A PROJECT REPORT ON

COURSE ATTAINMENT ANALYSIS USING BLOOM'S LEVEL

Mini project submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
(2020-2024)
BY

Medishetty Sumana	**20241A12F1**
Muchinthala Haritha	**20241A12F5**
Golla Sai Bhavani	**20241A12D5**

Under the Esteemed guidance
of
Dr.Y.Sri Lalitha
Professor
Dept of IT.

DEPARTMENT OF INFORMATION TECHNOLOGY
GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
HYDERABAD

CERTIFICATE

This is to certify that it is a bonafide record of Mini Project work entitled **“COURSE ATTAINMENT ANALYSIS USING BLOOM’S LEVELS”** done by **M. SUMANA (20241A12F1), M. HARITHA (20241A12F5), G. SAI BHAVANI (20241A12D5)** of **B.Tech (IT)** in the Department of Information Technology, **Gokaraju Rangaraju Institute of Engineering and Technology** during the period 2020-2024 in the partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from GRIET, Hyderabad.

Dr.Y.Sri Lalitha,
Professsor.
(Internal guide)

Dr .N.V. Ganapathi Raju ,
Head of the Department,

(Project External)

ACKNOWLEDGMENT

We take immense pleasure in expressing gratitude to our Internal guide**, Dr.Y.Sri Lalitha Professor, Dept. of IT**, GRIET. We express our sincere thanks for her encouragement ,suggestions and support, which provided the impetus and paved the way for the successful completion of the project work.

We wish to express our gratitude to **Dr. N. V. Ganapathi Raju,** HOD IT **,** our Project Coordinators**, P.K Abhilash and B. Kanaka Durga** for their constant support during the project.

We express our sincere thanks to **Dr. Jandhyala N Murthy,** Director, GRIET**,** and **Dr. J. Praveen,** Principal, GRIET**,** for providing us the conductive environment for carrying through our academic schedules and project with ease.

We also take this opportunity to convey our sincere thanks to the teaching and non-teaching staff of GRIET College, Hyderabad.

Name: Medishetty Sumana
Email: sumanamedishetty649@gmail.com
Contact No: 9381400447
Address: Vikarabad, Telangana.

Name: Muchinthala Haritha
Email: muchinthalaharitha111@gmail.com
Contact No: 8520071285
Address : JNTU, Hyderabad.

Name: Golla Sai Bhavani
E-mail: Bhavanigolla323@gmail.com
Contact No: 9652045651
Address: Nalgonda, Telangana.

DECLARATION

This is to certify that the project entitled “**COURSE ATTAINMENT ANALYSIS USING BLOOM’S LEVELS”** is a bonafide work done by us in partial fulfillment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites, books and paper publications are mentioned in the Bibliography.

This work was not submitted earlier at any other University or Institute for the award of any degree.

Medishetty Sumana	**20241A12F1**
Muchinthala Haritha	**20241A12F5**
Golla Sai Bhavani	**20241A12D5**

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Examinations are a significant part of teaching and learning. Questions are used to obtain information and assess knowledge and competence of students. Academicians in higher education use Mid-term and End Semester papers to assess the application skills of students. Question writing process is a very challenging step for academicians. Questions that used to evaluate different cognitive levels of students may be categorized as low, intermediate and Higher order. This work tries to derive a suitable methodology to analyse the course attainment of students in final examinations based on Bloom’s Taxonomy. The study will be conducted to check whether examination questions comply with the requirements of Bloom’s Taxonomy at various cognitive levels. Natural language processing and Machine Learning techniques will be used to identify the suitable cognitive level. The derived model introduces a quantitative approach to categorize undergraduate examination papers.

Examinations are one of the most popular ways for educational institutions to gauge students proficiency in a particular course, so the evaluation of exam questions is essential. As a result, it is imperative to design a fair and excellent exam that caters to students various cognitive abilities. Consequently, a lot of lecturers use Bloom's taxonomy, a popular framework created to evaluate students intellectual prowess and abilities. It has been suggested in a number of works that automatic Bloom's taxonomy question classification be handled. These works categorise questions based on a particular domain for the most part.

Additionally, we will conduct a comprehensive analysis of the course attainment using Bloom's levels. By examining the alignment of exam questions with different cognitive levels and evaluate the effectiveness of the exam in assessing students understanding and application of course material. The application of natural language processing and machine learning techniques will enable the identification and quantitative assessment of the cognitive levels represented in the exam papers.

By integrating the course attainment analysis using Bloom's levels into the classification model, this contributes to a more holistic approach to exam question analysis. The proposed methodology has the potential to provide valuable insights for educators in designing exams that effectively assess student knowledge acquisition and application across various cognitive levels and subject areas.

We used Support Vector Classifier and achieved an accuracy of 74.146% and using Logistic Regression we achieved an accuracy of 72.5%.

Domain(s): Natural Language Processing and Machine Learning.

1. INTRODUCTION

1.1Introduction to the project

The Bloom's Taxonomy concept was developed by Mr. Benjamin Bloom and Mr. Edward Frust and first made public in 1956. This was developed to encourage critical thinking in the classroom. It classifies cognitive and emotional abilities according to their degree of complexity. The taxonomy consists of a hierarchy of more difficult procedures that students may master. Teachers typically utilise Bloom's taxonomy because it offers a pre-made structure for developing learning objectives.

The system known as Bloom's Taxonomy divides educational goals and learning outcomes into many tiers of cognitive complexity. The taxonomy offers educators an organised method for creating efficient learning experiences, suitable examinations, and assisting students in their intellectual development. Six hierarchical levels, each of which reflects a different cognitive process, make up the basic structure of Bloom's Taxonomy. These levels are remembering, comprehending, applying, analysing, evaluating, and producing, and they are listed in escalating order of complexity. Each level improves on the one before it and includes more complex cognitive skills and talents. The fundamental level of memory involves recalling and recognising information. Understanding extends beyond rote learning.

Numerous benefits come with using Bloom's Taxonomy, which also offers educators a useful framework for improving teaching and learning results. Here are a few of the main benefits and how it functions:

Clear Learning Objectives: A clear and organised hierarchy of learning objectives is provided by Bloom's Taxonomy. It aids teachers in outlining precise learning objectives for their pupils, guaranteeing a concentrated and well-planned approach .Teachers can create suitable learning activities, evaluations, and instructional strategies to assist student progress at each cognitive level by having well stated objectives.

Progressive Development: From lower-order to higher-order thinking, the taxonomy shows a progression of cognitive abilities. Students are encouraged to build on their core knowledge and gradually acquire more complicated cognitive skills. It enables teachers to gradually introduce new concepts and structure learning experiences.

Personalised Learning: Using the taxonomy, learning experiences may be tailored to the requirements and aptitudes of certain students. In order to differentiate instruction and offer the right support and challenges, educators must first determine the cognitive level at which each student is functioning. It enables personalised learning opportunities that focus on students areas of strength and growth, encouraging engagement and academic performance. The Bloom's Taxonomy framework aids in the creation and alignment of curriculum and instructional materials. It makes sure that the learning objectives, teaching methods, and evaluations are all consistent with one another and linked. The taxonomy aids educators in creating a coherent curriculum that methodically increases students knowledge and abilities while fostering their overall learning progress.

