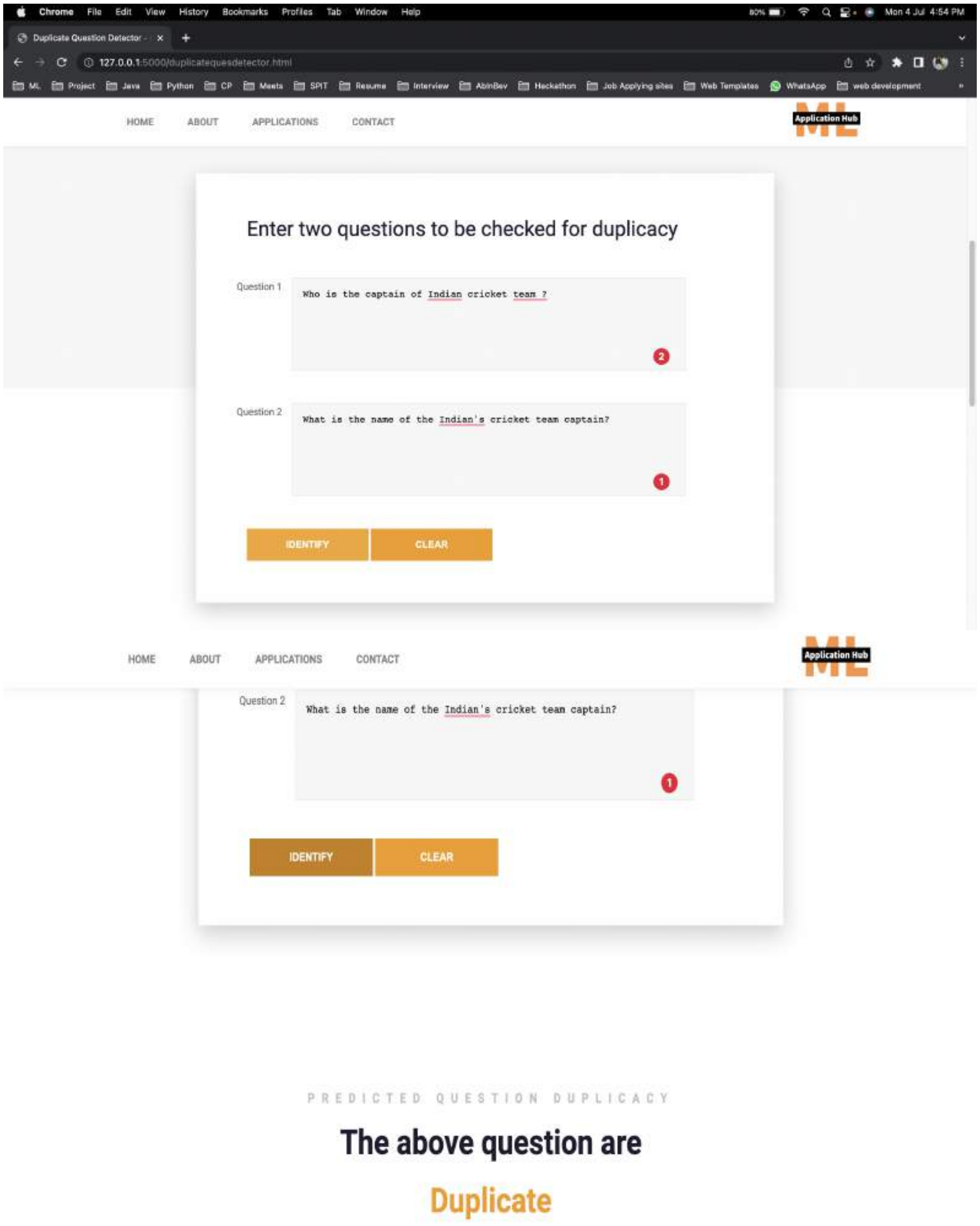
PREDICTED EMAIL CATEGORY

The predicted email type is -

Spam

4.8.1: Email Spam/Ham Classification

4.9 NLP - Duplicate Question Pair Identification


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 4 Jul 4:54 PM
Duplicate Question Detector
127.0.0.1:5000/duplicatequesdetector.html
ML Project Java Python CP Meets SPIT Resume Interview AbInBev Hackathon Job Applying sites Web Templates WhatsApp web development
HOME ABOUT APPLICATIONS CONTACT
Application Hub
ML
IDENTIFY
CLEAR

