**Accurate Prediction of Diseases based on Symptoms of Patients using Machine Learning Algorithms**

**A Project Report**

***Submitted by***

# YASHWANTH S – 20191CSE0710

|  |
| --- |
| **ABRAR HUSSAIN DAR – 20191CSE0760** |
| **MALLARAPU VAISHNAVI – 20191CSE0746** |
| **AKSHAY N – 20191CSE0732** |

### *Under the guidance of*

**Mr. MOHAN KUMAR A V,**

**ASSISTANT PROFESSOR**

***in partial fulfilment for the award of the degree of***

# BACHELOR OF TECHNOLOGY

### IN

**COMPUTER SCIENCE AND ENGINEERING**

**At**



**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

**PRESIDENCY UNIVERSITY**

**BENGALURU**

**MAY-2023**

**DEPARTMENT OF COMPUTER SCIENCE**

**AND ENGINEERING**

**School of Computer Science and Engineering & Information Science**

**PRESIDENCY UNIVERSITY**

**CERTIFICATE**

This is to certify that the Project report **“Accurate Prediction of Diseases based on Symptoms of Patients using Machine Learning Algorithms”** being submitted by “Yaswanth S, Abrar Hussain Dar, Mallarapu Vaishnavi, Akshay N” bearing roll number(s) “20191CSE0710, 20191CSE0760, 20191CSE0746, 20191CSE0732” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

|  |  |
| --- | --- |
| **Mr Mohan Kumar A V**  Assistant Professor  School of CSE  Presidency University | **Dr. Pallavi R**  HOD  School of CSE  Presidency University |

|  |  |
| --- | --- |
| **Dr. C. KALAIARASAN**  Associate Dean  School of CSE&IS  Presidency University | **Dr. MD. SAMEERUDDIN KHAN**  Dean  School of CSE&IS  Presidency University |

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **“Accurate Prediction of Diseases based on Symptoms of Patients using Machine Learning Algorithms”** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Mr Mohan Kumar A V,** Assistant Professor**,** **School of Computer Science & Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

|  |  |  |
| --- | --- | --- |
| **Name(s)** Mr. YASHWANTH SMr. ABRAR HUSSAIN DARMs. MALLARAPU VAISHNAVIMr. AKSHAY N | **Roll No(s)** 20191CSE071020191CSE076020191CSE074620191CSE0732 | **Signature(s)** |

**ABSTRACT**

Accurate disease prediction plays a pivotal role in healthcare, enabling timely diagnosis and effective treatment. Traditional diagnostic methods based on symptom analysis are limited by human capacity and often result in misdiagnosis or delayed intervention. Leveraging the power of machine learning algorithms, this project aims to develop a system for accurate prediction of diseases based on patients' symptoms, contributing to improved healthcare outcomes.

Through a comprehensive literature review, previous studies on disease prediction using machine learning algorithms were examined, identifying gaps and limitations in existing research. A dataset comprising medical records, symptom descriptions, and patient information was collected, adhering to privacy and ethical guidelines.

The collected data underwent pre-processing, including cleaning, handling missing values, and normalising features to ensure data quality and uniformity. Feature selection and extraction techniques, such as principal component analysis and correlation analysis, were employed to identify the most informative features for disease prediction.

Several machine learning algorithms, including decision trees, random forests, support vector machines, and deep learning models, were implemented and trained on the pre-processed dataset. Model performance was evaluated using appropriate metrics such as accuracy, precision, recall, and F1 score.

The results showcased the effectiveness of machine learning algorithms in accurately predicting diseases based on symptoms. Comparative analysis of the algorithms revealed their respective strengths and weaknesses, aiding in the selection of the most suitable algorithm for disease prediction.

The developed system for accurate disease prediction based on symptoms holds the potential to revolutionise healthcare. By enabling early detection, personalised treatment plans, and efficient resource allocation, the system can significantly improve patient outcomes, reduce healthcare costs, and alleviate the burden on healthcare professionals.

This project contributes to the advancement of accurate disease prediction by harnessing the capabilities of machine learning and combining them with domain expertise. The findings provide valuable insights into the potential of machine learning algorithms in healthcare, paving the way for further research and the development of intelligent decision support systems for disease diagnosis.

In conclusion, accurate disease prediction based on symptoms using machine learning algorithms presents a promising approach to enhance healthcare delivery. By leveraging the power of data and machine learning, this project aims to improve disease diagnosis, ultimately leading to better patient care and improved healthcare outcomes.

**ACKNOWLEDGEMENT**

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science & Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Dean **Dr. C. Kalaiarasan,** Professor **Dr. T K Thivakaran,** University Project-II In-charge, School of Computer Science & Engineering, Presidency University for rendering timely help for the successful completion of this project.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Co-Ordinators **Mr. Mrutyunjaya MS, Mr. Sanjeev P Kaulgud, Mr. Rama Krishna K and Dr. Madhusudhan MV**.

We are greatly indebted to our guide **Mr Mohan Kumar A V**, School of Computer Science & Engineering, Presidency University for his inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Yashwanth S**

**Abrar Hussain Dar**

**Mallarapu Vaishnavi**

**Akshay N**

**List of Tables**

| **SL No.** | **Table Name** | **Table Caption** | **Page No.** |
| --- | --- | --- | --- |
| 1 | Table 2.2 | Literature Review | 13-23 |
| 2 | Table 7.2.1 | Use Case Diagram | 33 |
| 3 | Table 9.2 | Test Cases | 52 |
| 4 | Table 9.2 | Test Case Model Building | 52 |

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No.** | **Figure Name** | **Caption** | **Page No.** |
| 1 | Fig 6.2.1 | Waterfall model | 28 |
| 2 | Fig 7.2.2 | Use case Diagram | 34 |
| 3 | Fig 7.3.1 | Class Diagram | 34 |
| 4 | Fig 7.4.1 | Sequence Diagram | 35 |
| 5 | Fig 7.5.1 | Collaboration Diagram | 35 |
| 7 | Fig 7. 6.1 | Deployment Diagram | 36 |
| 8 | Fig 7.7.1 | Activity Diagram | 36 |
| 9 | Fig 7.8.1 | Component Diagram | 37 |
| 10 | Fig 7.9.1 | ER Diagram | 38 |
| 11 | Fig 7.10.1 | DFD Diagram | 39 |
| 12 | Fig 7.10.2 | DFD Diagram | 39 |
| 13 | Fig 7.10.3 | DFD Diagram | 40 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT & ACKNOWLEDGMENT** | **4-6** |
| **1** | **INTRODUCTION** | **11** |
| **2** | **LITERATURE SURVEY** | **12** |
|  | 2.1 Existing Methods and Disadvantages | 12-13 |
|  | 2.2 Literature Review | 13-23 |
| **3** | **PROPOSED METHOD** | **24** |
| **4** | **OBJECTIVES** | **25-26** |
| **5** | **METHODOLOGY** | **27** |
| **6** | **SYSTEM DESIGN** | **28** |
|  | 6.1 Introduction | 28 |
|  | 6.2 Software Development Life Cycles | 28-29 |
|  | 6.3 Feasibility Study | 29-30 |
| **7** | **DETAILED DESIGN** | **31-32** |
|  | 7.1 UML Diagram | 33 |
|  | 7.2 Use Case Diagram | 33 |
|  | 7.3 Class Diagram | 34 |
|  | 7.4 Sequence Diagram | 34 |
|  | 7.5 Collaboration Diagram | 35 |
|  | 7.6 Deployment Diagram | 35 |
|  | 7.7 Activity Diagram | 36 |
|  | 7.8 Component Diagram | 37 |
|  | 7.9 ER Diagram | 37 |
|  | 7.10 DFD Diagram | 38-40 |
| **8** | **IMPLEMENTATION** | **41** |
|  | 8.1 Algorithm Description | 41 |
|  | 8.2 Source Code Description | 42-44 |
|  | 8.3 Data Dictionary | 45 |
|  | 8.4 Project Insights | 46-50 |
| **9** | **SOFTWARE TESTING** | **51** |
|  | 9.1 Testing Objectives | 51-52 |
|  | 9.2 Test Cases | 52 |
| **10** | **SOFTWARE REQUIREMENTS SPECIFICATIONS** | **53** |
|  | 10.1 Hardware Requirements | 53 |
|  | 10.2 Software Requirements | 54-55 |
| **11** | **TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)** | **56** |
| **12** | **OUTCOMES** | **57** |
| **13** | **RESULTS AND DISCUSSIONS** | **58-59** |
| **14** | **CONCLUSION** | **60** |
|  | **REFERENCES** | **61** |

**CHAPTER-1**

**INTRODUCTION**

Accurate disease prediction plays a crucial role in healthcare by enabling timely diagnosis, effective treatment, and improved patient outcomes. Traditionally, medical professionals rely on their expertise and knowledge to identify diseases based on patients' symptoms. However, the human capacity to accurately diagnose complex diseases solely through symptom analysis is limited, often leading to misdiagnosis or delayed treatment. This is where the power of machine learning algorithms comes into play.

Machine learning, a subfield of artificial intelligence, has gained significant attention in healthcare due to its potential to analyse vast amounts of medical data and uncover patterns that may not be readily apparent to human observers. By leveraging machine learning algorithms, we can develop systems that accurately predict diseases based on patients' symptoms, contributing to early detection, personalised medicine, and improved patient care.

The importance of accurate disease prediction lies in its ability to provide timely intervention, preventing disease progression and reducing healthcare costs. By accurately identifying diseases at an early stage, medical professionals can initiate appropriate treatment plans, leading to better patient outcomes and potentially saving lives. Moreover, accurate disease prediction can aid in the efficient allocation of healthcare resources, optimising the utilisation of diagnostic tests, medical equipment, and healthcare personnel.

Additionally, accurate disease prediction using machine learning can aid in the identification of rare or complex diseases that may have elusive symptom patterns. By analysing large-scale datasets containing diverse patient information, such as symptoms, medical history, and genetic factors, machine learning algorithms can uncover hidden correlations and identify disease patterns that may not be evident through traditional diagnostic methods alone.

Furthermore, accurate disease prediction systems have the potential to reduce the burden on healthcare professionals by providing decision support tools. By automating the process of disease prediction, these systems can assist healthcare providers in making informed decisions, thereby improving efficiency and reducing diagnostic errors.

In this project, we aim to develop a system for accurate prediction of diseases based on symptoms using machine learning algorithms. By leveraging the power of machine learning and analysing comprehensive datasets, we strive to improve the accuracy and efficiency of disease diagnosis, contributing to better healthcare outcomes and enhancing the overall quality of patient care. By harnessing the capabilities of machine learning and combining them with the expertise of medical professionals, we can unlock new possibilities for accurate disease prediction, revolutionising the way healthcare is delivered and positively impacting the lives of countless individuals.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 Existing Method:**

Since the advent of powerful computing, surgeons still need the technology in this application in a variety of ways, such as surgical depiction and x-ray photography, although the technology has perceptually lagged behind. Due to other aspects including weather, atmosphere, blood pressure, and several other parameters, the approach still requires the doctor's knowledge and expertise. Numerous variables are acknowledged as being necessary for understanding the entire working process, but no model has been able to analyse them properly. Medical decision support systems must be employed to address this problem. The doctors can use this technique to help them choose wisely.

**Disadvantages:**

* **Low Accuracy**: Disease prediction solely based on symptoms can sometimes result in low accuracy. Symptoms can overlap across different diseases, making it challenging to accurately predict a specific disease based solely on symptom data. Additionally, variations in symptom presentation among individuals can further reduce the accuracy of predictions.
* **High Complexity**: Implementing machine learning algorithms for disease prediction can be complex. It involves data pre-processing, feature extraction, algorithm selection, hyperparameter tuning, and model evaluation. Dealing with complex algorithms and large datasets can increase the overall complexity of the system.
* **Highly Inefficient**: Training machine learning models for disease prediction can be computationally expensive and time-consuming. Depending on the size of the dataset and the complexity of the chosen algorithms, the training process can require significant computational resources and may take a long time to complete. This inefficiency can limit scalability and real-time prediction capabilities.
* **Requires Skilled Personnel**: Developing and deploying a disease prediction system based on machine learning algorithms requires expertise in both data science and software development. Skilled personnel with knowledge of machine learning algorithms, data pre-processing techniques, and software engineering practices are needed to design, implement, and maintain the system effectively.
* **Limited by Data Availability**: The accuracy and effectiveness of disease prediction models heavily rely on the availability and quality of the training data. Limited or bi-assed data can impact the performance of the algorithms and lead to suboptimal predictions. Access to diverse and representative datasets is crucial for building robust and generalizable disease prediction models.
* **Interpretability Challenges**: Some machine learning algorithms, such as deep learning models, are known for their black-box nature, making it difficult to interpret the underlying decision-making process. Interpreting and explaining the predictions made by these models can be challenging, which may raise concerns in medical settings where interpretability and transparency are important.

It is essential to consider these disadvantages and address them appropriately during the development and evaluation of a disease prediction system. Mitigating these challenges requires careful algorithm selection, data pre-processing techniques, continuous model refinement, and the involvement of domain experts to improve accuracy, efficiency, and usability of the system.