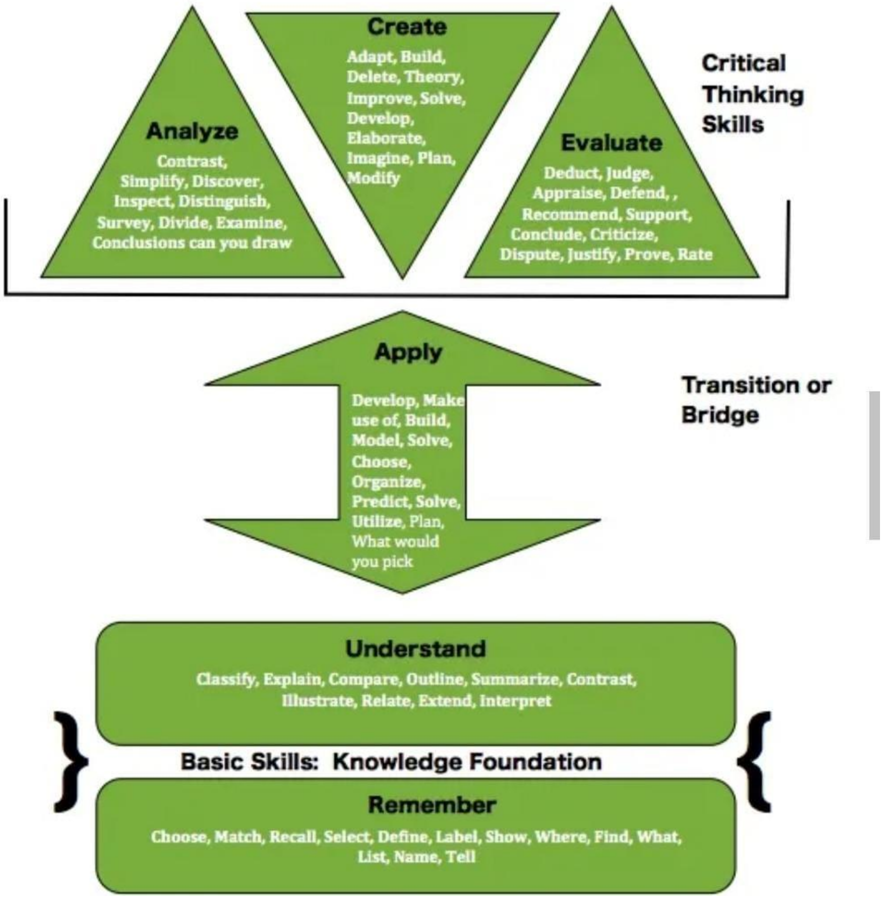

Create
Adapt, Build, Delete, Theory, Improve, Solve, Develop, Elaborate, Imagine, Plan, Modify
Analyze
Contrast, Simplify, Discover, Inspect, Distinguish, Survey, Divide, Examine, Conclusions can you draw
Evaluate
Deduct, Judge, Appraise, Defend, , Recommend, Support, Conclude, Criticize, Dispute, Justify, Prove, Rate
Critical Thinking Skills
Apply
Develop, Make use of, Build, Model, Solve, Choose, Organize, Predict, Solve, Utilize, Plan, What would you pick
Transition or Bridge
Understand
Classify, Explain, Compare, Outline, Summarize, Contrast, Illustrate, Relate, Extend, Interpret
Basic Skills: Knowledge Foundation
Remember
Choose, Match, Recall, Select, Define, Label, Show, Where, Find, What, List, Name, Tell

Figure 1. Bloom's Taxonomy

Educators can create a more dynamic and participative learning environment by planning learning experiences that include tasks and activities associated with the various cognitive levels. Instead of only receiving information, students actively participate in creating their own knowledge and deepening their comprehension of the subject.

The application of Bloom's Taxonomy in course attainment analysis provides a thorough framework for creating instruction, specifying learning goals, monitoring student progress, and encouraging the development of critical thinking abilities. Teachers may make sure that their lesson plans are in line with the intended results and give their students relevant learning opportunities that support their intellectual growth and development by including this taxonomy into the analysis of course attainment.

1.2 Existing System

Earlier people worked on Bloom’s taxonomy by automating the prediction of Bloom’s level for a given question by identifying significant keywords to determine the suitable cognitive level.

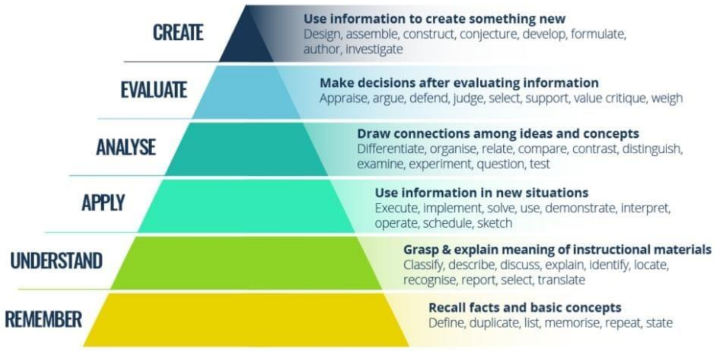

CREATE
Use information to create something new
Design, assemble, construct, conjecture, develop, formulate, author, investigate
EVALUATE
Make decisions after evaluating information
Appraise, argue, defend, judge, select, support, value critique, weigh
ANALYSE
Draw connections among ideas and concepts
Differentiate, organise, relate, compare, contrast, distinguish, examine, experiment, question, test
APPLY
Use information in new situations
Execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
UNDERSTAND
Grasp & explain meaning of instructional materials
Classify, describe, discuss, explain, identify, locate, recognise, report, select, translate
REMEMBER
Recall facts and basic concepts
Define, duplicate, list, memorise, repeat, state

Figure 2: Knowledge Domain Verbs

Disadvantages of Existing System

- **Lack of Contextual Understanding:** Using keywords alone to determine the cognitive level may not fully reflect the context and complexity of the topic. Bloom's Taxonomy states that, both the content and the associated thought processes are taken into account. Automated systems that just use keywords risk missing nuances in meaning and failing to correctly determine the proper cognitive level.
- **Overreliance on Surface Features:** Keyword-based techniques frequently overlook the deeper comprehension and connections necessary for higher-order thinking in favour of surface-level features. They could fail to recognise the fundamental ideas and distinguish between similar terms used at various cognitive levels.

1.3 Proposed System

We analyse the question paper and determine each question's Bloom's level. Bloom’s level is identified using the techniques like Machine Learning and Natural Language Processing. The aim is to analyse the Course Attainment of the question paper of each Bloom’s level. We conclude by visualizing the Bloom’s level in the question paper using a pie chart. We also determine the attainment of the bloom’s level individually using the predicted bloom’s levels and visualize it using a Pie Chart.

Advantages of Proposed System

Effective course attainment analysis: The system permits the analysis of a question paper's course attainment based on Bloom's level. This enables instructors and administrators to analyse the assessment's overall rigour and how well the questions match with the learning objectives.