Enter two questions to be checked for duplicacy

Question 1

```
Who is the captain of Indian cricket team ?
```

Question 2

```
What is the name of the Indian's cricket team captain?
```

Question 2

```
What is the name of the Indian's cricket team captain?
```

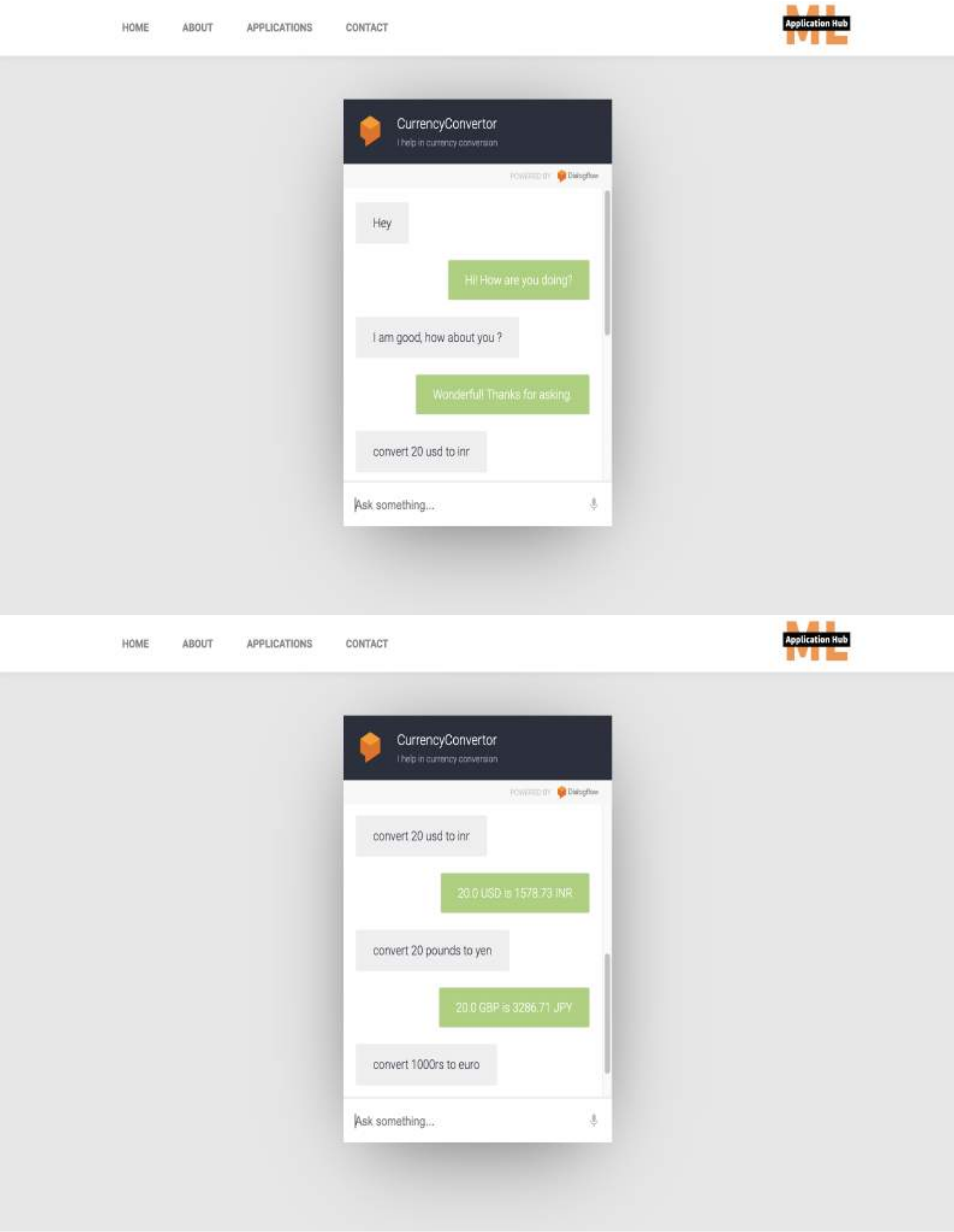
PREDICTED QUESTION DUPLICACY

The above question are

Duplicate

4.9.1: Duplicate Question Pair Identification

4.10 Chatbot - Currency Conversion



CurrencyConvertor

I help in currency conversion

Hey

Hi! How are you doing?

I am good, how about you ?