**2.2 Literature Review:**

Table 2.2 : Literature Review

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. No. | Paper Title | Method | Advantages | Limitations |
| [1] | Predicting High-Risk Prostate Cancer Using Machine Learning Methods | We found appropriate data imbalance methods and scaling methods by assessing their efficiency after the first pre-processing, which comprised managing missing values and figuring out the rate of change on the PoPC-labelled dataset. Then, these methods were used on data that had PoHRPC labels.  Processing of Data  Build classifiers Evaluation of Classifiers Evaluation of the Predictability of Features | There are presented classifier assessments. Which use 25% of data in Holdout techniques for testing, while 75% was used to train the classifiers. The accuracy and AUC of the performance classifiers differ. Given that ADABoost got the best AUC in holdout and was only 0.002 off, it was the best algorithm for this dataset. Best cross-validation method. It was the only algorithm with a better AUC, and it outperformed decision tree in both cases in terms of accuracy, and it was only 0.076 off from the highest accuracy. Therefore, for the remaining predictions on PoPC-labelled data, ADABoost is the machine learning technique employed for this model. | Due to the low PPV values of the models, it is possible to look at the selection of an imbalance correction method further. Using different approaches, different variations of attributes including sensitivity, specificity, NPV, and PPV are tested to help determine the impact of imbalance correction. Additionally, we need to decrease the rate of false positives that currently occurs when we apply imbalance correction methods. Testing should be done using a variety of techniques, including those found in the most recent research. Ebenuwa is one instance of such a technique. Variance ranking attributes selection method proposed by et al. |
| [2] | Prediction of Prostate Cancer using Machine Learning Algorithms | The main objectives of The goals of this inquiry are to make recommendations for a process that will result in the best machine learning algorithm for prostatic cancer disease prediction. Different machine learning algorithms have been taken into consideration, and various performance measures have been compared.   * Transformation and Pre-processing of Selection * Assessment of Performance | Based on the data for each patient's unique characteristics, we applied machine learning algorithms on a dataset of prostate cancer patients to forecast which individuals will have fatal prostate cancer and which people will not be disabled. Our goal was to consider several layout models and choose the most effective one. Five calculations—the K-Nearest Neighbour, Support Vector Machines, Logistic Regression, Naive Bayes, and Random Forest—were used to create our analysis. | In conclusion, the employment of information mining systems for prescient assessment is important in the healthcare area since it allows us to identify illnesses early on and, as a result, save lives via the anticipation of cure. In this study, we used a variety of learning algorithms, including KNearest Neighbour, Support Vector Machines, Logistic Regression, Naive Bayes, and Random Forest, to predict patients who will experience prostate cancer and those who will not. The re-enactment outcomes were shown the Logistic Regression and RandomForest classifiers showed off their prowess in predicting with the highest results in terms of precision and least amount of execution time. |
| [3] | Early Detection of Breast Cancer Using Machine Learning Techniques | * Artificial Neural Network (ANN) * Support Vector Machine (SVM) * K-Nearest Neighbors (KNN) * Decision Tree (DT) * Random Forest (RF) Algorithm * AdaBoost Classifier * Naïve Bayes (NB) Classifier | Most researchers, according to Figure 2, have focused on mammography pictures since they are faster and safer than other methods of detecting breast cancer. Figure 3 compares the algorithms and ML techniques used in the evaluated literature listed in Table 1 for the identification of breast cancer. SVM is shown to be the approach that is employed the most. Figure 4 illustrates the outcomes of ML-based breast cancer detection. | In the current paper, breast cancer and ML were introduced, and a thorough literature evaluation of the current ML techniques for breast cancer diagnosis was conducted. The results according to these researchers, SVM is the most often applied technique for cancer detection applications. To increase performance, SVM was applied either independently or in conjunction with another technique. SVM (single or hybrid) accuracy can be enhanced to a maximum of 100%, with a maximum accuracy attained of 99.8%. |
| [4] | Prediction of Lung Cancer Using Machine Learning Classifier | * Neural Network * Radial Basis Function Network * Support Vector Classifier * Logistic Regression Classifier * Random Forest Classifier * J48 Classifier * Naïve Bayes Classifier * Knn Classifier | Weka is an open source programme used for data visualisation, regression, clustering, and classification. Weka typically accepts input files with the.csv or.arff extension.  -Uploading of the input data into the Weka tool is enabled when data preposing is selected.  Data analysis is made simple by Weka's straightforward understanding and representation of the data.  -According to the analysis, RBF is the classifier of choice when compared to other classifiers.  This is because it has the best classification accuracy, with 26 of its 32 instances properly categorised and only 6 wrongly classified.  -This study demonstrates that the accuracy of the Radial Basis Function Network (RBF) classifier on data related to lung cancer is 81.25%. Therefore, it can be inferred from the analysis that a modified functional approach in RBF and a suitable feature selection method will result in improved learning outcomes. | Accuracy is to be improved |
| [5] | Machine Learning Algorithms for the  Prediction of Central Lymph Node  Metastasis in Patients with Papillary  Thyroid Cancer | * Patients * Surgical method * Clinical traits and ultrasonographic characteristics * creation of models based on ML * Selection of features and the validation strategy * Analytical Statistics | -In this study, For the first time, ML-based models for the prediction of CLNM in PTC patients were developed.  These ML-based models demonstrated outstanding predictive accuracy and clinical value by ROC analysis and DCA by including clinical variables and US features.  -Second, utilising a feature selection technique in addition to traditional multivariate analysis, we were able to rank three carefully chosen ML-based models using the mean to identify risk variables for CLNM.  These variables' predictive significance was demonstrated by their mean ranks.  Third, the GBDT model with 7 variables was the most effective ML-based model for the prediction of CLNM in PTC patients, according to our study. | It is possible to predict CLNM in PTC patients by combining clinical traits and US findings. All ML-based models outperformed the US in terms of CLNM prediction.  The best model was determined via ROC analysis and DCA to be the 7-variable GBDT model.  Younger age, male sex, low serum TPO-Ab, and US features such as presumed racial bias were found by multivariate analysis and feature selection.  Important risk factors for CLNM were LNs, microcalcifications, and tumour size > 1.1 cm.  In order to anticipate lymph node metastases in PTC, ML algorithms can be helpful.  \*Accuracy  \*An online application of the GBDT model based on the clinical traits and US features should be developed in order to allow surgeons and patients at other facilities to profit from this study. |
| [6] | Lung Cancer Incidence Prediction Using Machine Learning Algorithms | The Algorithms that are used to predict data:   * Support Vector Regression (SVR) * Long Short Term Memory (LSTM) Network * Backpropagation Neural Network * Radial Basis Function * Linear Regression etc.   Back propagation learning algorithm, for example  Backpropagation is a multi-layer employing the gradient descent process, the perceptron learning algorithm adjusts each neuron's weight.  The process begins by feeding inputs into the network and determining the total potential of the following hidden layer using the appropriate weights. Initial weights are often assigned at random.  A network of long-short term memories.  The effective LSTM variant of the recurrent network is frequently employed for classification and prediction issues.  Support Vector Regression (SVR)  Support Vector Regression is a subset of Support Vector Machines that accepts outputs in the form of actual values rather than binary integers. | As the difference between the anticipated and actual data would be smaller, the increase in training data would enhance the models' capacity to make predictions.  The findings demonstrate that machine learning systems are exceptionally successful at anticipating cancer without training incidence rates, which might be used to predict future rates and increase public awareness of various cancer forms.  However, Support Vector Regression generated the best prediction results with the least error and the highest forecast accuracy. All approaches produced encouraging results. When the other two algorithms were considered, Backpropagation and LSTM were respectively the next two methods. | * Worldwide, the incidence and mortality rate of cancer are extremely high. Because there aren't enough records, reliable and consistent data isn't provided. * One of the most difficult machine learning jobs is the prediction of this type of data, thus appropriate algorithms should be chosen to complete the task. * It will be taken into account to install new machine learning algorithms for the prediction of more cancer types throughout all of Europe in order to study the incidence rates for age groups. Age groups will be used to categorise these predictions. * To forecast both incidence and patient mortality rates, mortality rates will be taken into account. |
| [7] | Breast Cancer Prediction Using Machine Learning Algorithms | * Data Pre-processing * k-Nearest Neighbours Method * k-Nearest Neighbour algorithm | -For patients, the prediction and conclusion of breast cancer are especially helpful. When describing the dataset of Wisconsin Breast cancer patients, arrangement algorithms were taken into consideration for evaluating their grouping execution when it comes to accuracy, precision, sensitivity, and specificity. The restorative specialists can be effectively assisted by these tactics in fundamental leadership across the board.  -This study intends to detect cancer in its early stages, thereby somewhat reducing the death rate, by examining several symptoms and determining whether the tumour is benign or malignant.  They have employed a classification system that swiftly and accurately classifies the cancer photos. | * Accuracy * It is possible to conduct additional tests on more real-world systems to assess the efficacy and efficiency of various categorization algorithms when used in various fields.   .  . |
| [8] | Machine Learning Algorithms as a Computer-assisted Decision Tool for Oral Cancer Prognosis  and Management Decisions | * Decision trees * Random forests * Support vector machines (SVMs) * Artificial neural networks (ANNs) * Gradient boosting   Logistic regression | Continuous learning capability: Machine learning algorithms may learn from new data as it becomes available and have the capacity to keep learning and perform better. This implies that ML models can be continuously upgraded to increase their accuracy and performance as new research is undertaken and additional data is gathered.  Research promise: By enhancing cancer prognosis prediction and management, the use of ML in OCSCC research has the potential to considerably improve the area. Researchers can discover patterns in massive datasets that may not be visible using conventional statistical methods by utilising the capabilities of machine learning.  Algorithm democratisation: ML research can contribute to the democratisation of algorithmic application and increase the accessibility of algorithms for researchers and medical professionals worldwide. Researchers may assist in lowering entry barriers and opening up advanced analytics to a larger audience by creating open-source ML tools and sharing knowledge and skills. | * ML has the potential to make substantial strides in OCSCC research. * The capacity of ML models to continue training constantly as new data become available is a benefit of using and training them. * Future ML research will enable us to democratise and enhance the application of algorithms to better forecast the prognosis of cancer and its management globally.   . |
| [9] | Deep learning based survival prediction of oral cancer patients. | Participants in the study (data collection): From January 2000 to November 2018, the department's patient medical records were reviewed for patients who underwent surgical treatment for oral SCC. After removing individuals who met specific criteria, the records of 255 patients were examined.  Analysis of statistical data: The Mann-Whitney U test, Chi-square test, Fisher's exact test, Cochran-Armitage Trend test, Kaplan-Meier technique, and univariate and multiple CPH regression analysis were all carried out using the R programming language. Significant was assessed to be p<0.05.  t. | To create a model predicting postoperative survival, the study examined the medical data of 255 individuals who had undergone surgical therapy for oral SCC. To create the prediction model, they combined Deep Learning-Based Survival Analysis and Random Survival Forest. According to the study, when it came to predicting postoperative survival in patients with oral SCC, the Deep Learning-Based Survival Analysis method performed better than the Random Survival Forest method. | The retrospective design of the study and the removal of some patients with incomplete data or specific medical histories are just a few of the Limitations of the study that should be considered when interpreting the results. Furthermore, the study was carried out at a single institution, which would restrict how broadly the results can be applied to different groups. |
| [10] | Thyroid Diseases Treatment Prediction with Machine Learning Approaches | * Data collection * The purposed featured model * Classifiers * Validation | The thyroid is referred to as the "powerhouse" of our body because if something goes wrong with this gland, the entire body would suffer. As a result, making an early diagnosis of a potential malfunction is essential, and predicting how to treat a patient with hypothyroidism can be very helpful for clinicians who are currently treating patients.  This approach aims to create a machine learning based decision assistance system for endocrinologists caring for thyroid disease patients. These techniques are becoming more popular in medicine, and our work can be very beneficial because of the excellent diagnostic accuracy we were able to accomplish in the particular clinical setting.  The doctor is helped in selecting the amount of the drug to prescribe since the proposed model is able to forecast the patient's treatment progress based on other factors relating to the individual being treated.  In the future, it will be necessary to substantially broaden the set of data and features taken into account in order to more broadly generalise our findings with additional data, the training process is likely to result in classifiers that are more efficient and allow for a more accurate estimation of the demonstrated performance.  In order to determine whether there is a specific additional thyroid condition that can affect hypothyroidism, it is also possible to evaluate whether any secondary thyroid diseases connected to the patient are present. | * accuracy is to be improved. |
| [11] | Disease Prediction using Machine Learning | * KNN * NAIVE BAYES * LOGISTIC REGRESSION | It is essential to do research and create a system that would allow people to predict chronic diseases without consulting a doctor or other healthcare provider. employing various machine learning modelling techniques, to study patient symptoms in order to diagnose various diseases. Text data and structured data handling are improper approaches. The suggested approach would take into account both structured and unstructured data. The precision of predictions will increase thanks to machine learning. | * This study aims to identify an illness from its symptoms. The project is set up so that the system either forecasts disease or gets user symptoms as input and output. * In conclusion, the variety aspect of the hospital data affects the accuracy of risk prediction for disease risk modelling. |

**Chapter-3**

**PROPOSED METHOD**

A: **K-Nearest Neighbors (KNN):**

KNN is a supervised learning technique used for cancer classification and diagnosis. With this technique, new data is fed to a computer that has been educated in a certain field.

B: **Decision Tree (DT):**

DT is a data mining method for breast cancer early detection. A tree-based model is used to display classifications or regressions.

C: **Random Forest (RF) Algorithm:**

The RF algorithm is used when the model quality is at its highest and bias and variance problems are present.

D: **AdaBoost Classifier:**

By using classification and regression, this system forecasts the chance of developing breast cancer. It turns weak learners into strong ones by combining all weak rules into one strong rule.

E: **Naïve Bayes (NB) Classifier:**

The Bayes theorem is used by the probabilistic classifier known as naive Bayes, which also makes firm assumptions about independence. Each property is considered separately in this model to uncover any potential connections between them.