Time-saving and automation: The system automates the Bloom's level evaluation procedure, which can be laborious to complete manually. Since there is no longer a need for teachers to individually analyse each question, significant time that could be spent on other educational activities is saved.

Improved instructional design: The system offers information on how Bloom's levels are distributed across a test. Any gaps or biases in the assessment can be found using this data, which can also be used to guide decisions about instructional design. This information can be used by teachers to develop more balanced and thorough test questions that fully account for various cognitive abilities.

Visual Representation: The approach provides graphic representations of the Bloom's levels in the question paper and the degree to which each level has been reached. The distribution and attainment of Bloom's levels can be more easily understood and communicated through visuals like pie charts, which supports data-driven discussion and decision-making.

2. REQUIREMENT ENGINEERING

2.1 Software Requirements:

A requirement is a quality or ability that a system or software component must possess in order to address a real-world problem. The problems could include maintaining a device, automating a component of a system, correcting bugs in an existing system, and other problems.

The requirements that are commonly taken into account are divided into three categories: domain requirements, non-functional requirements, and functional requirements.

- Operating System: -Windows
- Technology: - Flask ,Machine Learning ,Natural Language Processing
- Software: - Visual Studio Code

2.2 Hardware Requirements:

Hardware specifications describe the physical elements required for a device or system to operate properly. Depending on the kind of system or device being utilised, additional restrictions may apply. It may consist of peripherals like the CPU, memory, storage, graphics card, and others. To prevent performance issues or compatibility concerns, it's crucial to make sure your hardware complies with the needs of the programme or application you want to use.

- Operating system :- windows
- Processor :-i5 processor
- RAM :- 4GB or 8GB RAM

2.3 Functional Requirements: -

Prediction of Bloom's Levels: The system should be able to predict the cognitive level (Bloom's level) of a given question by analysing its content and characteristics. This prediction will be useful for analysing and visualizing the distribution of cognitive levels in the question paper.

Visualization of Individual Attainment: The system should calculate and display the percentage or frequency of questions belonging to each cognitive level. This will provide a breakdown of how many questions are categorized under each Bloom's level.

User-Friendly Interface: The system should have an interface that is easy for users to navigate. Users should be able to input the question paper, view the analysis results, and access visualizations without any confusion. The interface should be intuitive and user-friendly.

Reporting and Exporting: The system should allow users to generate reports summarizing the analysis of Bloom's levels and the attainment results. It should also support exporting the visualizations and analysis data in commonly used formats so that users can further analyse or share the information easily.

3. LITERATURE SURVEY

[1] This thesis addresses the problem of categorising questions posted on well-known social networks like Stack Overflow (SO), with a particular emphasis on the field of knowledge management. The abundance of technical questions and answers on SO, which include popular discussions and can be a useful resource for the education industry, is what gave rise to this issue.

The thesis uses Latent Dirichlet Allocation (LDA) to reduce the dimensions of each document from SO in order to address this issue. To achieve the desired classification outcomes, the k-means algorithm is then applied to a set of documents that contains both unlabelled and labelled data. With LDA and k-means, the initial strategy achieves an accuracy of 30.2%.

The feature set is expanded with new features to improve classification performance. Each question's score, answer count, and views are among these features. The accuracy of the classification is increased to 56.33% by adding these extra attributes to the feature set.

This review of the literature focuses on the investigation into the categorization of questions from social networks like Stack Overflow. An initial method of addressing the issue that achieves a certain level of accuracy is the use of LDA and k-means. But the classification accuracy is significantly increased by adding extra features like score, answer count, and view count. By offering knowledge about how to efficiently categorise queries from well-known social networks, this research advances the field of knowledge management while ultimately assisting the education sector.

[2] The cognitive level of student questions is frequently determined using Bloom's taxonomy. The manual identification of these levels can take some time, though. In order to meet this challenge, the field of programming known as Natural Language Processing (NLP), which focuses on processing natural language, has emerged.

Tokenization, which entails dividing lengthy passages of text into smaller components known as tokens, is a crucial step in NLP. Words, phrases, or even entire sentences can serve as tokens. Understanding the context of the language and creating NLP models depend greatly on tokenization. Tokenization methods and databases are offered by a number of libraries, including NLTK, Gensim , and Keras.

The manual process of determining Bloom's taxonomy levels in problems is automated by the method presented in this paper. This method speeds up the identification process by utilising tokenization and other NLP techniques. By precisely identifying the cognitive domain level of the questions, instructors can gather understanding about how students intellectual capacities and knowledge are growing.

In conclusion, the paper emphasises the value of Bloom's taxonomy in determining cognitive levels, the applicability of NLP in processing natural language, and the significance of tokenization as a pre-processing step. The suggested method for automating the determination of cognitive domain levels provides a workable method that enables trainers to gather precise information for efficient instruction and evaluation.

[3] To evaluate the complexity of questions, it is suggested that Bloom's taxonomy be applied in question-answering systems. A framework for gauging question complexity and assessing learning levels is provided by Bloom's taxonomy, which was initially developed in pedagogy. The determination of question complexity can help identify reasoning chains and gauge the sufficiency of answers by incorporating Bloom's taxonomy into question-answering systems.

The revised Bloom's taxonomy is used, where question indicator words from educational studies are integrated, to close the gap between pedagogy and question-answering systems. Bloom used keywords to estimate question complexity for each category in the taxonomy matrix. The question dataset has about 770 samples, 90% of which are used for training and 10% for validation.

The samples are divided into n-grams, where n is a number between 1 and 4, and the resulting sequences are then transformed into vectors to process the questions. Using tf-idf (term frequency-inverse document frequency), a scoring method that takes into account the frequency of terms in a document relative to their occurrence across the entire dataset, the significance of these vectors is assessed.

This literature study concludes by recommending the use of Bloom's taxonomy to assess the difficulty of questions in question-answering systems. The complexity of the questions can be determined by incorporating educational keywords into the taxonomy, which enables the system to provide suitable answers and reasoning chains. The study makes use of a question dataset and processes and scores question vectors using methods like n-grams and tf-idf.

[4] An NLP algorithm was created in the context of iSTART, an intelligent tutoring system designed to teach reading strategies, to assess the calibre of questions created by students. This algorithm coded a four-level taxonomy applied to a corpus of 4575 questions. Using machine learning methods and calculating NLP indices for each question, the algorithm was able to predict the quality of the questions.

Several NLP tools, including the Constructed Response Analysis Tool (CRAT), the Tool for the Automatic Analysis of Lexical Sophistication (TAALES), and the Tool for the Automatic Analysis of Cohesion (TAACO), were used to evaluate the question quality. These tools offered indices that were used to forecast human scores for the question corpus, which ranged from 1 to 4. Non-normally distributed indices were excluded from the analysis.

The NLP indicators were treated as dependent variables in a MANOVA (Multivariate Analysis of Variance), while the four question categories were treated as independent variables.to examine the relationship between the NLP indices and the question categories. A significant correlation between the NLP indices and the question categories was found by the MANOVA results.

In addition, a Discriminant Function Analysis (DFA) was carried out, and the results produced a set of 28 variables that successfully differentiated between various question types. The DFA was reported to have had a 61.19% accuracy rate in predicting question quality.