Wonderful! Thanks for asking

convert 20 usd to inr

CurrencyConvertor

I help in currency conversion

convert 20 usd to inr

20.0 USD is 1578.73 INR

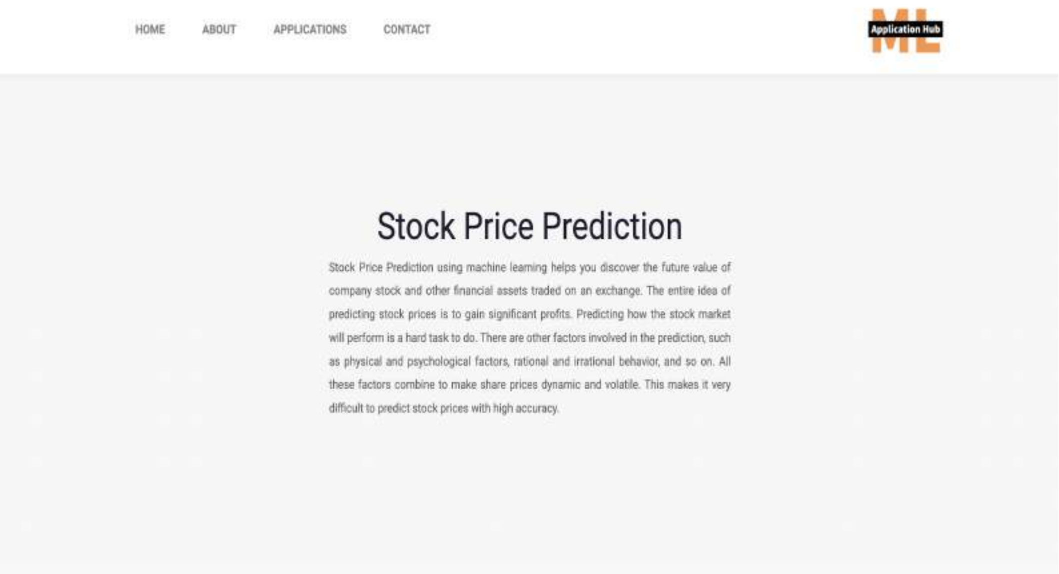
convert 20 pounds to yen

20.0 GBP is 3286.71 JPY