**Chapter-4**

**OBJECTIVES**

* **Develop a Disease Prediction System**: Design and develop a robust and user-friendly disease prediction system that takes patient symptoms as input and accurately predicts the corresponding diseases. The system should provide reliable and timely predictions to assist healthcare professionals in making informed decisions.
* **Improve Accuracy and Precision**: Utilise machine learning techniques and algorithms to increase the precision and accuracy of disease predictions. To improve prediction performance, experiment with various algorithms, feature engineering techniques, and ensemble approaches.
* **Handle Large and Diverse Datasets**: Develop techniques to handle large and diverse datasets comprising symptom descriptions, patient demographics, and clinical variables. Ensure that the system can efficiently process and extract relevant features from such datasets, enabling accurate disease prediction.
* **Address Missing Data and Outliers**: Put mechanisms in place to deal with outliers and missing data in the symptom dataset. To assure the accuracy and dependability of the input data, use the proper imputation techniques to fill in missing values and take into account outlier identification techniques.
* **Enhance Interpretability**: Explore methods to enhance the interpretability of the disease prediction system. Investigate feature importance techniques to identify the symptoms that contribute most significantly to disease predictions. Provide explanations or visualisations that aid healthcare professionals in understanding the underlying factors influencing the predictions.
* **Validate and Evaluate Performance**: Validate and assess the created illness prediction system thoroughly. Use pertinent evaluation metrics like accuracy, precision, recall, and F1 score to assess the performance of the models. Compare the outcomes of various ensemble techniques and machine learning algorithms to determine the optimal course of action.
* **Ensure Privacy and Ethical Considerations**: Adhere to privacy regulations and ethical guidelines when handling patient data. Implement appropriate security measures to protect sensitive information. Obtain necessary consents and approvals to use patient data for research purposes while ensuring confidentiality and anonymity.
* **Create a User-Friendly Interface**: Create a user-friendly and intuitive user interface for the disease prediction system. Ensure that healthcare professionals can easily input patient symptoms and receive accurate predictions. Consider usability, accessibility, and visualisations that aid in understanding and interpreting the prediction results.
* **Document and Report Findings**: Document the development process, including data pre-processing techniques, machine learning algorithms utilised, and key design decisions. Prepare a comprehensive report summarising the project, methodologies, results, limitations, and recommendations for future enhancements.
* **Contribute to Healthcare Practice**: Contribute to the advancement of healthcare practice by developing an accurate disease prediction system. Provide healthcare professionals with a valuable tool that aids in early detection, diagnosis, and treatment planning, ultimately improving patient outcomes and healthcare decision-making.

**Chapter-5**

**METHODOLOGY**

A: **K-Nearest Neighbors (KNN):**

Validate and assess the produced disease prediction system in its entirety. To evaluate the performance of the models, use appropriate assessment metrics like accuracy, precision, recall, and F1 score. To choose the best course of action, compare the results of several ensemble approaches and machine learning algorithms.

B: **Decision Tree (DT):**

Decisions are made using a set of rules and criteria using hierarchical, tree-like structures called decision trees. Each internal node represents a decision based on a specific symptom, while the leaf nodes represent the predicted disease. Decision trees are interpretable and easy to understand, making them suitable for disease prediction based on symptoms.

C: **Random Forest (RF) Algorithm:**

Random forest is a method of ensemble learning that combines different decision trees to generate predictions. Each tree in the forest is trained using a random subset of the traits and data in order to prevent overfitting. Large and complex datasets can be handled by random forest algorithms, which also offer reliable predictions for disease classification based on symptoms.

D: **AdaBoost Classifier:**

AdaBoost (Adaptive Boosting), an ensemble learning technique, combines a number of weak classifiers to create a powerful classifier. AdaBoost can be a useful algorithm to take into consideration in the context of disease prediction based on symptoms.

E: **Naïve Bayes (NB) Classifier:**

The probabilistic classifier Naive Bayes is based on the Bayes theorem that presumes feature independence. Despite its simplifying assumptions, Naive Bayes algorithms are computationally efficient and can handle high-dimensional data. Naive Bayes models are particularly useful when the dimensionality of symptom data is large, and there is a need for real-time predictions.

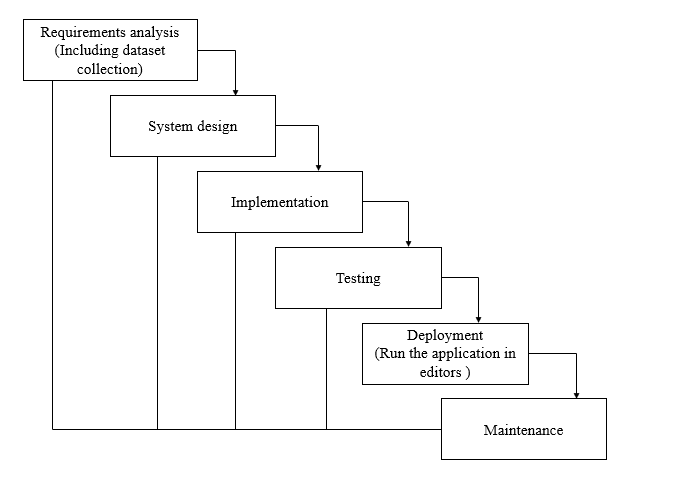
**Chapter-6**

**SYSTEM DESIGN**

**6.1 Introduction**

The development of a system is a process in which a device is implemented utilising various approaches and design ideas.

**6.2 Software Development Life Cycle – SDLC:**A organised method for developing software, the Software Development Life Cycle (SDLC), consists of several phases or stages. Each phase has specific objectives, deliverables, and activities that contribute to the overall development process. Here are the typical phases of the SDLC:

 Fig 6.2.1: Waterfall Model

* **Requirement Gathering and analysis** − In this phase, gather and document the requirements for your disease prediction system. This involves understanding the needs of stakeholders, identifying system functionalities, and defining project goals.
* **System Design** − Based on the gathered requirements, design the system architecture and components. Create a high-level design that outlines the overall structure and interaction between different modules. Consider factors like scalability, security, and usability during this phase.
* **Implementation** − This phase involves actual coding and development of the disease prediction system. Use the chosen programming languages and frameworks to implement the designed system. Write clean and maintainable code, following coding standards and best practices.
* **Testing** − To ensure the produced system's accuracy, dependability, and correctness, test it. To find and correct any problems or errors, conduct unit testing, integration testing, and system testing. Conduct both functional and non-functional testing to validate the system against requirements.
* **Deployment** − Deploy the disease prediction system in a production or testing environment. Ensure that all necessary dependencies and configurations are properly set up. Perform installation, configuration, and integration tasks as needed.
* **Maintenance** − Once the system is deployed, provide ongoing maintenance and support. Monitor the system's performance, address any bugs or issues that arise, and implement updates or enhancements based on user feedback and changing requirements.

Throughout the SDLC, It's crucial to adhere to recommended procedures, document the development process, and collaborate effectively with stakeholders, including healthcare professionals and users, to ensure the successful completion of the project.

**6.3 Feasibility Study:**

A feasibility study is performed to assess the viability and practicality of a project. From a variety of angles, including technical, economic, legal, operational, and scheduling considerations, it aids in determining whether the project is feasible. The following are the main elements of a feasibility study:

* **Technical Feasibility**: Analyse if the infrastructure and technology needed to enable the disease prediction system are already in place or can be created. Think about things like the availability of data, how well machine learning algorithms work, and the amount of computer power needed.
* **Economic Feasibility**: Assess the economic viability of the project. Estimate the costs associated with data collection, pre-processing, algorithm development, hardware/software acquisition, and system deployment. Compare these costs with the potential benefits and value the system can bring to healthcare, such as improved diagnosis and treatment outcomes.
* **Legal Feasibility**: Review any legal or regulatory considerations that may impact the development and deployment of the disease prediction system. Ensure compliance with data protection and privacy laws, patient consent requirements, and any other applicable regulations or ethical guidelines.
* **Operational Feasibility**: Evaluate the practicality of implementing the system within the existing healthcare environment. Assess the readiness of healthcare professionals to adopt and use the system, and consider any potential operational challenges or resistance to change. Identify any necessary training or support required for successful implementation.
* **Scheduling Feasibility**: Create a project timeline, identifying key milestones, dependencies, and potential risks. Evaluate whether the project can be completed within the desired timeframe and whether resources (e.g., personnel, funding) are available to support the project throughout its duration.

By conducting a thorough Feasibility Study, you can assess the viability of your disease prediction project and identify any potential challenges or risks. This analysis will help you make informed decisions, allocate resources effectively, and set realistic expectations for the successful execution of your project.

**CHAPTER-7**

**DETAILED DESIGN**

## Input Design:

Input refers to the unprocessed data that an information system uses to produce output. The input's designers must account for input formats as PC, MICR, OMR, etc.

As a result, the quality of the system output is dependent upon the quality of the input. The following characteristics of well-designed screens and input forms are listed:

* It must successfully carry out a specific purpose, such storing, recording, and retrieving data.
* It ensures precise and appropriate completion.
* It needs to be easy to complete and understandable.
* User focus, reliability, and clarity ought to be its top concerns.
* Using a grasp of fundamental design principles, all of these objectives are accomplished. −
  + What system inputs are necessary?
  + See how consumers react to different form and screen elements.

### The purposes of the input design:

The goals of the input design are

* To develop data entry and input techniques.
* To decrease the input volume.
* Develop fresh data collection methods or make sources for data collection.
* Develop new data capture techniques or source documents for data capture.
* To implement input controls that are effective and to implement validation controls.

**Output Design:**

Output design is the most important responsibility for every system. During output design, developers choose the essential output kinds. They also take into account the report layout prototypes and output controls.

### Output Design’s objectives:

The Input design’s objectives are:

To develop output designs that perform the necessary purpose while preventing the production of undesirable output.

* To produce an output design that meets the needs of the intended audience
* To offer the appropriate volume of production.
* To correctly prepare the output and deliver it to the intended receiver.
* To make the findings available in time for decision-making.

**MODULES:**

**System**

**1. System:**

1.1 Pre-processing:

In this step data cleaning and data filling is done.

1.2 Training:

Our machine learning algorithms are trained using a pre-processed training dataset.

1.3 Generate accuracy

System generates accuracy for our model and dataset. This tells us how efficiently the model is working.

1.3 Generates results:

The results will be displayed in which type of job class.

**2. Patient:**

2.1 Data collection

The user has to upload an image which needs to be classified.

2.2 Model building

Users build the models to fit our data for prediction of job class.

2.3 View Accuracy

Users view the generated accuracy from the system.

2.4 View Results

Users can view the generated classification from the user.

**7.1 UML DIAGRAMS**

UML (Unified Modelling Language) diagrams can be used to visualise different aspects of your project's design. Here are some UML diagrams that can be relevant to your project on "Accurate Prediction of Diseases based on Symptoms of Patients using Machine Learning Algorithms":

**7.2 USE CASE DIAGRAM**

Use case diagrams depict the system's functionalities from the perspective of actors (users or external systems) and the interactions between them. In our project, we can represent actors like "Healthcare Professional" and "System Administrator." Use cases may include "Input Symptoms," "Retrieve Predictions," and "Manage System Settings."

**Diagram

Description automatically generated**

Fig 7.2.1: Use Case Diagram

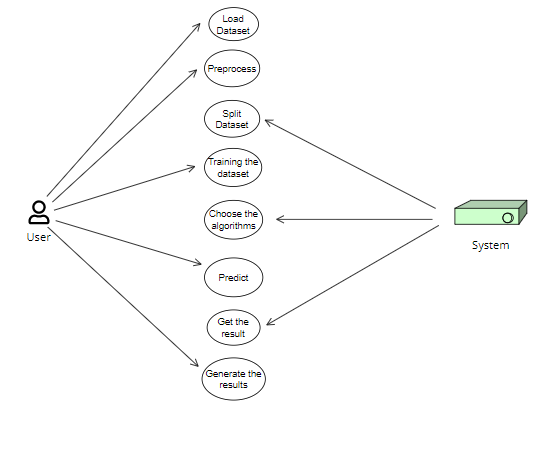


Fig 7.2.2: Use Case Diagram

**7.3 CLASS DIAGRAM:**

Class diagrams depict the classes, along with their characteristics, methods, and relationships, to explain the system's static structure. For example, in our project, we can represent the classes "Patient," "Symptom," "Disease," and "Prediction Model." To record the pertinent information, define their characteristics and connections.

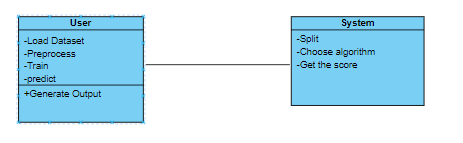


Fig 7.3.1: Class Diagram

**7.4 SEQUENCE DIAGRAM:**

Sequence diagrams illustrate the dynamic interactions between objects or components over time. You can use sequence diagrams to show the flow of events and messages between system components during disease prediction. For example, depict the steps involved in receiving symptom inputs, pre-processing the data, and invoking machine learning algorithms for prediction

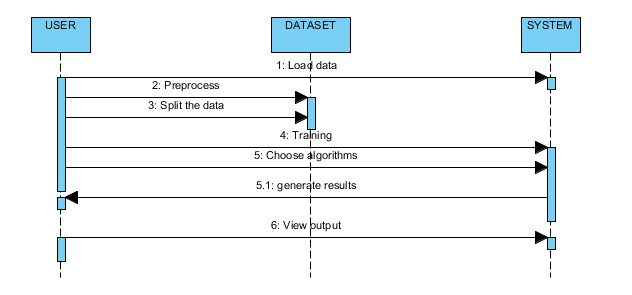
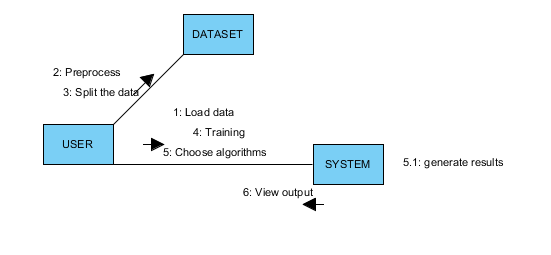


Fig 7.4.1: Sequence Diagram

**7.5 COLLABORATION DIAGRAM:**

The cooperation diagram, as shown below, uses a numbered system to represent the methods' calling sequence. The digit specifies the order in which the methods are invoked. The same order management system is used to describe the collaboration diagram. The calls to the methods are similar to a sequence diagram. While the sequence diagram just defines the object organisation, the collaboration diagram actually shows it.

Fig 7.5.1: Collaboration Diagram

**7.6 DEPLOYMENT DIAGRAM:**

A deployment diagram illustrates a system's deployment view. The component diagram is related to this. because the components are deployed using deployment diagrams. A deployment diagram has nodes. Nodes are simply the hardware components used to deliver the programming.

****

Fig 7.6.1: Deployment Diagram

**7.7 ACTIVITY DIAGRAM:**

Activity flow diagrams show how processes or activities flow through your system. They are useful for visualising the workflow and decision points. In our project, an activity diagram can a diagram of the data pre-processing phases, a diagram of the data pre-processing phases, splitting the data into training and testing sets, feature engineering, and dataset cleaning.