This literature review focuses on the creation and use of an NLP algorithm for iSTART's question quality evaluation. The algorithm shows promising results in predicting the depth and quality of student-generated questions by utilising NLP indices and machine learning techniques, adding to the efficiency of the intelligent tutoring system.

4. TECHNOLOGY

4.1 ABOUT NATURAL LANGUAGE PROCESSING:

The branch of "artificial intelligence" (AI) known as "natural language processing" (NLP) is more specifically the area of computer science that aims to enable machines to grasp spoken and written language similarly to people.

NLP blends computational linguistics rule-based human language modelling with statistical, machine learning, and deep learning models. Together, these technologies enable computers to fully "understand" human language, including the speaker's or writer's intention and sentiment, whether it takes the form of text or speech data.

Computer systems that translate text between languages, respond to spoken requests, and summarise vast volumes of text quickly—even in real time—are all powered by NLP. Digital assistants, chatbots for customer care, speech-to-text dictation software, voice-activated GPS devices, and other consumer conveniences all make use of NLP. However, NLP is also playing a larger part in corporate solutions that assist reorganise workspaces, increase worker productivity, and simplify difficult but essential company processes.

ADVANTAGES OF NATURAL LANGUAGE PROCESSING:

- The response time for customer assistance is quicker.
- The NLP system gives accurate responses to the questions without adding extra or unnecessary information.
- The more relevant the information in the query, the more precise the answer. Organising a significant amount of unstructured data.
- Users may ask inquiries on any subject and quickly get a direct response.
- It is simple to apply.
- Employing a person is more expensive than using a programme.The activities listed above could take a person two or three times as long to perform as they would on a machine.
- The NLP system questions with natural language responses.
- Permit you to carry out more fair and consistent language-based data comparisons to individuals without becoming weary. The NLP technique scales other activities involving language and enables a machine to speak with a human in their language.

DISADVANTAGES OF NATURAL PROCESSING LANGUAGE:

- The NLP system's user interface contains elements that let people engage with it more.
- Building a model from scratch without utilising a pre-trained model may need a week to function well, depending on the volume of data. Due to its constrained functionality, the system is only designed to perform a single, narrowly defined task.

- If a question is poorly phrased or ambiguous, the system might not be able to respond with the right response in complex query language.
- It is never completely dependable; it is never 100% reliable. Errors in its results and prediction are conceivable.

APPLICATIONS OF NATURAL LANGUAGE PROCESSING:

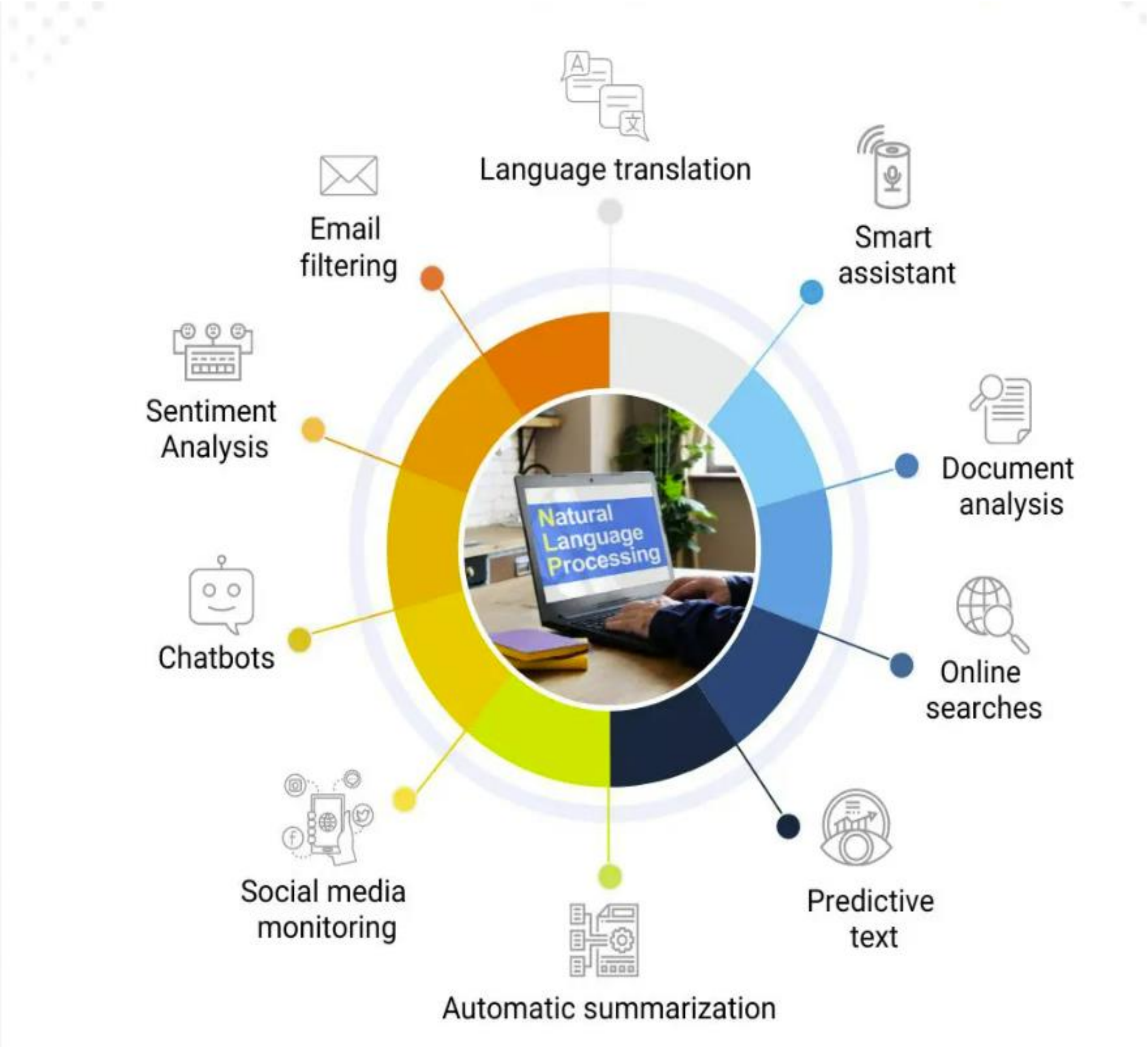


Figure 3: Applications of NLP

- Email filtering
- Language translation
- Smart assistant
- Document analysis
- Online searches
- Predictive text
- Automatic summarization
- Sentiment analysis

4.2 INTRODUCTION OF FLASK:

Web applications can be created using the Python Flask API. Armin Ronacher was the person who created it. Because less basic code is needed to create a straightforward web application with Flask than it is with Django, it is more explicit and simpler to understand. An organisation of modules and libraries, often known as a web framework or web application framework, enables programmers to create applications without having to write low-level code for things like thread management and protocols. The Jinja2 template engine and the WSGI (Web Server Gateway Interface) toolset are the foundations of Flask.

4.2.1 WSGI:

The Web Server Gateway Interface (WSGI, sometimes called the Web Server Gateway Interface) has historically been used in Python web application development. WSGI specifies a common interface for web servers and web applications.