convert 1000rs to euro

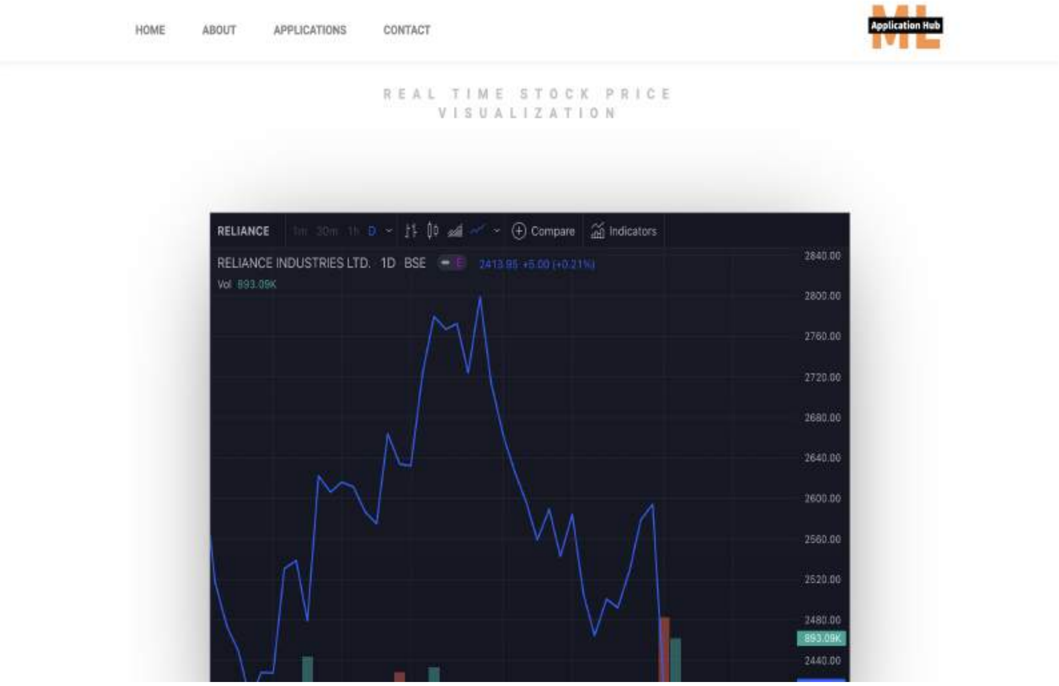
4.10.1: Currency Conversion

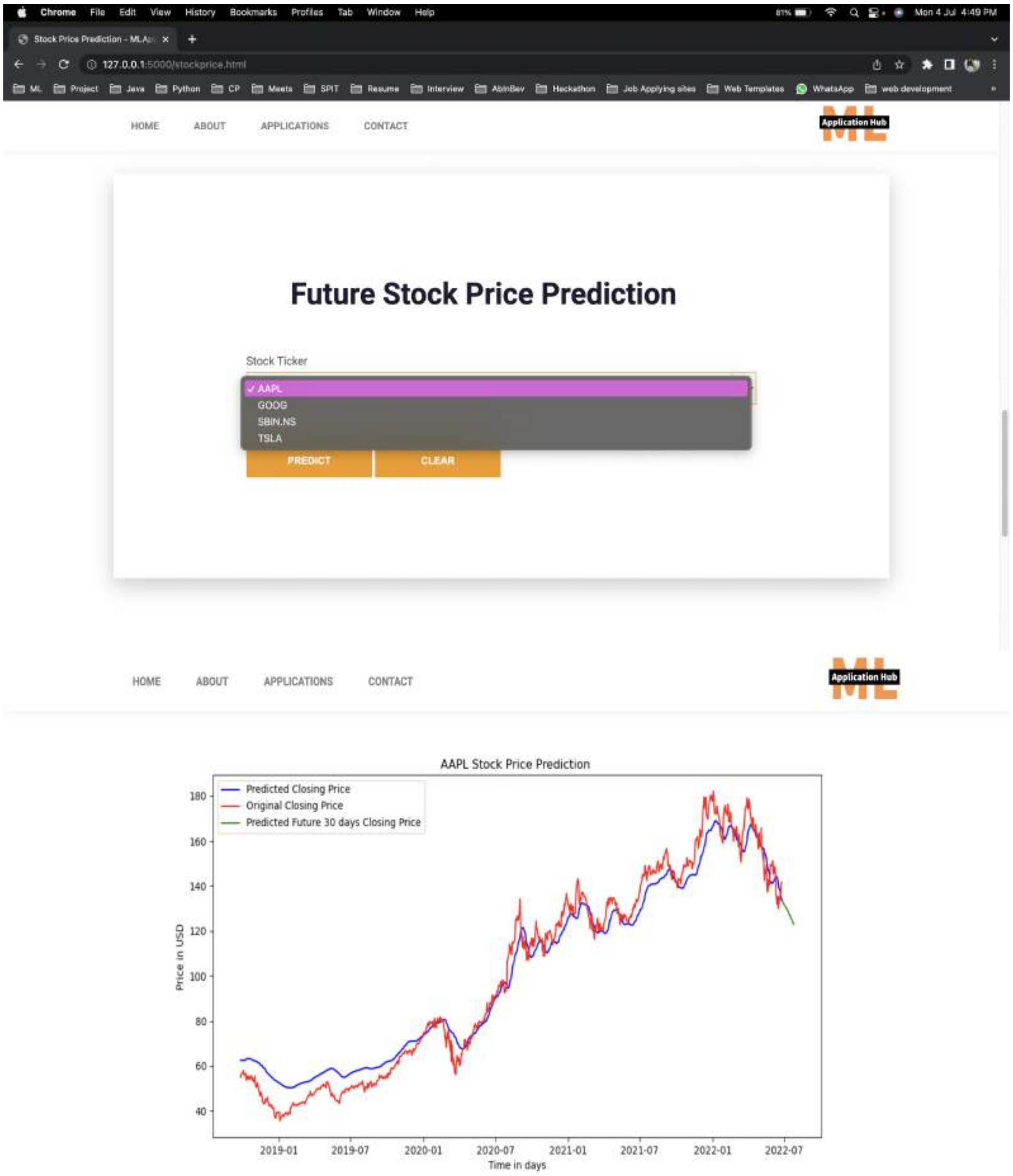
4.11 Deep Learning - Stock Price Prediction



Stock Price Prediction

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.