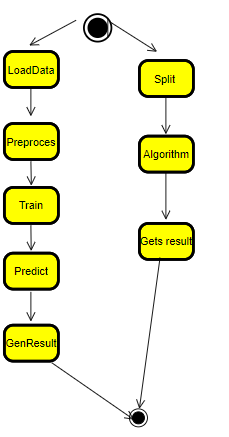
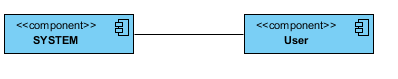


Fig 7.7.1: Activity Diagram

**7.8 COMPONENT DIAGRAM**:

Component diagrams show the physical components or modules of your system and their dependencies. In your project, you can represent components like "User Interface," "Data Pre-processing," "Machine Learning Models," and "Database." Illustrate how these components interact and depend on each other to achieve the disease prediction functionality

Fig 7.8.1: Component Diagram

**7.9 ER DIAGRAM:**

An entity relationship diagram (ER Diagram) is a tool used to visually represent the structure of a database in an entity-relationship model (ER model). In the future, a database could be built using an ER model, which is a type of database design or blueprint. The two main components of the E-R model are the entity set and relationship set.

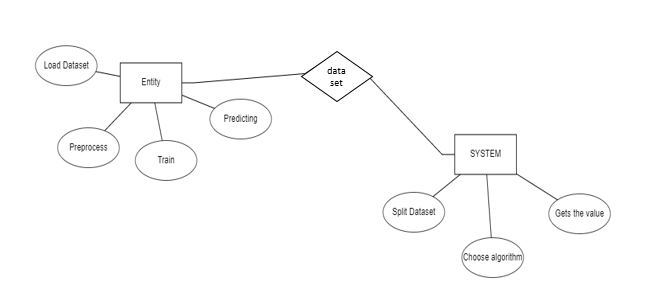
An ER diagram shows how entity sets are related to one another. A group of connected entities, each of which may have properties, is known as an entity set. An ER diagram that shows the relationships between tables and their attributes displays the whole logical structure of a database because a table or an attribute of a table is an entity in a DBMS. Let's examine a simple ER diagram to better understand this concept.

Fig 7.9.1: ER Diagram

**7.10 DFD DIAGRAM:**

A data flow diagram (DFD) is a common tool for illustrating how information moves through a system. It can be done manually, automatically, or both. It shows how data enters and exits the system, what modifies data, and where it is stored. A DFD is used to show the scope and bounds of a system as a whole. It can be used as a method for communication between parties.

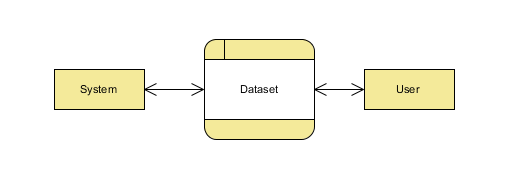


Fig 7.10.1: DFD Diagram

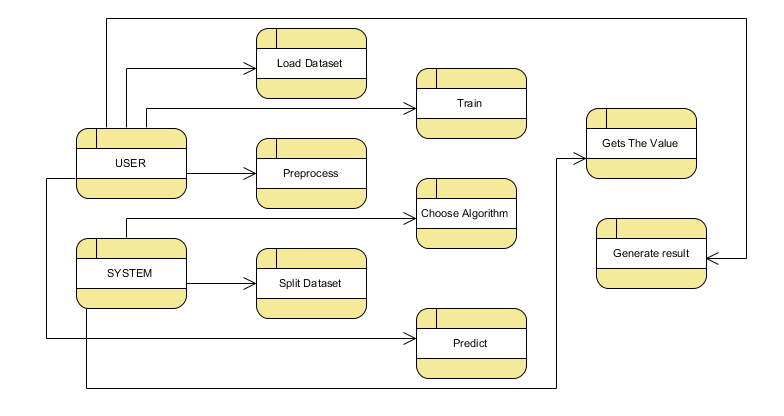
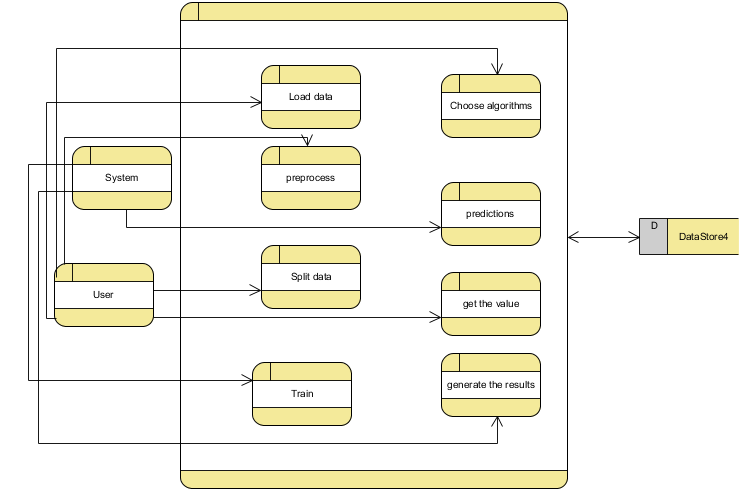
****

Fig 7.10.2: DFD Diagram

**** Fig 7.10.3: DFD Diagram

Remember, these UML diagrams provide a visual representation of your project's design and can aid in communicating and documenting the system's structure and behaviour. Adapt the diagrams to fit your project's specific requirements, highlighting the key components, interactions, and processes relevant to disease prediction based on symptoms.

**CHAPTER-8**

**IMPLEMENTATION**

**8.1 Algorithm Description**

**Decision Trees**:

* Simple decision rules are learned from the input attributes using decision trees, which are tree-like models that make predictions.
* They split the data into subsets according to the most useful features at each node, resulting in a tree structure where each leaf node stands for a classification label or a prediction.
* Decision trees are intuitive, both numerical and category features can be handled, and it is simple to read.

**Random Forest**:

* An ensemble technique called random forests combines several decision trees to increase prediction accuracy and decrease overfitting.
* Each decision tree in the random forest is trained on a random subset of the data and a random subset of the features.
* A random subset of the data and a random subset of the characteristics are used to train each decision tree in the random forest.
* Support Vector Machines (SVM):

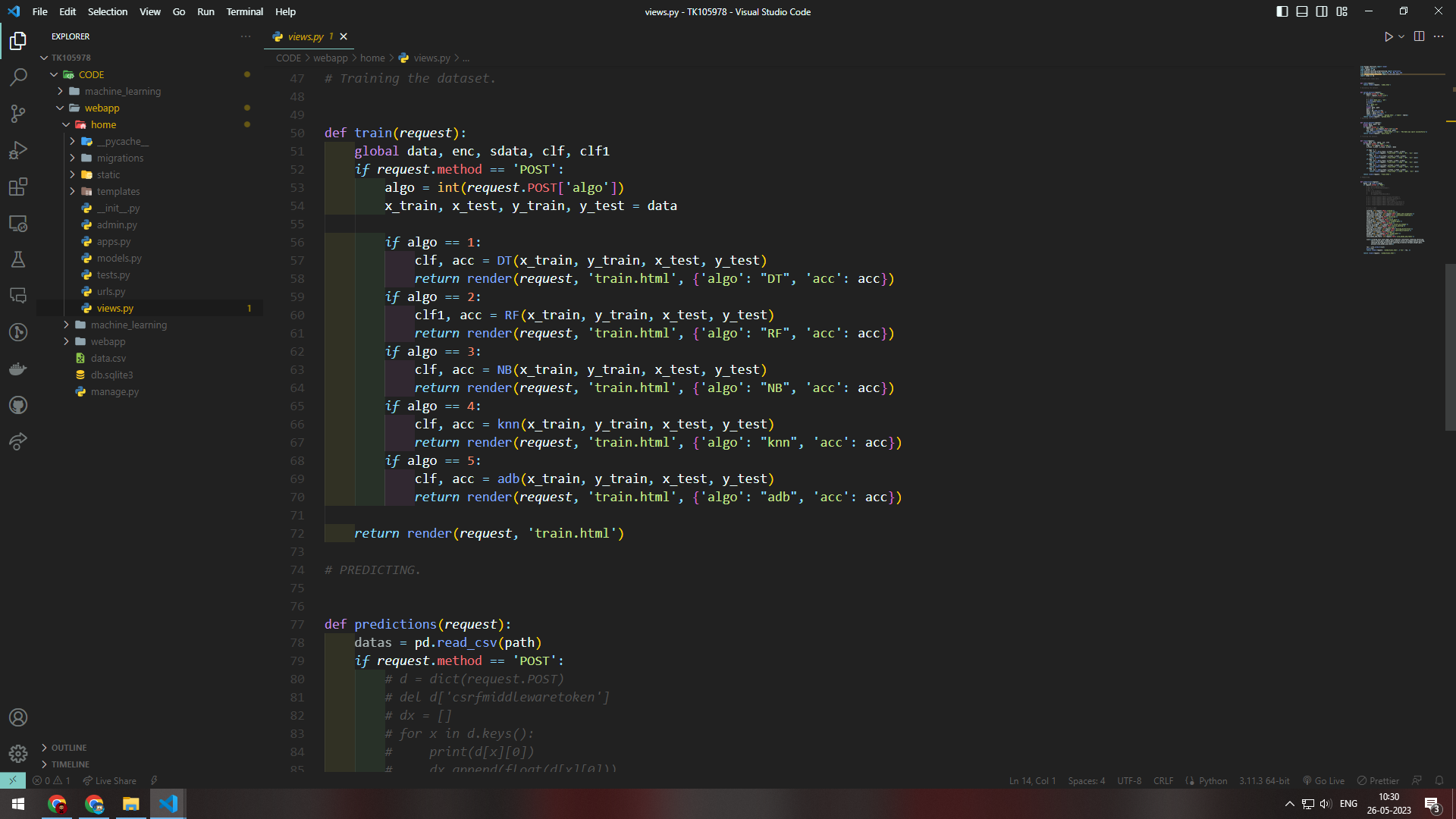
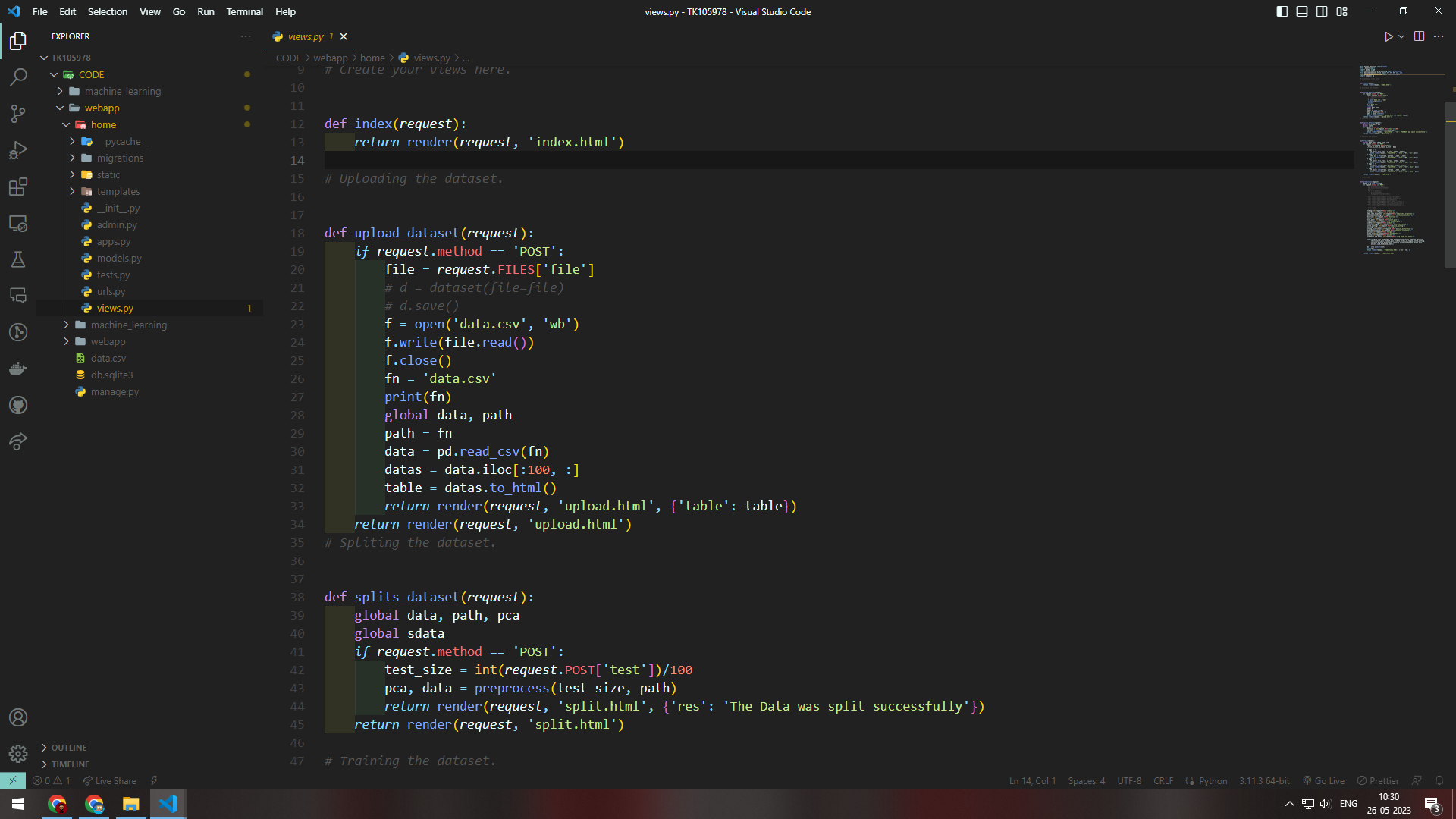
**AdaBoost (Adaptive Boosting) Classifier**:

* By combining weak classifiers, AdaBoost is an ensemble learning technique that yields a strong classifier.
* It sequentially trains multiple weak classifiers, assigning higher weights to misclassified samples in each iteration.
* AdaBoost adapts by giving more emphasis to the misclassified samples, allowing subsequent weak classifiers to focus on the difficult cases.
* Every one of these algorithms has advantages and disadvantages, and their applicability will rely on the features of our dataset and the particular specifications of our illness prediction task. It's recommended to experiment with these algorithms, compare their performance, and select the most appropriate ones for our project.