4.2.2 Werkzeug:

The Werkzeug WSGI toolkit implements requests, response objects, and utility functions. A web frame may now be constructed on top of it. One of the pillars of the Flask framework is Werkzeg.

4.2.3 jinja2:

The Python template engine jinja2 is popular. A dynamic web page is produced by a web template system by fusing a template with a particular data source.

APPLICATIONS OF FLASK FRAMEWORK:

- A rapid debugger is available, and a development server is built-in.
- Lightweight
- Secure cookies are supported.
- Jinja2 template creation.
- REST-based request dispatching.
- There is built-in support for unit testing.

4.3 ABOUT PYTHON:

Python is a powerful, interactive, object-oriented, and interpreted scripting language. Python was created to be very readable. It has fewer syntactical structures than other languages and typically employs English terms rather than punctuation.

- **Python is Interpreted :** The interpreter processes Python at runtime. Your programme does not need to be compiled before running. This is comparable to PHP and PERL.
- **Python is Interactive :** To write your programmes, you can actually sit at a Python prompt and communicate with the interpreter directly.
- Python's support for object-oriented programming, which wraps code into objects, makes it an object-oriented language.
- Python is an excellent tool for beginners to learn and facilitates the creation of a broad range of programmes, including simple text processing, web browsers, and games.

FEATURES OF PYTHON:

- Simple to learn : Easy to learn Python has a limited set of keywords, a clear structure, and a short syntax. This helps the pupil take up the language.
- Readability : Python code that is easy to comprehend is more readable and clearly described.
- Simple to maintain : Python's source code is not difficult to keep up with.
- A huge standard library : The majority of Python's library consists of a sizable standard library that is extremely portable and interoperable with UNIX, Windows, and Macintosh systems.
- Python features an interactive mode that makes it possible to test and debug code snippets while they are being performed.
- Python is extremely portable and provides the same user interface across all hardware platforms.
- Expandable : The Python interpreter may be expanded by adding low-level modules.

ADVANTAGES OF PYTHON:

- Functional and structured programming approaches are supported in addition to OOP.
- By compiling it to bytecode or utilising it as a scripting language, it is possible to develop large programmes.
- It supports extremely high-level dynamic data types and allows dynamic type verification.
- Automatic garbage collection is supported.
- It is simple to integrate with Java, C, C++, ActiveX, and CORBA.
- Extensive support libraries (Pandas for data analytics, NumPy for numerical calculations, etc.).
- Open source and a sizable, vibrant community.

- Flexible and simple to read, learn, and write .
- comprehensible data structures.
- Programming language that is procedural and object-oriented.
- Interactive and portable.
- Extremely effective Since Python has a clear object-oriented design and excellent text processing and integration features in addition to having a built-in unit testing framework, it is more effective.
- Opportunities with the Internet of Things (IoT)
- Translated Language

4.4 MACHINE LEARNING

The subject area known as machine learning has enabled computers to learn without explicit programming. Machine learning is one of the most fascinating technologies. As implied by the name, it gives the computer the ability to learn, enabling it to become more human-like. Machine learning is currently in use, maybe much more often than one might imagine.

The Seven Steps of Machine Learning are as follows:

collecting data
Preparing that data
selecting a model
Training
Evaluation
Hyperparameter Tuning
Prediction

Classification of Machine Learning

Machine learning can be broadly divided into three categories:

Supervised Machine Learning
Unsupervised Machine Learning
Semi-supervised Learning
Reinforcement Learning

Supervised Machine Learning:

Models are trained using labelled training data in supervised learning, a method of machine learning. The input features and accompanying output labels that make up the training data give the model explicit information it needs to understand patterns and correlations. The model examines the labelled data during training to produce precise predictions or classifications on unobserved data. This method is frequently utilised in many fields, including sentiment analysis, speech recognition, and image recognition. Accuracy, precision, recall, and F1 score are among the metrics used to evaluate the effectiveness of supervised learning models. Popular supervised learning techniques include decision trees, logistic regression, neural networks, and linear regression. However, obtaining the necessary amount of labelled data for the training of a supervised learning model can be time-consuming and expensive. The labelled data's quality and representativeness have a big impact on the model's performance. Thus, acquiring a balanced and diverse labelled dataset is essential for creating reliable supervised learning models.

Unsupervised learning

A potent machine learning method called unsupervised learning works without labelled data. Algorithms are instead entrusted with figuring out structures, relationships, or patterns in unlabelled data. When there is no specific advice or supervision available, this strategy is especially helpful. For a variety of applications, including as clustering, anomaly detection, dimensionality reduction, and generative modelling, unsupervised learning methods are used. By putting comparable data points together based on their intrinsic qualities, clustering algorithms make it possible to spot natural clusters within the data. Anomaly detection algorithms concentrate on finding uncommon or infrequent occurrences that dramatically depart from the norm. The goal of dimensionality reduction approaches is to save crucial information while reducing the complexity of high-dimensional data.

Reinforcement Learning

Machine learning's reinforcement learning (RL) sector focuses on the creation of algorithms and methods that teach an agent how to make decisions sequentially in a way that maximises a reward signal. It draws inspiration from the way that both people and animals learn through making mistakes.

In real life, an agent engages with the environment, takes actions, and is rewarded or punished for those activities. The agent's objective is to discover a course of action or set of policies that maximises the cumulative reward over time. In contrast to supervised learning, where the agent is given examples with labels, RL depends on exploration and exploitation to find the best tactics.

Applications of Machine Learning

1. Facial recognition/Image recognition
2. Automatic Speech Recognition
3. Financial Services
4. Marketing and Sales
5. Healthcare
6. Recommendation Systems

Figure 4: Machine Learning

Advantages:

It can be used for pattern detection.
It can be used to make predictions about future data.
It can be used to automatically create new features from data.
It can be used to automatically cluster data.
It can be used to automatically find outliers in data.

Disadvantages:

A few disadvantages include the potential for biased data, overfitting data, and lack of explainability.

4.5 ALGORITHMS

SUPPORT VECTOR MACHINE

Support Vector Classifier:

Popular machine learning algorithms for classification tasks include the Support Vector Classifier, or SVC, algorithm. It is a member of the Support Vector Machines (SVMs) family of algorithms, which can be applied to both classification and regression analysis.

Finding an ideal hyperplane that successfully divides the data points into various classes is the main goal of SVC. The SVC algorithm learns from a set of text samples and their corresponding labels to classify new text samples into various categories based on their features when it comes to text classification, such as predicting the Bloom's level of a question. Using a linear kernel, the SVC algorithm operates. The decision boundary that divides the classes in the data is the kernel. A linear kernel presupposes that the data can be divided into two groups by a hyperplane or a straight line in the feature space. In essence, it is based on the supposition that the relationship between the features and the classes being classified is linear. It's crucial to remember that different kernel options, such as polynomial or radial basis function kernels, can also be used, based on the specific problem at hand, the type of data, and the requirements.

Based on the features extracted from the text, the SVC algorithm will try to identify the best hyperplane that can effectively separate the various Bloom's levels.