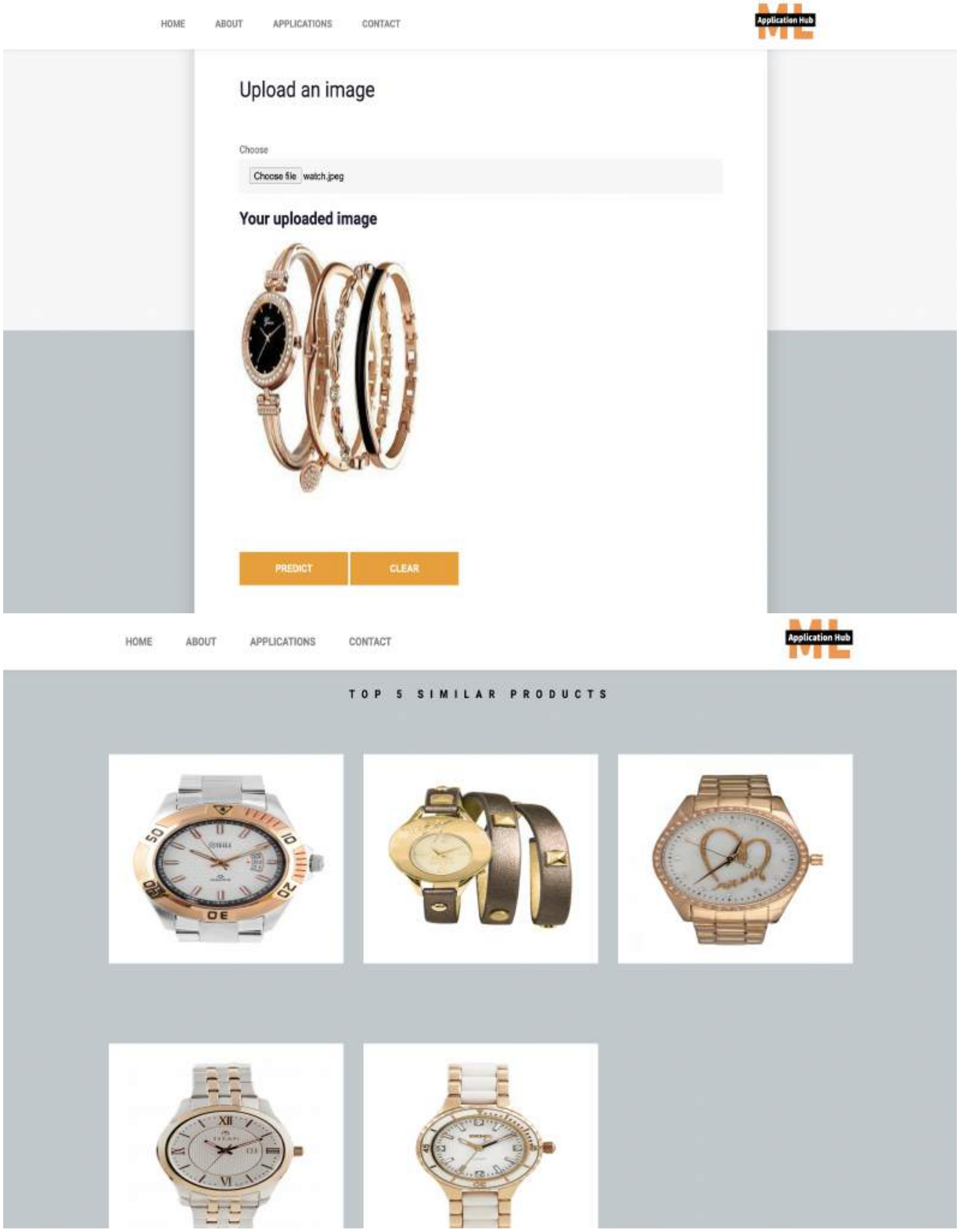
Future Stock Price Prediction

Stock Ticker

✓ AAPL
GOOG
SBIN.NS
TSLA

4.11.1: Stock Price Prediction

4.12 Deep Learning - Fashion Reverse Image Search Engine



Upload an image

Choose

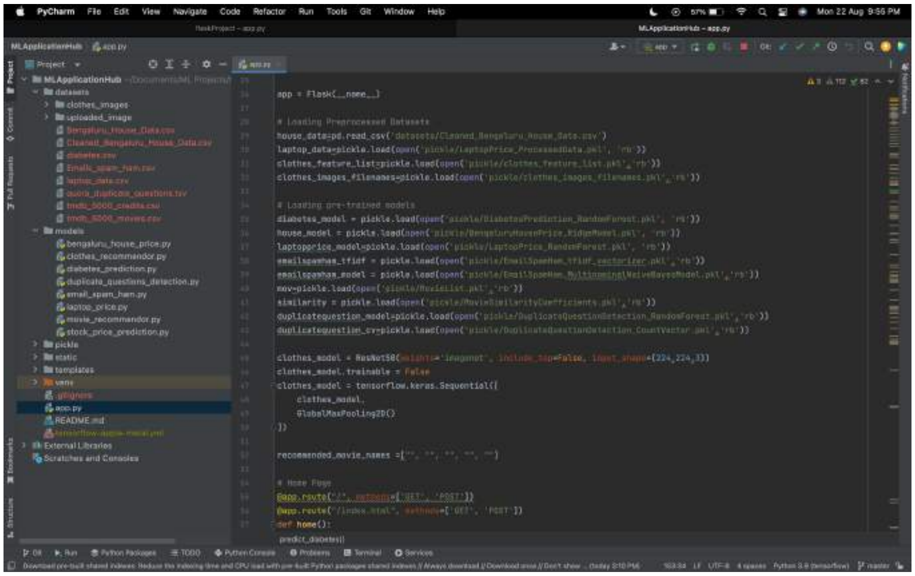
Choose file watch.jpeg

Your uploaded image

TOP 5 SIMILAR PRODUCTS

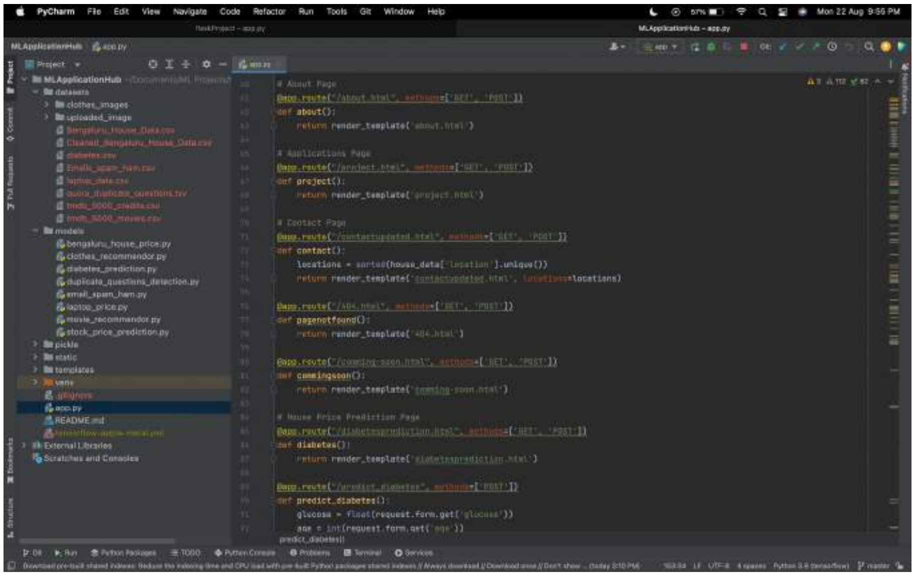
4.12.1: Fashion Reverse Image Search Engine

4.13 Code 1



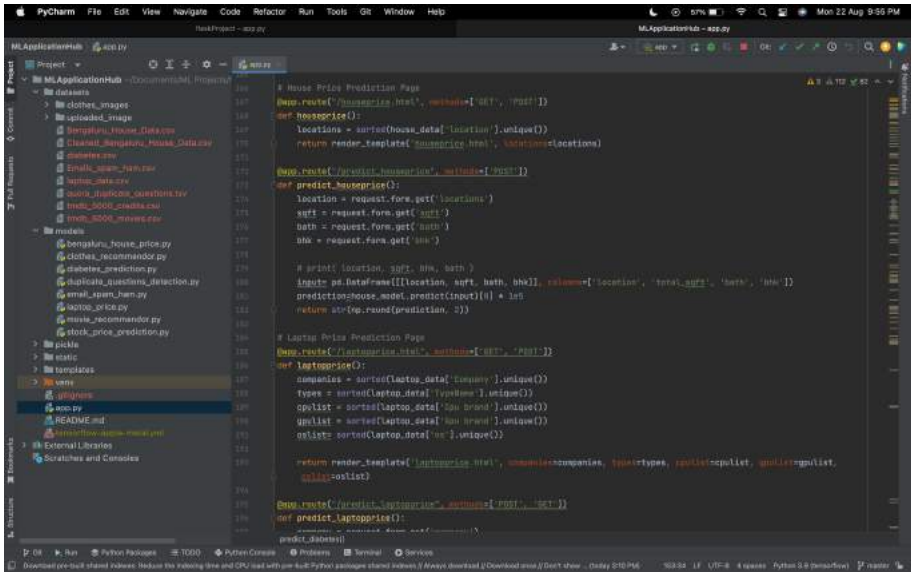
```
app = Flask(__name__)

# Loading Preprocessed Datasets
house_data=pd.read_csv('datasets/Cleaned_Bengaluru_House_Data.csv')
laptop_data=pickle.load(open('pickle/LaptopPrice_ProcessedData.pkl', 'rb'))
clothes_feature_list=pickle.load(open('pickle/clothes_feature_list.pkl','rb'))
clothes_images_filenames=pickle.load(open('pickle/clothes_images_filenames.pkl','rb'))

# Loading pre-trained models
diabetes_model = pickle.load(open('pickle/DiabetesPrediction_RandomForest.pkl', 'rb'))
house_model = pickle.load(open('pickle/BengaluruHousePrice_RidgeModel.pkl', 'rb'))
laptopprice_model=pickle.load(open('pickle/LaptopPrice_RandomForest.pkl', 'rb'))
emailspamham_tfidf = pickle.load(open('pickle/EmailSpamHam_tfidf_vectorizer.pkl','rb'))
emailspamham_model = pickle.load(open('pickle/EmailSpamHam_MultinomialNaiveBayesModel.pkl','rb'))
mov=pickle.load(open('pickle/Movielist.pkl','rb'))
similarity = pickle.load(open('pickle/MovieSimilarityCoefficients.pkl','rb'))
duplicatequestion_model=pickle.load(open('pickle/DuplicateQuestionDetection_RandomForest.pkl','rb'))
duplicatequestion_cv=pickle.load(open('pickle/DuplicateQuestionDetection_CountVector.pkl','rb'))

clothes_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224,224,3))
clothes_model.trainable = False
clothes_model = tensorflow.keras.Sequential([
    clothes_model,
    GlobalMaxPooling2D()
])

recommended_movie_names =["", "", "", "", ""]

# Home Page
@app.route("/", methods=['GET', 'POST'])
@app.route("/index.html", methods=['GET', 'POST'])
def home():
```