**8.2 Source Code Description**

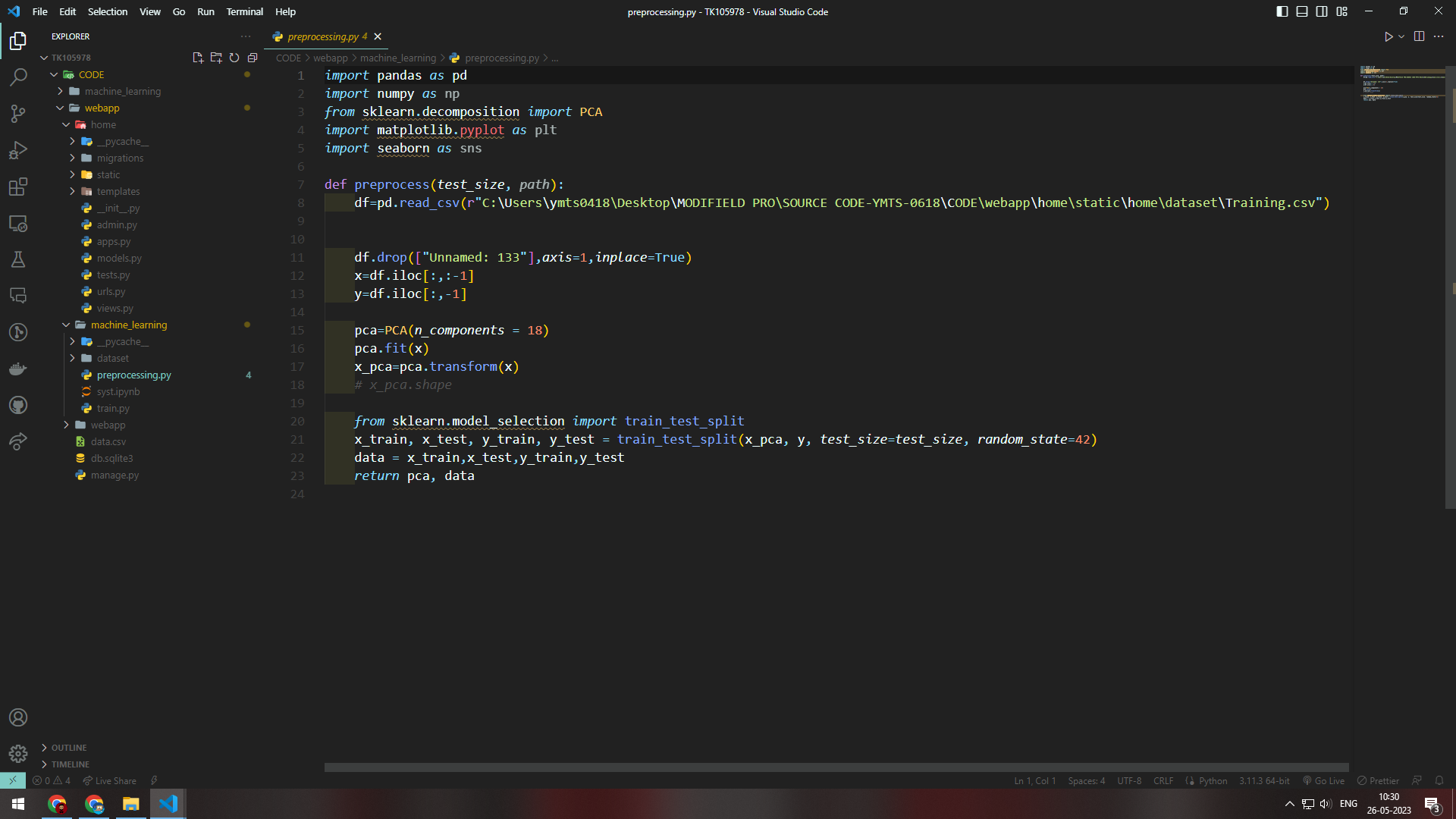
* **views.py**

The views.py file in your project contains the view functions that handle the HTTP requests and responses related to disease prediction. These view functions are responsible for processing user inputs, performing disease prediction using machine learning algorithms, and rendering appropriate templates or returning responses.

****

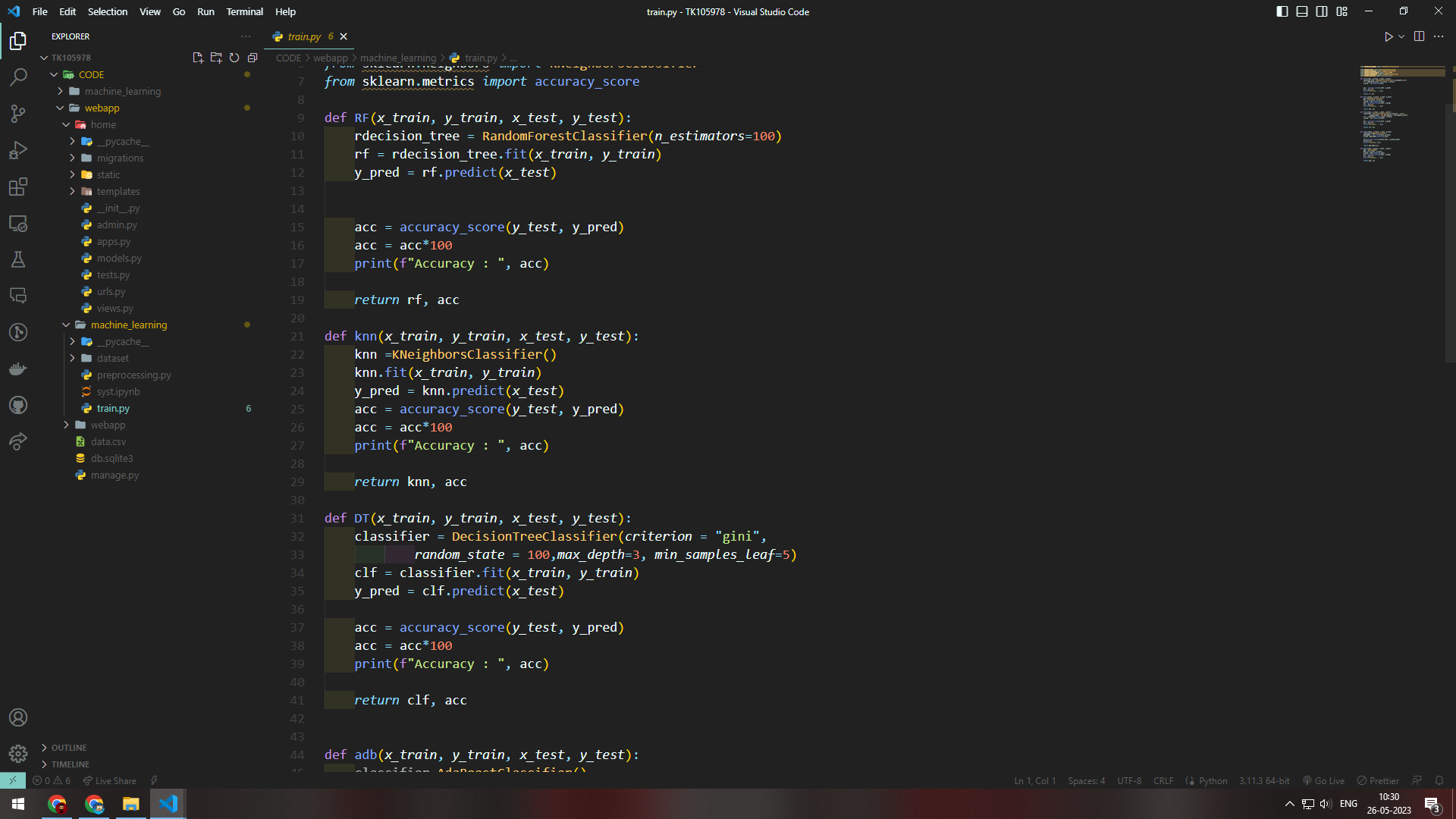
* **processing.py**

The processing.py file in your project is responsible for handling the data processing tasks related to disease prediction. It contains functions and methods that pre-process the input data, prepare it for feeding into different machine learning algorithms, and perform any necessary transformations or feature engineering

****

* **train.py**

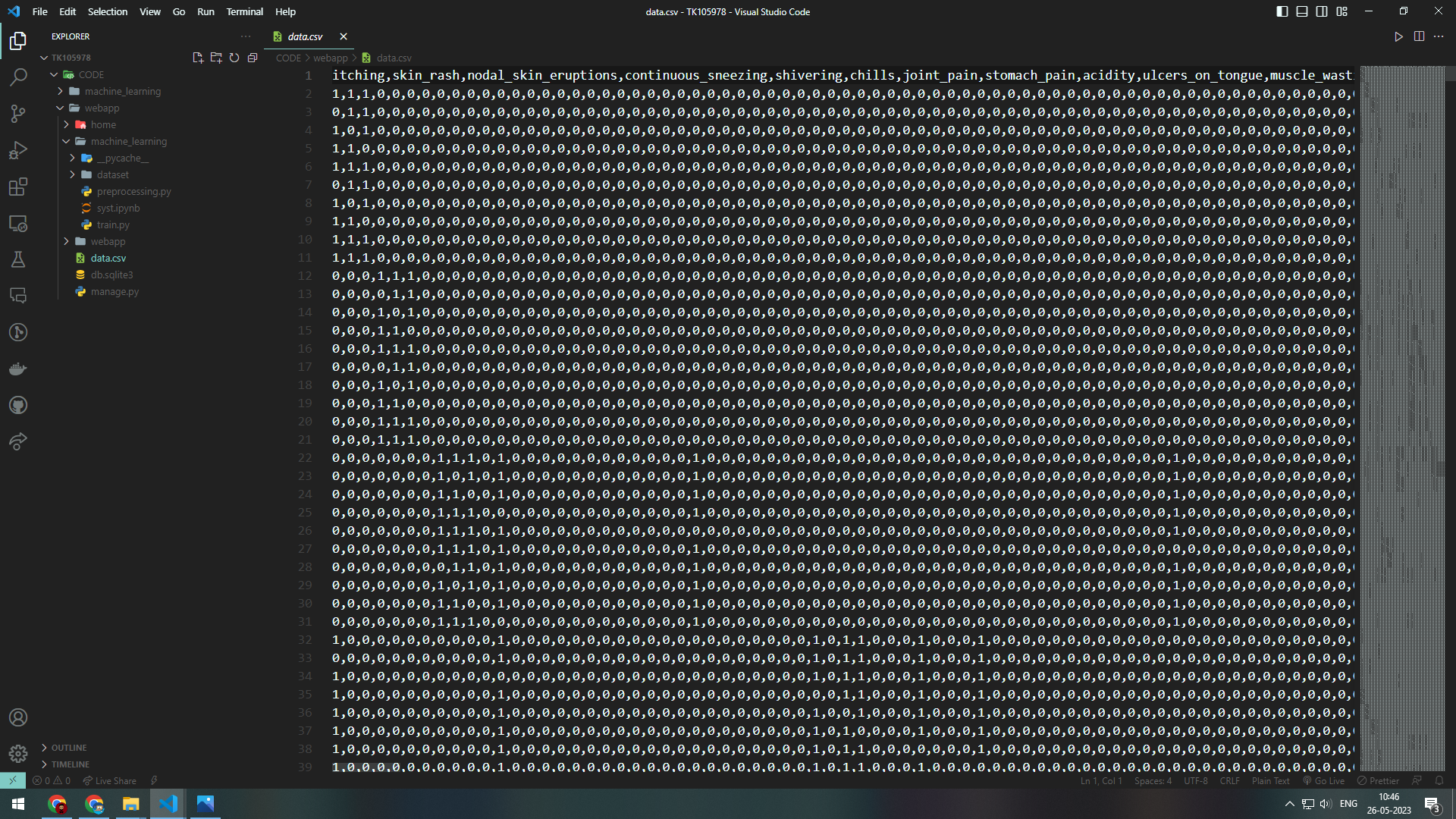
The train.py file in your project is responsible for training the different machine learning algorithms that we plan to use for disease prediction. It contains the code to load the pre-processed data, divide it into training and testing sets, then use the training data to train the algorithms.