Ultimately, the kernel selection is significant because it affects the decision boundary's shape, which has a direct impact on the SVC algorithm's precision and effectiveness in classifying the questions' Bloom's levels. Using the SVC classifier we got an accuracy of 74.16%.

LOGISTIC REGRESSION

Multinomial logistic regression:

For target variables that are binary or dichotomous, logistic regression is frequently utilised. When the target variable includes more than two categories, a variant termed multinomial logistic regression—also referred to as polytomous logistic regression or nominal logistic regression—can be utilised.

In order to handle categorical outcomes with more than two levels, multinomial logistic regression expands the concepts of logistic regression. The link between several independent factors and a categorical dependent variable with three or unordered categories is modelled using this particular sort of generalised linear model.

The objective of multinomial logistic regression is to estimate the probabilities of each category in relation to a selected reference category. The dependent variable is split into several categories (K categories).

The model compares the probabilities of each category to the reference category using a set of regression equations, one for each category.

Multinomial logistic regression's equation can be written as:

$\text{logit}(P(Y = k)) = \beta k0 + \beta k1X1 + \beta k2X2 + ... + \beta knXn$

Where k, $\beta k0$ is the intercept term for category k, $\beta k1$, $\beta k2$, ..., βkn are the coefficients corresponding to each predictor variable for category k, and X1, X2,..., Xn are the values of the predictor variables, logit(P(Y = k)) represents the natural logarithm of the odds of the dependent variable being in category.

When all other variables are maintained constant, the coefficients (β) in multinomial logistic regression describe the change in the log odds of the dependent variable falling into category k associated with a one-unit change in the corresponding predictor variable.

Maximum likelihood estimation, which aims to identify the set of coefficients that maximise the probability of the observed data, is used to estimate the model.

In multinomial logistic regression, the odds ratios for each category in relation to the reference category must be compared in order to interpret the coefficients. In comparison to the reference category, these odds ratios represent the multiplicative change in the probability of falling into a given category for a one-unit change in the predictor variable.

In conclusion, when the target variable contains more than two categories, multinomial logistic regression is utilised. In order to model the link between various variables and a categorical outcome, it expands the concepts of logistic regression. The model calculates each category's probability in relation to a reference category, enabling the interpretation of the effects of predictor variables on the categorical outcome. Using logistic regression we achieved an accuracy of 72.5%.

5.DESIGN REQUIRMENT ENGINEERING

CONCEPT OF UML:

A standard modelling language made up of numerous related diagrams is called Unified Modelling Language, or UML. It was created to aid in the definition, visualisation, construction, and documentation of the artefacts of software systems as well as business models and other non-software systems by system and software engineers. The best technical techniques for simulating huge, complicated systems are combined in the UML. The construction of object-oriented software and the software development process both strongly rely on the UML. The UML primarily employs graphical notations to convey a software project's design. Project teams can consult, investigate potential designs, and validate the software's architectural design using the UML.

UML DIAGRAMS:

5.1 ACTIVITY DIAGRAM:

The UML activity diagram is used to show the system's control flow rather than actually implementing it. Both concurrent and sequential activity models are used.

The flow of work from one action to the next can be seen by using the activity diagram. It emphasises the presence of flow and the sequence in which it takes place . The activity diagram has been designed with a fork, join, etc. to deal with the various types of flows, which can be sequential, branching, or concurrent.

It is also known as an object-oriented flowchart. It contains tasks that call for using a string of methods or actions to mimic the behavioural diagram

- **Activity:** Represented by rounded rectangles, activities depict specific actions or tasks within the system. They can represent operations, processes, computations, or other behaviours. Activities are labelled with a descriptive name.
- **Control Flow:** Represented by arrows, control flows show the sequential order of activities. They depict the flow of control from one activity to another, indicating the order in which activities are performed. The arrows indicate the direction of flow.
- **Decision Node:** Represented by a diamond shape, a decision node represents a point in the workflow where a decision needs to be made. It represents a condition that determines which path to follow based on the evaluation of the condition. The decision node has multiple outgoing control flows, each labelled with a condition.
- **Merge Node:** Represented by a diamond shape with a "+" symbol inside, a merge node is used to synchronize multiple control flows into a single flow. It represents the merging of multiple paths back into a single flow of control.
- **Initial Node:** Represented by a solid circle, an initial node indicates the starting point of the activity diagram. It shows where the control flow begins.

- **Final Node:** Represented by a bullseye symbol, a final node represents the end or completion point of the activity diagram. It indicates the termination of the workflow.
- **Fork and Join Nodes:** Represented by a bar shape with multiple incoming or outgoing control flows, fork and join nodes are used to indicate concurrency or parallelism in the workflow. Fork nodes split the control flow into multiple parallel paths, while join nodes synchronize and merge the parallel paths back into a single flow.
- **Activity diagrams** are useful for modelling and understanding complex workflows, business processes, or system behaviours. They help in visualizing the steps involved, the decision points, and the control flow within a system. Activity diagrams facilitate communication among stakeholders, aid in software design and analysis, and can be used as a basis for implementing system behaviour in software development.

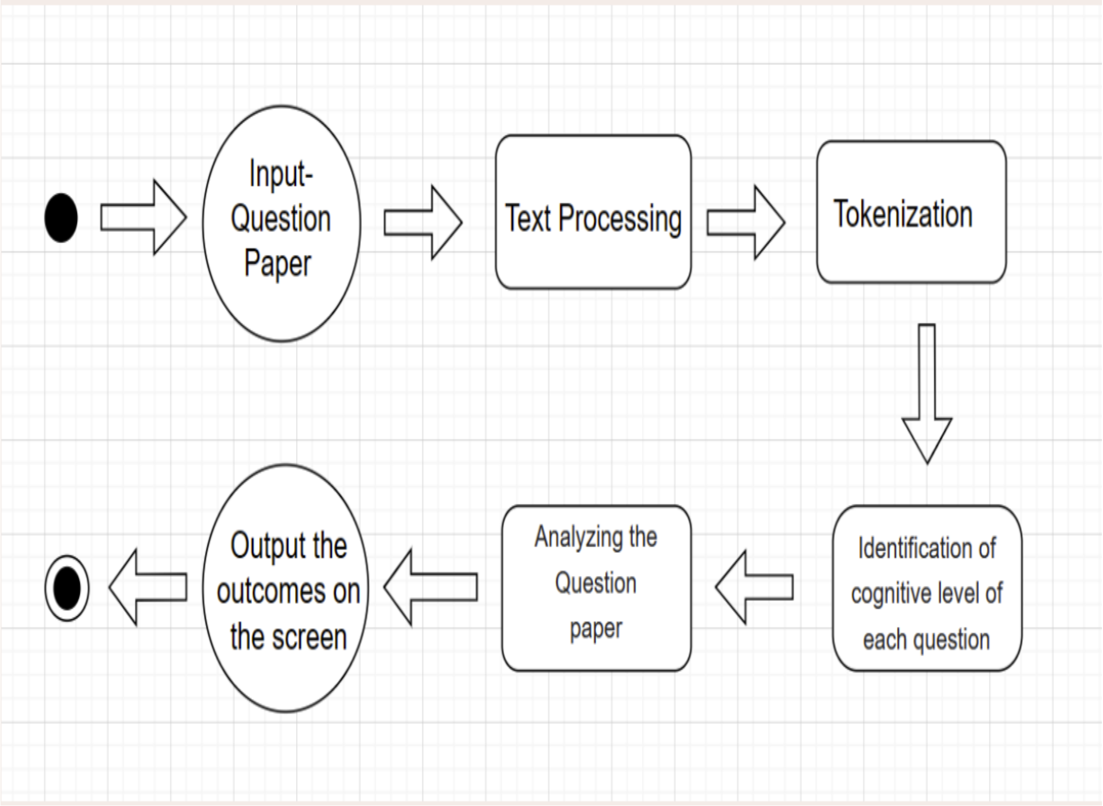

Input-Question Paper
Text Processing
Tokenization
Identification of cognitive level of each question
Analyzing the Question paper
Output the outcomes on the screen

Figure 5: Activity Diagram

<u>5.2 USE CASE DIAGRAM</u>:

In UML, use case diagrams help to represent system behaviours and capture system requirements.