4.14 Code 2



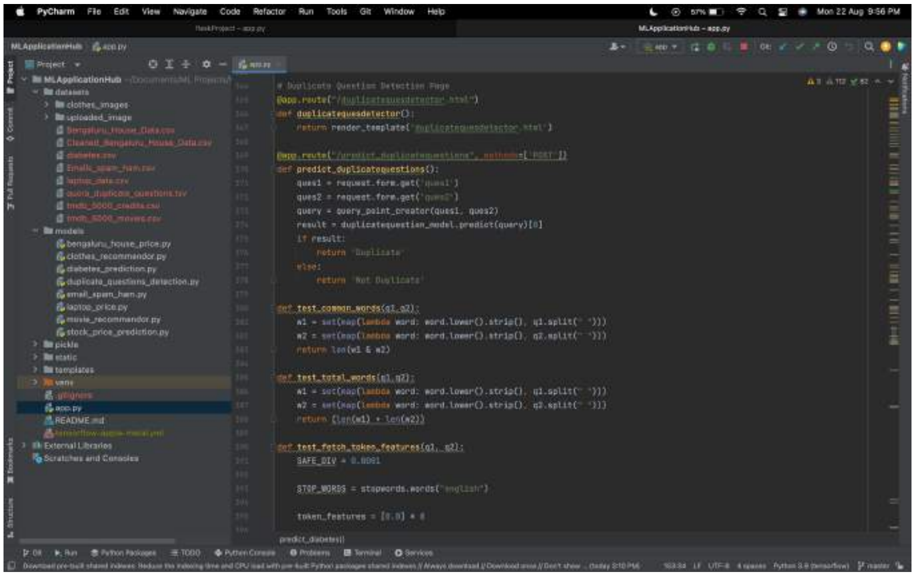
```
# About Page
@app.route("/about.html", methods=['GET', 'POST'])
def about():
    return render_template('about.html')

# Applications Page
@app.route("/project.html", methods=['GET', 'POST'])
def project():
    return render_template('project.html')

# Contact Page
@app.route("/contactupdated.html", methods=['GET', 'POST'])
def contact():
    locations = sorted(house_data['location'].unique())
    return render_template('contactupdated.html', locations=locations)

@app.route("/404.html", methods=['GET', 'POST'])
def pagenotfound():
    return render_template('404.html')

@app.route("/coming-soon.html", methods=['GET', 'POST'])
def commingsoon():
    return render_template('coming-soon.html')

# House Price Prediction Page
@app.route("/diabetesprediction.html", methods=['GET', 'POST'])
def diabetes():
    return render_template('diabetesprediction.html')

@app.route("/predict_diabetes", methods=['POST'])
def predict_diabetes():
    glucose = float(request.form.get('glucose'))
    age = int(request.form.get('age'))
```

4.15 Code 3



```python
# House Price Prediction Page
@app.route("/houseprice.html", methods=['GET', 'POST'])
def houseprice():
    locations = sorted(house_data['location'].unique())
    return render_template('houseprice.html', locations=locations)

@app.route("/predict_houseprice", methods=['POST'])
def predict_houseprice():
    location = request.form.get('locations')
    sqft = request.form.get('sqft')
    bath = request.form.get('bath')
    bhk = request.form.get('bhk')

    # print( location, sqft, bhk, bath )
    input= pd.DataFrame([[location, sqft, bath, bhk]], columns=['location', 'total_sqft', 'bath', 'bhk'])
    prediction=house_model.predict(input)[0] * 1e5
    return str(np.round(prediction, 2))

# Laptop Price Prediction Page
@app.route("/laptopprice.html", methods=['GET', 'POST'])
def laptopprice():
    companies = sorted(laptop_data['Company'].unique())
    types = sorted(laptop_data['TypeName'].unique())
    cpulist = sorted(laptop_data['Cpu brand'].unique())
    gpulist = sorted(laptop_data['Gpu brand'].unique())
    oslist= sorted(laptop_data['os'].unique())

    return render_template('laptopprice.html', companies=companies, types=types, cpulist=cpulist, gpulist=gpulist,
     oslist=oslist)

@app.route("/predict_laptopprice", methods=['POST', 'GET'])
def predict_laptopprice():
```