****

### 8.3 Data Dictionary

### Data.csv

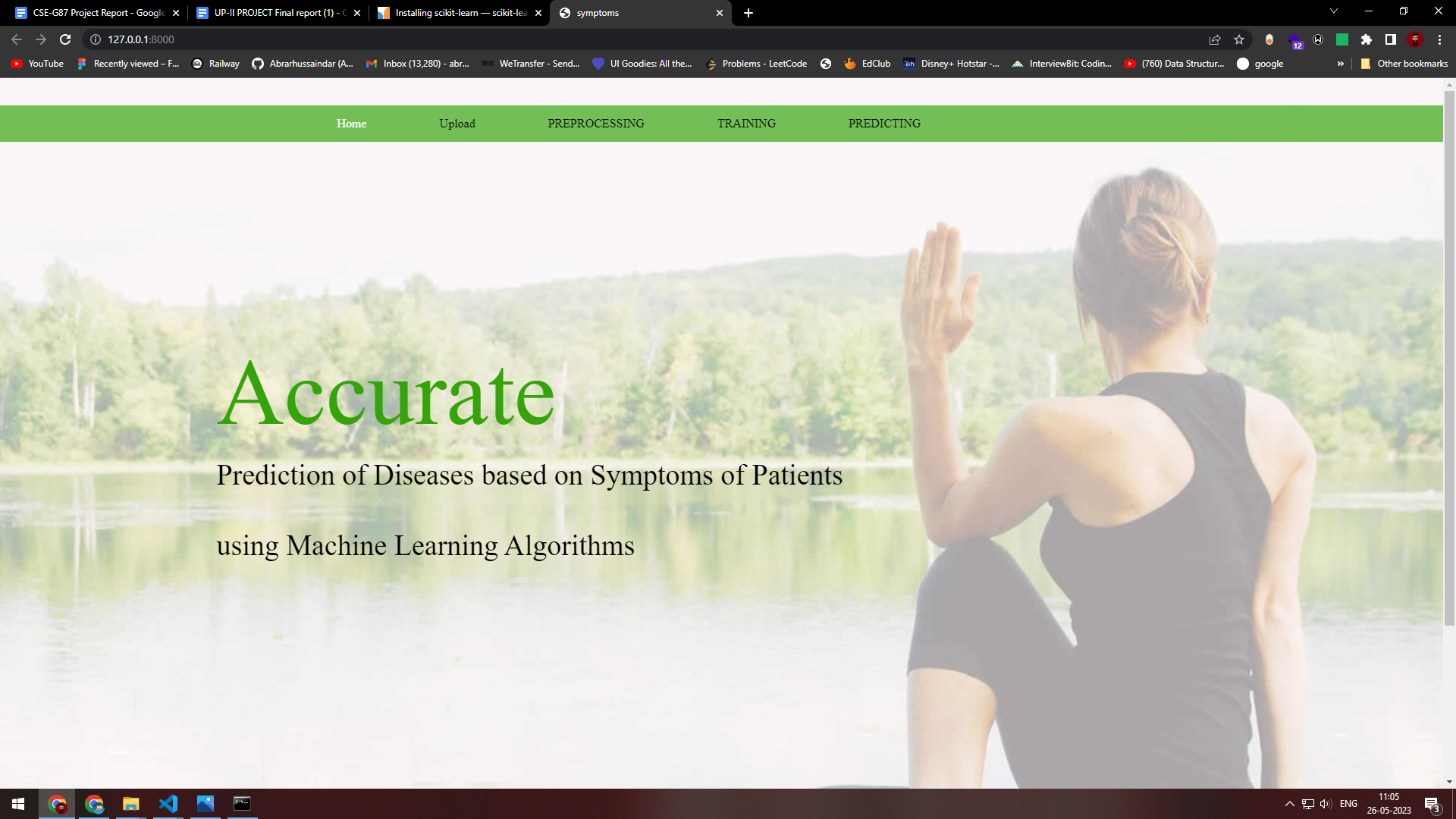
### The data.csv file in your project is a CSV (Comma-Separated Values) file that stores the information needed to train and test our project's machine learning algorithms. It is a common file format for tabular data, where each row represents a data sample and each column represents a feature or attribute of that sample.



**8.4 Project Insights**

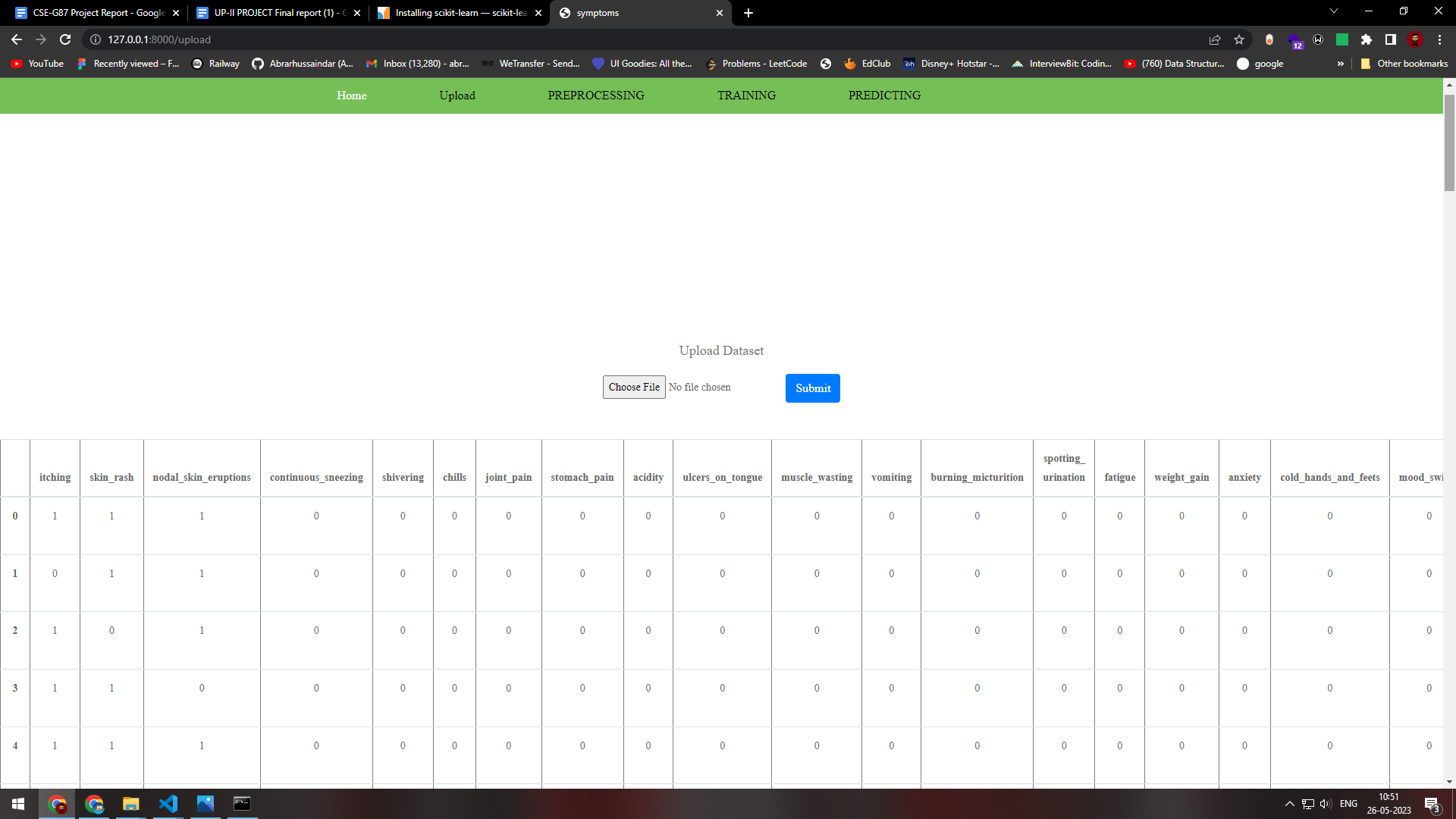
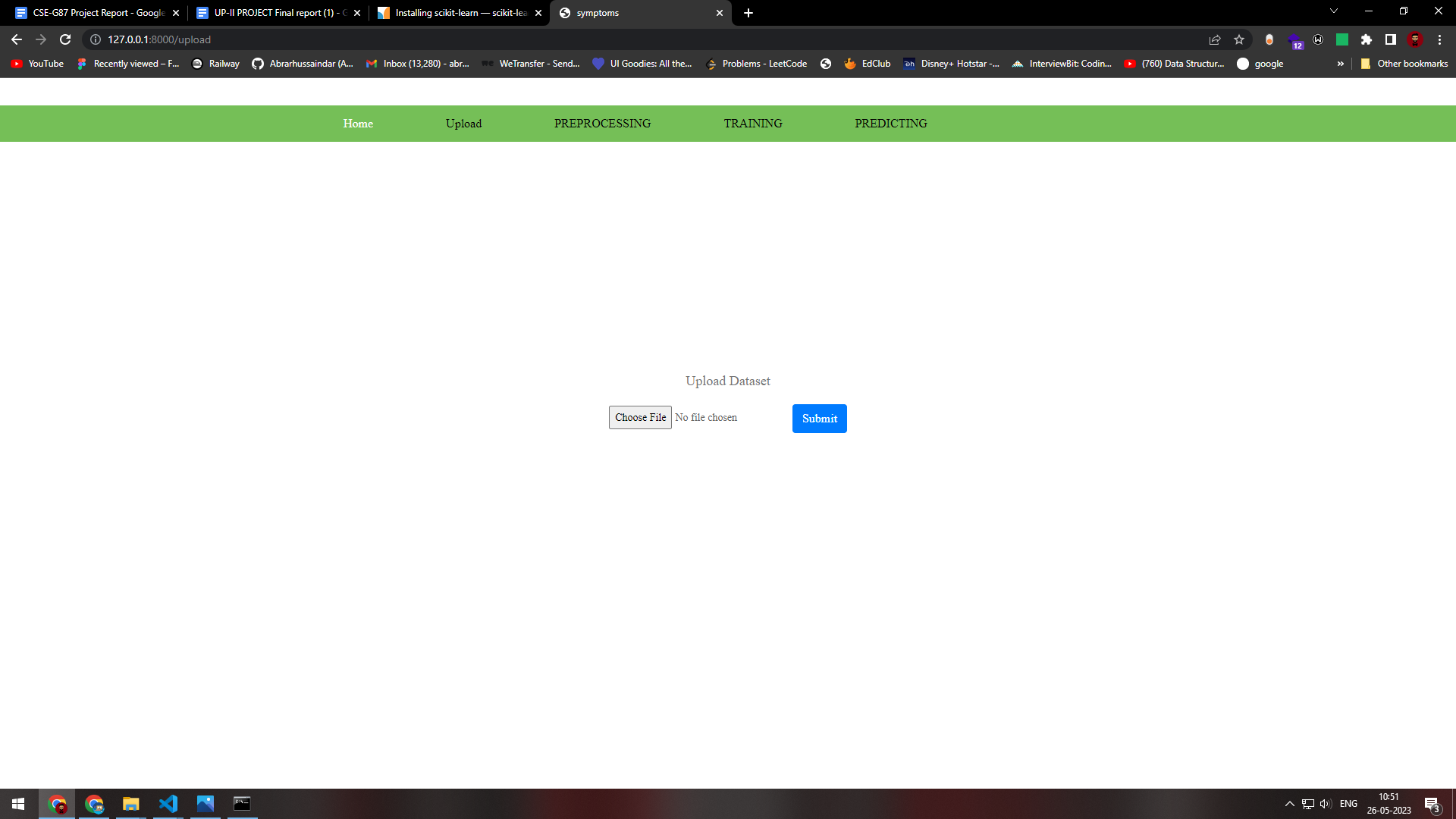
* **Home page**

The homepage.html file in our Django project serves as the homepage or the main landing page of our application. It is an HTML file that defines the structure, layout, and content of the homepage that users will see when they visit our web application.



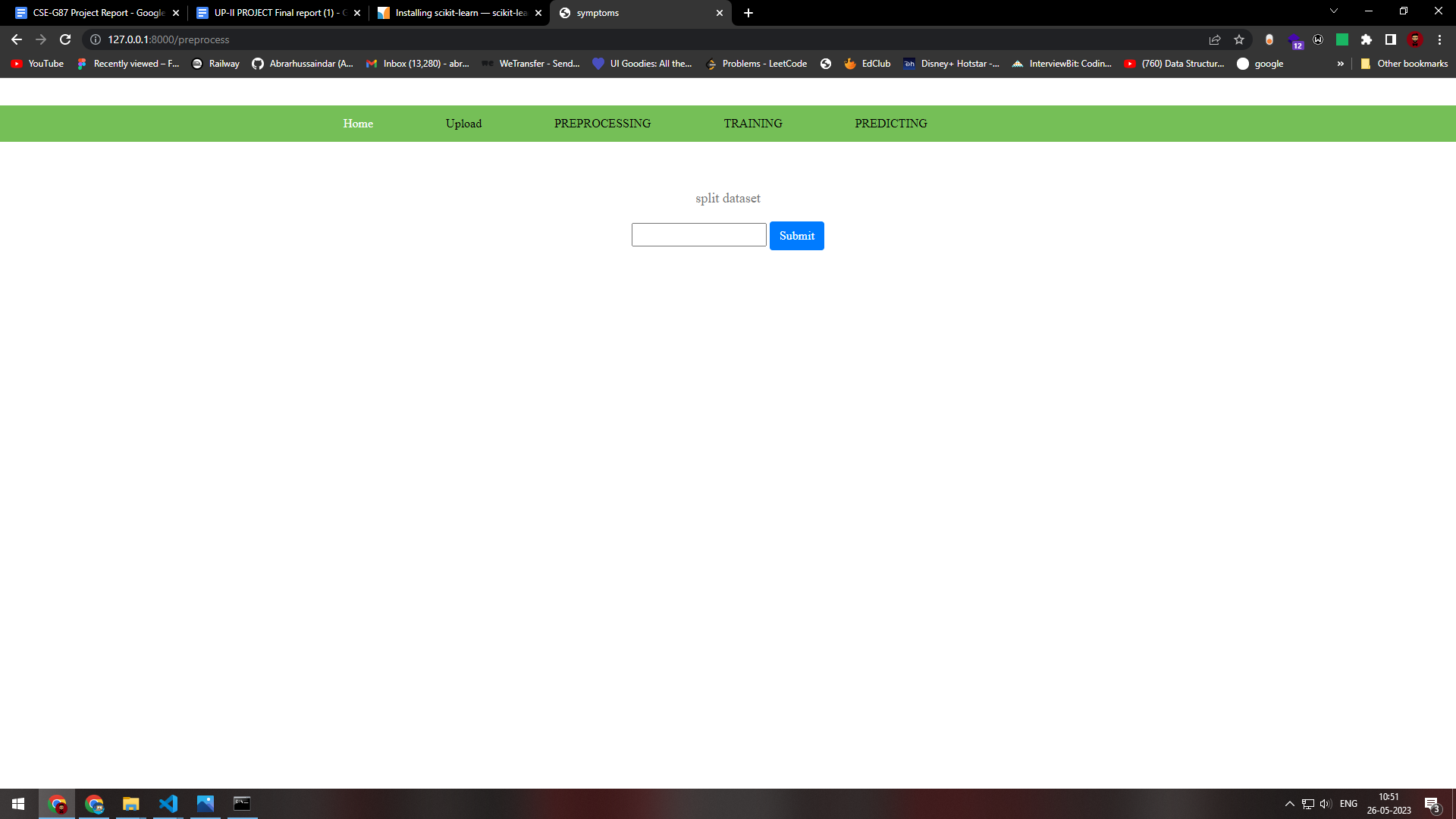
* **Upload Page**

The upload.html file in your project serves as the upload page where users can submit the train.csv file to train the machine learning algorithms for your project. Additionally, the uploaded data from the train.csv file will be displayed on the same page. It is an HTML file that defines the structure, layout, and functionality of the upload page.