A use case diagram illustrates the high-level functionality and scope of the system. These diagrams also show how the system and its actors interact with one another. The use cases and actors in a use case diagram depict what the system does and how users engage with it, but not how the system fundamentally works.

A use case diagram displays and defines the context and needs of a complete system or substantial pieces of one. A simple system can be represented by a single use case diagram, or multiple use case diagrams can be created to illustrate the various components of your system. A use case diagram is typically created early on in the project and used as a reference throughout the development phase.

An approach used in software development and system analysis to record and explain a system's functional needs from the viewpoint of its users or actors is called a "use case" . It gives a thorough explanation of how users interact with the system to carry out particular objectives or tasks. Use cases are frequently employed in system modelling, software design, and requirement collection.

Typically, a use case includes the following components:

An actor stands in for a user or any other outside party that communicates with the system. Actors might be living things, different systems, or even physical objects. They are described in terms of the duties they perform within the system.

Use Case: From the viewpoint of the user, a use case depicts a certain system functionality or behaviour.

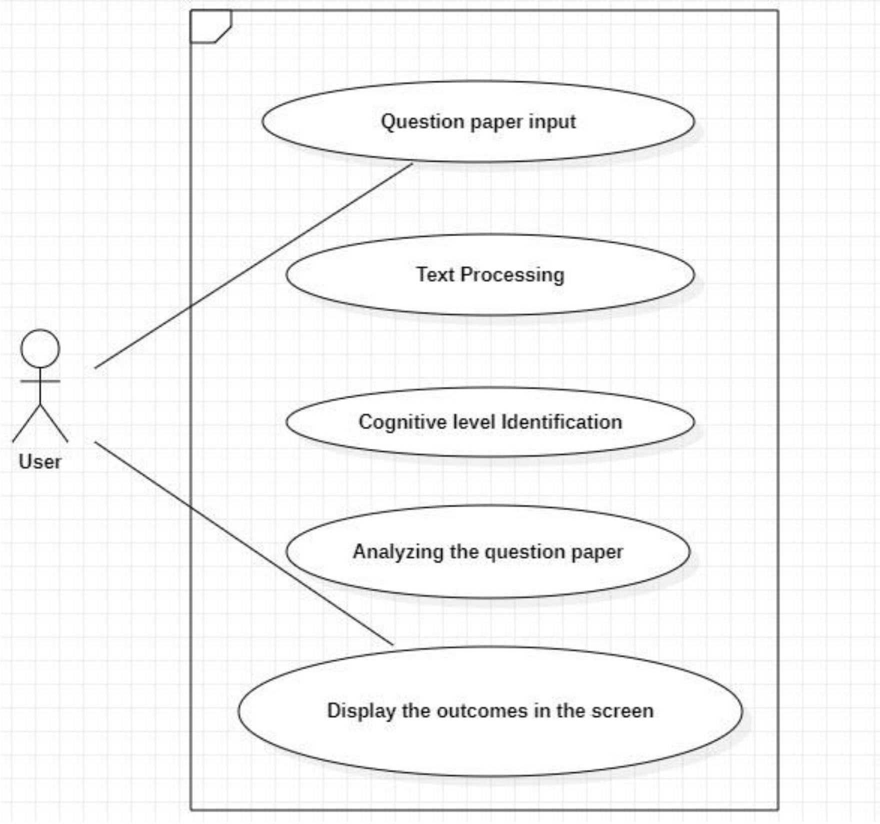

Question paper input
Text Processing
User
Cognitive level Identification
Analyzing the question paper
Display the outcomes in the screen

Figure 6: Use Case Diagram

5.3 SEQUENCE DIAGRAM:

A sequence diagram simply shows the orderly progression of events as they relate to one another. Sequence diagrams are also known as events diagrams and event scenarios. Sequence diagrams show the operations carried out by a system's components in a time-based manner. Software developers regularly use these diagrams to document and comprehend the requirements for new and current systems. There are four main parts to a DFD:

Process: A process, symbolised by a rectangle, denotes a particular action or transformation that occurs within the system. It might involve a calculation, data processing, or some other action.

A sequence diagram is a type of interaction diagram that demonstrates the relationship between different system components in a given order. By displaying the flow of messages or method calls between objects across time, it emphasises the order in which these interactions occur. System analysis, software design, and system behaviour modelling typically use sequence diagrams.

Usually present in a sequence diagram are the following components:

Vertical dashed lines, or "lifelines," are used to depict the items or components involved in the series of interactions. Each lifeline extends vertically to represent the lifespan of a particular object or component and is associated with a single lifeline.

Activation Boxes: Depicted on the lifelines as horizontal rectangles, activation boxes display the amount of time that an object is active.