4.16 Code 4



```python
# Duplicate Question Detection Page
@app.route("/duplicatequesdetector.html")
def duplicatequesdetector():
    return render_template('duplicatequesdetector.html')

@app.route("/predict_duplicatequestions", methods=['POST'])
def predict_duplicatequestions():
    ques1 = request.form.get('ques1')
    ques2 = request.form.get('ques2')
    query = query_point_creator(ques1, ques2)
    result = duplicatequestion_model.predict(query)[0]
    if result:
        return 'Duplicate'
    else:
        return 'Not Duplicate'

def test_common_words(q1,q2):
    w1 = set(map(lambda word: word.lower().strip(), q1.split(" ")))
    w2 = set(map(lambda word: word.lower().strip(), q2.split(" ")))
    return len(w1 & w2)

def test_total_words(q1,q2):
    w1 = set(map(lambda word: word.lower().strip(), q1.split(" ")))
    w2 = set(map(lambda word: word.lower().strip(), q2.split(" ")))
    return (len(w1) + len(w2))

def test_fetch_token_features(q1, q2):
    SAFE_DIV = 0.0001

    STOP_WORDS = stopwords.words("english")

    token_features = [0.0] * 8
```

5 Test Cases

Table 5.1: Test Case - Indian PIMA Diabetes Classification

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
TC01	Validate PIMA Diabetes Input Parameters	Validates the age parameter on the client side before submiDng the form.	Doesn't lets the user submit form and shows error message - Characters not allowed.	Lets the user submit form without showing any error message.	Fail
TC02	Validate PIMA Diabetes Input Parameters	The gender radio buMons lets the user click only one of the op0ons and it is a required field.	The user is required to select the one of the op0ons from gender category and can't submit the fro without filling this field.	As Excepted	Pass
TC03	Validate PIMA Diabetes Input Parameters	The pregnancy field is disabled if the gender is Male and the default value is 0.	The male users can't modify the no. of pregnancies field.	As Excepted	Pass

6 Limitations

- Certain modules like Indian PIMA Diabetes Classification, takes into consideration input parameters which might require laboratory analysis. This module maybe well utilised by laboratory researchers rather than layman users.
- The stock price prediction module is built using LSTM module, which takes approximately fifteen minutes to run the epochs. Hence, to choose the current dates and show real-time prediction delays the response time of the application.
- The house price prediction module is restricted to regions of Bengaluru, India since the dataset comprises attributes of those areas.
- The reverse image search engine gives efficient output based on how huge and varied the dataset is. If an uploaded image differs immensely from the dataset, then the output suggested by the model may seems to be irrelevant.

7 Future Enhancements

- **Visualisation Feature** - The visualisation feature will demonstrate graphically how different independent features of the dataset affect the dependent target value and represents the correlation among the attributes.
- **Code Visibility** - The project will have a view code module to outline the detailed step by step process of model building. This module will provide a learning pathway for the individuals keen to venture in the world of data science and learn about the ML algorithms along with their applications.
- **Improvising the Body Posture Detection Module** - Currently this module identifies the 17 body points of the human in the video stream and outline his sketch. In future, this module can be extended to become an intelligent yoga posture detector which analysis whether a human is doing yoga posture correctly or not.

7.1 Web References

[1.] Deploy Machine Learning Model using Flask
Produced by : Krish Naik

[2.] Real-time Human Pose Estimation in the Browser with TensorFlow.js
Author : Dan Oved

[3.] Deep learning based reverse image search for industrial applications
Author : Vegard Flovik

[4.] Stock Market Predictions with LSTM in Python
Author : Thushan Ganegedara

[5.] Decision Tree Algorithm – A Complete Guide
Author : Anshul Saini

[6.] Dialogflow Documentation
Maintained by : Team Google

[7.] Ultimate Guide to Understand and Implement Natural Language Processing
Author : Shivam Bansal

[8.] Comprehensive Guide to build a Recommendation Engine from scratch
Author : Pulkit Sharma