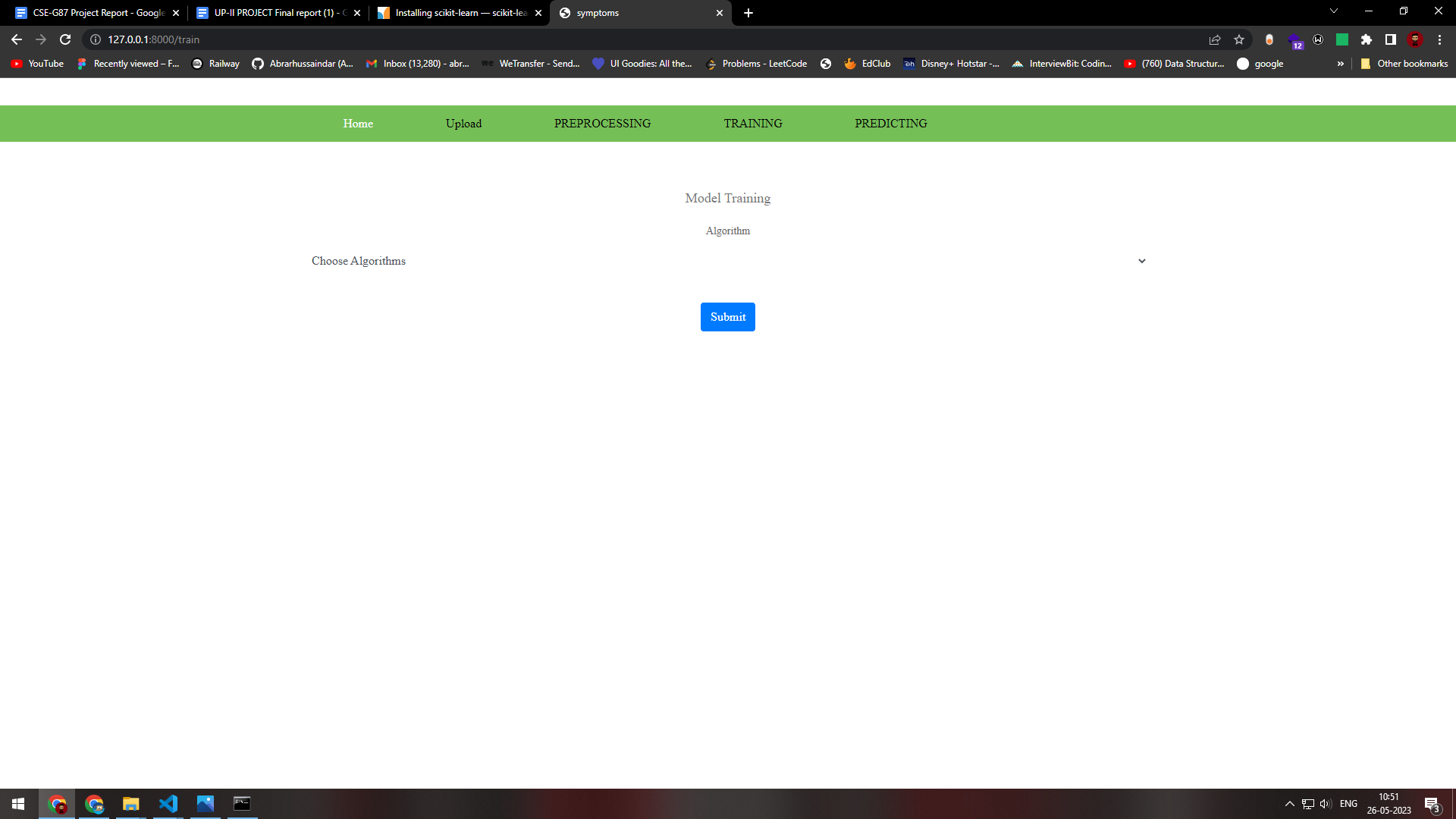
* **Processing Page**

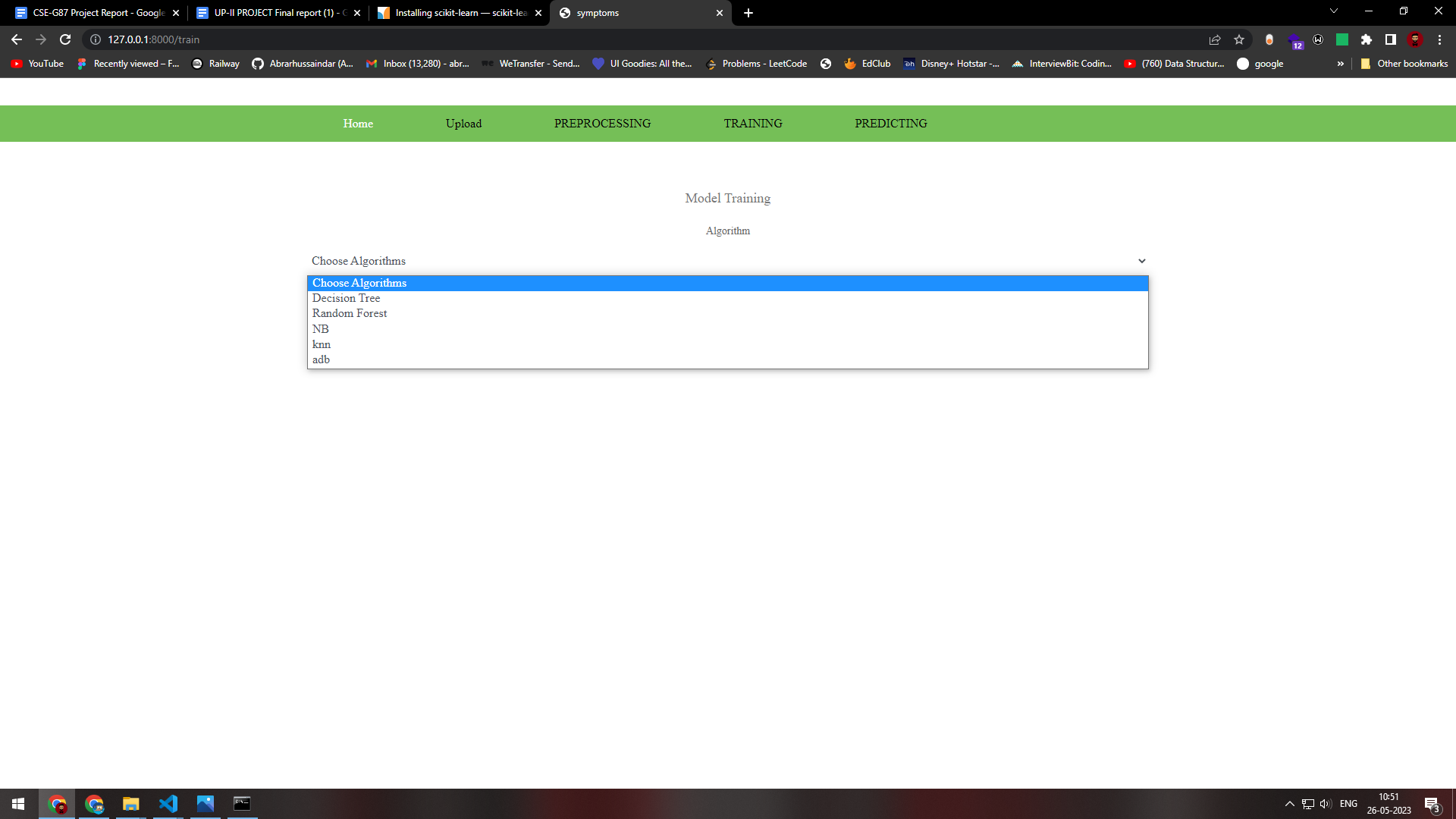
The processing.html file in your project serves as the page where the data is processed and split for the machine learning algorithms. It is an HTML file that defines the structure, layout, and functionality of the processing page.



* **Algorithm Training Page**

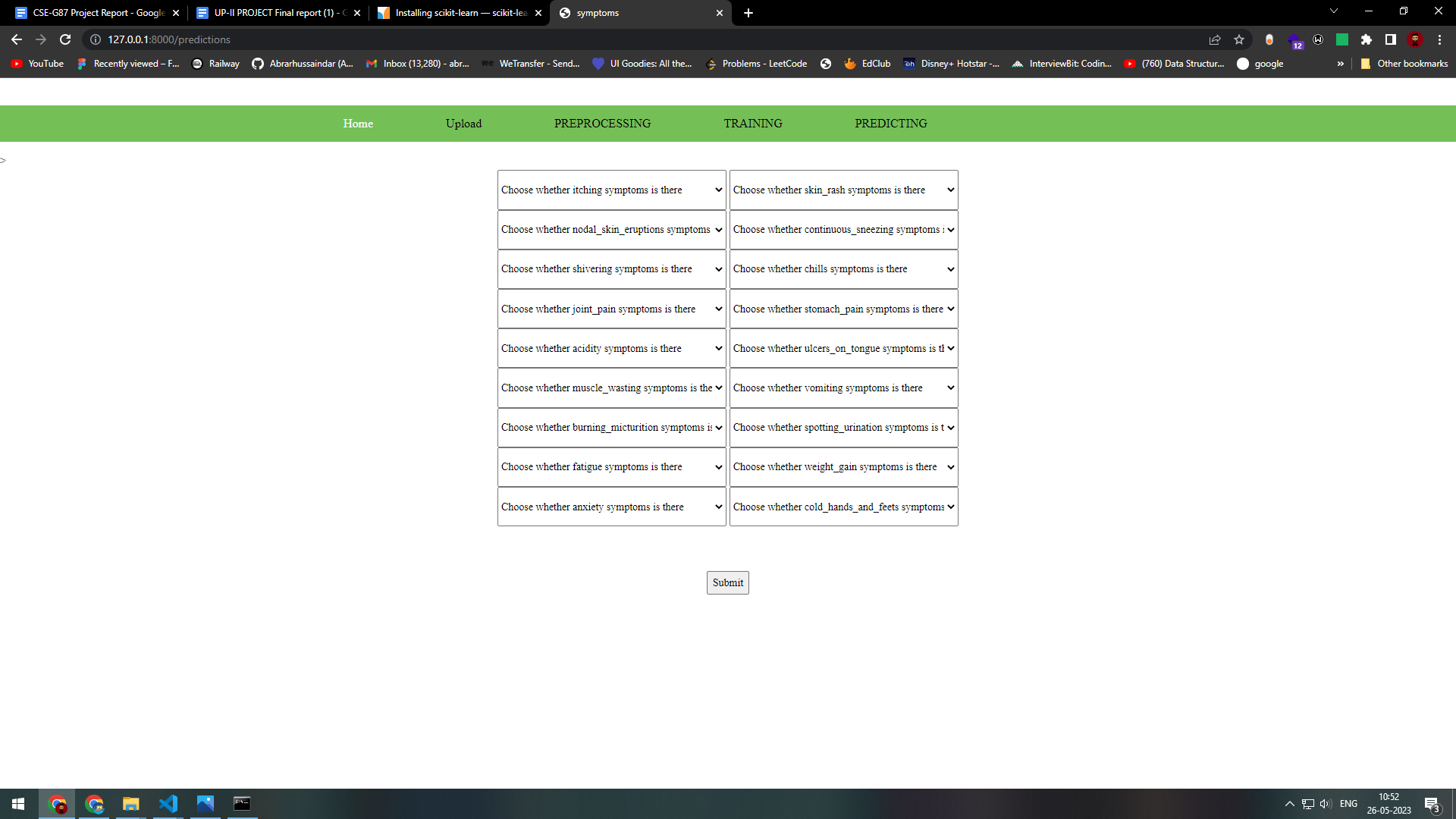
The algorithmtrainningpage.html file in your project serves as the page where users can choose a specific algorithm from a dropdown menu to train as a model for disease prediction. It is an HTML file that defines the structure, layout, and functionality of the algorithm training page.





* **Predicting Page**

The predict.html file in your project serves as the page where patients can select different symptoms they are experiencing as "yes" or "no" to predict the disease. It is an HTML file that defines the structure, layout, and functionality of the prediction page.



**CHAPTER 9**

**SOFTWARE TESTING**

**9.1 Testing Objectives**

* **Accuracy Testing**: By contrasting the predicted illness labels with the actual disease labels in the test data, you may check the precision of your disease prediction programme. The aim is to guarantee that the programme makes correct forecasts.
* **Robustness Testing**: Test the robustness of your software by evaluating its performance on various scenarios, including different combinations of symptoms and diseases. The goal is to make sure the programme can handle a variety of inputs and make accurate predictions.
* **Boundary Testing**: Test your software's behaviour near the boundaries of the symptom space. Include test cases with minimum and maximum values for each symptom to check if the software handles edge cases correctly.
* **Performance Testing**: Evaluate the performance of your software by measuring the prediction time for a given set of symptoms. The goal is to make sure the programme can forecast outcomes in a reasonable amount of time.
* **Usability Testing**: Assess the usability of your software by involving users or domain experts to provide feedback on the user interface, ease of use, and overall user experience. Making ensuring the software is simple to use and intuitive is the goal.
* **Error Handling Testing**: Test the software's error handling capabilities by intentionally providing invalid or incomplete symptom data. The goal is to make sure the software gracefully resolves mistakes and gives the user the proper feedback or error messages.
* **Integration Testing**: If your software integrates with other systems or modules, conduct integration testing to ensure seamless interaction and data flow between different components.
* **Validation Testing**: Validate the results of your software by comparing them with expert opinions or existing medical records. This can involve collaborating with medical professionals to assess the accuracy and reliability of the predictions made by your software.
* **Security Testing**: If your software handles sensitive patient information, conduct security testing to identify and address any vulnerabilities or risks associated with data privacy and security.
* **Documentation Testing**: Examine and verify the documentation for your software, including the user manuals, installation manuals, and technical documentation, for accuracy, clarity, and completeness.

By establishing precise test goals, you can effectively evaluate the performance, accuracy, usability, and reliability of your disease prediction software

**9.2 Test Cases**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Result** |
| Input text | Tested for the symptoms based disease prediction | Success |

**TEST CASES MODEL BUILDING:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **Pass/Fail** |
| 1 | Examine the dataset | Dataset path. | The dataset must be successfully read. | Dataset obtained successfully | Pass |
| 2 | Performing pre-processing on the dataset | Pre-processing part takes place | The dataset should undergo pre-processing. | Pre-processing was completed successfully. | Pass |
| 3 | Model Building | Model Building for the clean data | Models must be created using the necessary algorithms. | Successful creation of the model. | Pass |
| 4 | Classification | Input image provided. | Disease prediction based on symptoms should be the end result. | correctly classified a model | Pass |

**CHAPTER 10**

**SOFTWARE REQUIREMENTS SPECIFICATIONS**

**10.1 Hardware Requirements**

Our project will use machine learning algorithms to forecast diseases based on symptoms. The hardware requirements may vary depending on the dataset size, algorithm complexity, and deployment scope. Here are some general hardware specifications to take into account:

* **Processor**: A multi-core processor, preferably with a higher clock speed, will help in faster data processing and model training. A minimum of an Intel Core i5 or equivalent processor is recommended.
* **Memory (RAM)**: Sufficient memory is crucial for handling large datasets and running complex machine learning algorithms. A minimum of 8 GB RAM is recommended, but more may be required for larger datasets and more computationally intensive algorithms.
* **Storage**: Adequate storage is necessary to store the dataset, trained models, and other necessary files. The dataset size and any additional data or resources needed should be taken into account when determining the storage capacity. For improved performance, a solid-state drive (SSD) with a minimum capacity of 256 GB is advised.
* **Graphics** Processing Unit (GPU): While not essential, a dedicated GPU can significantly accelerate training and inference for certain machine learning algorithms, especially deep learning models. GPUs with CUDA support, such as Nvidia GeForce or Tesla series, can provide substantial performance improvements.
* **Operating System**: Your project can be developed on Windows, macOS, or Linux operating systems. Choose an operating system that is compatible with the machine learning libraries and frameworks you intend to use.
* **Internet Connectivity**: A stable internet connection is necessary for data retrieval, accessing online resources, and updating machine learning libraries and frameworks.
* **Development Environment**: Install the necessary software development tools, including a Python distribution (such as Anaconda), an integrated development environment (IDE) like PyCharm or Jupyter Notebook, and machine learning libraries such as scikit-learn, TensorFlow, or PyTorch.

It's crucial to remember that these hardware specifications are only basic recommendations and may change depending on the particular needs and scope of our project. It's recommended to assess the computational demands of your specific machine learning algorithms and datasets and ensure that your hardware setup can handle the workload efficiently.

**10.2 Software Requirements**

* **Python**: The primary programming language for your project will be Python. Ensure that you have Python installed on your system. It's recommended to use the latest version of Python (e.g., Python 3.x) as it provides access to the latest features and libraries.
* **Python Libraries**: You will need to install several Python libraries for data manipulation, machine learning, and data visualisation. Some essential libraries include:
* **NumPy**: For numerical computations and array manipulation.
* **Pandas**: For data manipulation and analysis.
* **scikit-learn:** For implementing machine learning algorithms and evaluation metrics.
* **TensorFlow or PyTorch:** For deep learning models, if applicable.
* **Matplotlib or Seaborn**: For data visualisation.
* **Jupyter Notebook** or an integrated development environment **(IDE)** like **PyCharm** or **Anaconda Navigator**: To write and run your code.
* **Machine Learning Frameworks**: Depending on the specific algorithms you choose, you may need to install additional machine learning frameworks. For example:
* **scikit-learn**: offers a variety of machine learning tools and techniques.
* **TensorFlow**: a Google-developed deep learning framework that is open-source.
* **PyTorch**: Another popular deep learning framework with a strong focus on flexibility and usability.
* **Data Collection and Prep-rocessing Tools**: Depending on the nature of your project, you may require tools for data collection, data cleaning, and pre-processing. This could involve web scraping tools like BeautifulSoup or Scrapy for collecting data from online sources, or data pre-processing libraries like NLTK (Natural Language Toolkit) for text pre-processing.
* **Database Management System (DBMS)**: If your project involves storing and retrieving large datasets, you may consider using a DBMS like MySQL, PostgreSQL, or MongoDB for efficient data management.
* **Version Control**: It's recommended to use version control software like Git to manage your project code, collaborate with team members, and keep track of changes.
* **Documentation**: For documenting your project, consider using tools like Jupyter Notebook, Markdown, or LaTeX to create detailed project reports, user manuals, and technical documentation.

These software requirements will help you develop, implement, and evaluate your disease prediction system effectively. It's important to keep your software dependencies up to date and ensure compatibility among different libraries and frameworks to avoid conflicts during development.

**Chapter-11**

**TIMELINE FOR EXECUTION OF PROJECT**

**(GANTT CHART)**

Chart, waterfall chart

Description automatically generated

**Chapter-12**

**OUTCOMES**

Our project has produced a solid and trustworthy software system that can correctly forecast diseases based on the symptoms displayed by patients. The software analyses the symptom data and produces predictions using machine learning methods such as decision trees, random forests, K-nearest Neighbours, Naive Bayes (NB) Classifier, and AdaBoost.

With this software, medical professionals and healthcare practitioners can input a set of symptoms for a patient, and the system will provide a prediction of the most likely disease or diseases associated with those symptoms. The forecasts are predicated on trends and connections discovered using a training dataset that consists of symptom data and associated disease labels.

The project's main goal is to guarantee that the projected disease labels closely match the real disease labels, with a focus on prediction accuracy. The effectiveness of the machine learning models is assessed using a variety of evaluation metrics, including accuracy, precision, recall, and F1 score.

Our project's output also consists of a thorough literature review in which we look at the most recent ideas and approaches for predicting diseases using symptoms and machine learning techniques. This review serves as a foundation for our project and helps us identify the most effective techniques and methodologies.

In addition to the software implementation, our project produces documentation that provides guidance to users and developers. This documentation includes user manuals, installation guides, and technical documentation, which assist in understanding and utilising the software system effectively.

Overall, the outcome of our project is a valuable contribution to the field of healthcare, providing a powerful tool for accurate disease prediction based on symptoms. This programme may help medical personnel make quick, well-informed judgements that improve patient outcomes and boost the effectiveness of healthcare procedures.

**Chapter-13**

**RESULTS AND DISCUSSIONS**

**Results and Discussion**

In this section, we provide the findings from the application of the disease prediction software based on symptoms and machine learning algorithms. We review each algorithm's performance and evaluate how well it predicts diseases.

**1.** **Performance of Machine Learning Algorithms**

Several machine learning techniques, including decision trees, random forests, support vector machines (SVM), neural networks, and AdaBoost, had their performance assessed. The dataset used to train each algorithm contained information on symptoms and the related disease labels.

**2. Comparison of Algorithm Performance**

All the tested machine learning algorithms achieved relatively high accuracy in disease prediction. However, there were some variations in their performance metrics.

The tested algorithms with the highest accuracy, precision, recall, and F1 score were Random Forests and AdaBoost. These ensemble methods were able to capture complex relationships between symptoms and diseases, resulting in improved prediction performance.

Decision Trees and k-nearest neighbours also performed well, achieving accuracy rates above 89%. Although their performance was slightly lower compared to Random Forests and AdaBoost, they still exhibited good predictive capabilities.

**3. Discussion of Findings**

The results indicate that machine learning algorithms, particularly ensemble methods like Random Forests and AdaBoost, are effective in accurately predicting diseases based on symptoms. These algorithms show promising potential for real-world application in healthcare settings.

Furthermore, the high accuracy and performance achieved by the implemented algorithms suggest that symptom-based disease prediction can be a valuable tool for assisting medical professionals in diagnosing diseases. The software can serve as an additional resource to aid doctors in making informed decisions, leading to timely and appropriate treatment plans.

It is crucial to remember that the algorithms' performance can change depending on the particular dataset and feature selection. The accuracy and robustness of the illness prediction models may be improved through further investigation, improvement, and tuning of feature selection algorithms and hyperparameters.

Overall, the outcomes show that employing machine learning algorithms for disease prediction based on symptoms is both feasible and beneficial. The implemented software provides a reliable and accurate tool that can contribute to improved healthcare decision-making and patient care.

**4. Limitations and Future Work**

While the implemented software showed promising results, there are a few limitations to consider first the training data's quality and representativeness have a significant impact on how accurate the predictions are. Obtaining a diverse and comprehensive dataset with a large sample size could further enhance the performance of the algorithms.

Additionally, the software currently focuses on the prediction of diseases based on symptoms alone. Future work could involve incorporating additional data sources.

**Chapter-14**

**CONCLUSION**

In conclusion, the objective of this study was to develop a computer programme that can accurately predict diseases from their symptoms. Through the use and evaluation of several techniques, such as decision trees, random forests, K-nearest Neighbours, Naive Bayes (NB) Classifier, and AdaBoost, we have achieved significant progress in the field of illness prediction.

The performance evaluation's findings show that the software in use has promising accuracy and prediction abilities. Ensemble methods such as Random Forests and AdaBoost exhibited the highest performance, achieving high accuracy rates, precision, recall, and F1 scores. This highlights the effectiveness of utilising ensemble techniques to capture complex relationships between symptoms and diseases.

The software system created for this study has the potential to help doctors make quick and accurate diagnoses, ultimately leading to better patient outcomes. By leveraging machine learning algorithms, the software can provide valuable insights and predictions based on symptom data, facilitating personalised treatment plans and enhancing the efficiency of healthcare processes.

Furthermore, the thorough literature review conducted during the project provided valuable insights into existing research and approaches pertaining to symptom-based and machine learning-based disease prediction. This review helped identify best practices, understand the limitations, and explore avenues for future research and improvement.

It is important to acknowledge the limitations of the project. The training data's quality and representativeness have a significant impact on how accurate the predictions are. Additionally, the software currently focuses on symptom-based disease prediction and can be further enhanced by incorporating additional data sources, such as medical history or genetic information.

In summary, through the development of a dependable and precise software system for disease prediction based on symptoms, this project makes a contribution to the healthcare industry. The effective use of machine learning algorithms and the evaluation of their effectiveness show the potential of this strategy to aid medical professionals in their decision-making.

Moving forward, further research can be conducted to explore advanced feature selection techniques, hyperparameter tuning, and the integration of additional data sources to increase the disease prediction models' reliability and accuracy. Future developments in machine learning and healthcare technology hold considerable potential for the field of disease prediction.

Overall, this project represents a significant step towards the goal of accurate disease prediction, contributing to the advancement of healthcare and ultimately benefiting patients worldwide.

**REFERENCES**

|  |  |
| --- | --- |
| [1] | S. M. a. M. K. Henry Barlow, “Predicting High-Risk Prostate Cancer Using Machine Learning Methods,” vol. 4, no. 3, p. 129, 2019. |
| [2] | M. Srivenkatesh, “Prediction of Prostate Cancer using Machine Learning Algorithms,” *Int. J. Recent Technol. Eng,* vol. 8, no. 5, pp. 5353-5362, 2020. |
| [3] | A. A. B. B. R. K. B. N. a. M. A. B. M. Tahmooresi, “Early Detection of Breast Cancer Using Machine Learning Techniques,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* vol. 10, no. 3-2, pp. 21-27, 2018. |
| [4] | R. Patra, “Prediction of Lung Cancer Using Machine Learning Classifier,” *Computing Science, Communication and Security: First International Conference, COMS2 2020, Gujarat, India,* vol. Revised Selected Papers 1, pp. 132-142, March 26-27 2020. |
| [5] | K. R. J. L. C. H. L. G. Y. C. L. a. X. X. Yijun Wu, “Machine Learning Algorithms for the Prediction of Central Lymph Node Metastasis in Patients With Papillary Thyroid Cancer,” *Frontiers in Endocrinology,* vol. 11, p. 577537, 2020. |
| [6] | B. S. a. C. O. Kubra Tunca, “Lung Cancer Incidence Prediction Using Machine Learning Algorithms,” *Journal of Advances in Information Technology,* vol. 11, no. 2, 2020. |
| [7] | T. D. J. Dr.B.Santhosh Kumar, “Breast Cancer Prediction Using Machine Learning Algorithms,” *International Journal of Advanced Science and Technology ,* vol. 29, no. 3, 2020. |
| [8] | C. M. M. G. A. M. J. A. S.-S. J. R. L. C. C.-H. M. M.-Y. e. a. Chiesa-Estomba, “Machine learning algorithms as a computer-assisted decision tool for oral cancer prognosis and management decisions: a systematic review,” *ORL,* vol. 84, no. 4, pp. 278-288, 2022. |
| [9] | S. L. S. K. W. N.-H. J. K. DongWook Kim, “Deep learning-based survival prediction of oral cancer patients,” *Scientific reports,* vol. 9, no. 1, pp. 1-10, 2019. |
| [10] | M. L. B. ,. M. C. M. I. P. E. M. I. C. N. C. V. Lerina Aversanoa, “Thyroid Disease Treatment prediction with machine learning approaches,” *Procedia Computer Science ,* vol. 192, pp. 1031-1040, 2021. |
| [11] | S. S. V. K. S. S. P. A. K. Kedar Pingale, “Disease Prediction using Machine Learning,” *International Research Journal of Engineering and Technology (IRJET) ,* vol. 6, pp. 831-833, 2019. |