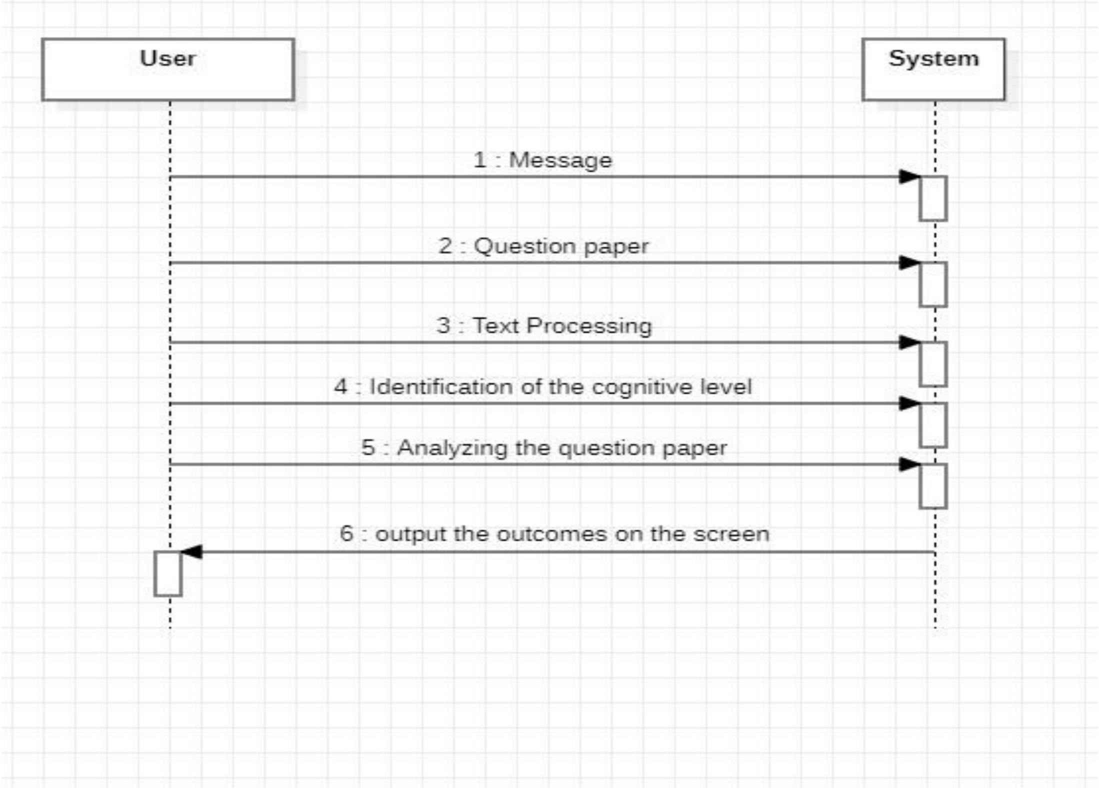

User
System
1 : Message
2 : Question paper
3 : Text Processing
4 : Identification of the cognitive level
5 : Analyzing the question paper
6 : output the outcomes on the screen

Figure 7: Sequence Diagram

5.4 DATA FLOW DIAGRAM:

A DFD is a diagram that shows data flow. A DFD shows the data flow of a system or process. Additionally, for each entity, it clarifies the inputs, outputs, and process itself. DFD does not employ loops, decision-making rules, or control flows. A flowchart can describe particular processes depending on the type of data.

It is a visual tool that may be used to communicate graphically with clients, supervisors, and coworkers. It can be used to examine both brand-new and old systems.

It provides a quick explanation of

- how the system handles data processing.
- What alterations are made.
- what data is retained.
- what results are produced, etc.

A data flow diagram can be represented in many different ways. A modelling tool for structured analysis is the DFD. Data flow diagrams are widely used because they make it simpler to comprehend the important processes and data involved in software-system operations.

A data flow diagram (DFD) is a visual representation of data flow within a system. It provides examples of the data input, processing, and output of various system components. In order to visualise and understand data flow and system processes, DFDs are frequently used in systems analysis and software engineering.

Main parts of a DFD:

Process: A process, symbolised by a rectangle, denotes a particular action or transformation that occurs within the system. It might involve a calculation, data processing, or some other action.

Data Flow: The transfer of data between external entities, processes, and data stores is represented by an arrow symbol that indicates the direction of the flow. This data may be given orally, in writing, or electronically. Input and output data flows are marked in accordance with the type of data, the associated process, or the data repository, and this name is printed next to the arrow.

Data store: A data store doesn't perform any operations; it only saves information for access in the future. A batch of documents that are briefly retained while they wait to be processed or files that are kept for an extended period of time are both examples of data stores. Input flows to a data storage comprise information or activities that affect the stored data. Data that is retrieved from the store would flow as output.

External Entity: The data sources and consumers that travel between the entity and the system being diagrammed are known as external entities, also known as actors, sources, sinks, and terminators. The inputs and outputs of the DFD are provided via these data flows. Since they are not a part of the system being examined, these entities are frequently shown at the diagram's margins. They could represent another system or display a subsystem.

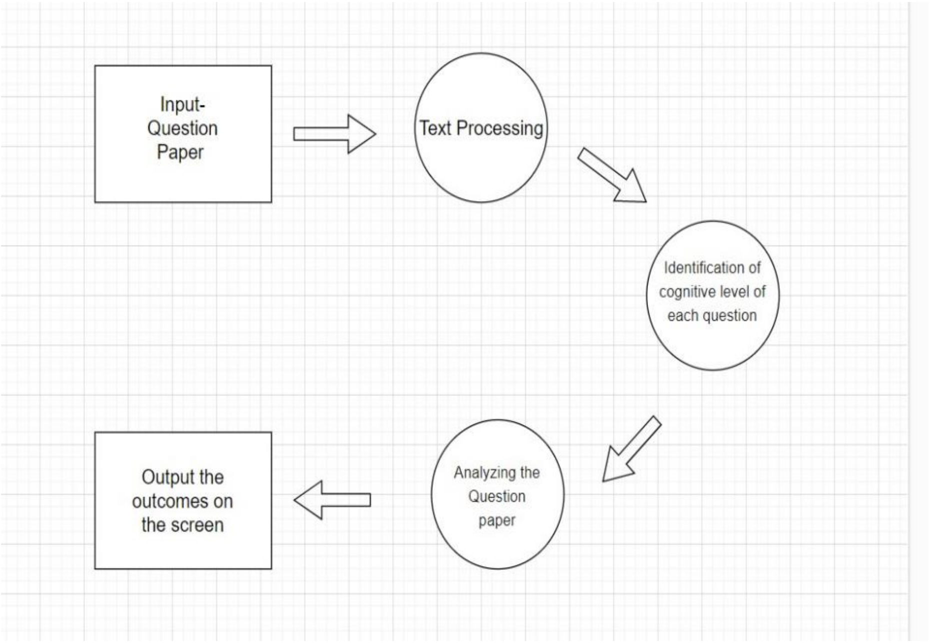


Figure 8: Data Flow Diagram

5.5 SYSTEM ARCHITECTURE:

System architecture is a conceptual model that defines the organisation and operation of numerous components and subsystems, including network devices, hardware, software, and even the machinery of other systems. The design Description Language (ADL) is crucial in describing the overall system design. So, it is a much bigger subject. Decentralised and centralised architectural organisations are two major categories for system architecture.

The design and arrangement of a complex system, including its constituent parts, their interactions, and the overall structure, is referred to as system architecture. It offers a guide for creating, implementing, and operating the system.

System architecture in the context of software systems often entails creating the overall structure of an application or a group of applications that collaborate to satisfy a particular set of requirements. It outlines the connections between various parts, including user interfaces, modules, libraries, databases, and external systems. The technologies, data formats, and communication protocols that are employed to make these interactions possible are also included in the architecture.

There are many kinds of system architectures, one of the system architecture is Monolithic Architecture: Using this method, the system is constructed as a single, intricately connected unit. An application is created as a single, self-contained unit using the classic software design approach known as monolithic architecture. The components are tightly connected and run on a single platform with a single codebase. The architecture streamlines application deployment and monitoring by providing centralised control and management. Due to in-memory function calls, it has a decent performance but may have scalability issues as the application expands. Comparatively to distributed systems, monolithic architectures are easier to create, test, and maintain. However, altering one part can need redeploying the entire application. Agility may be hampered and deployment timelines lengthened as a result. Monolithic designs can be useful for smaller or less complex projects but are less appropriate for extremely complex and rapidly changing applications.

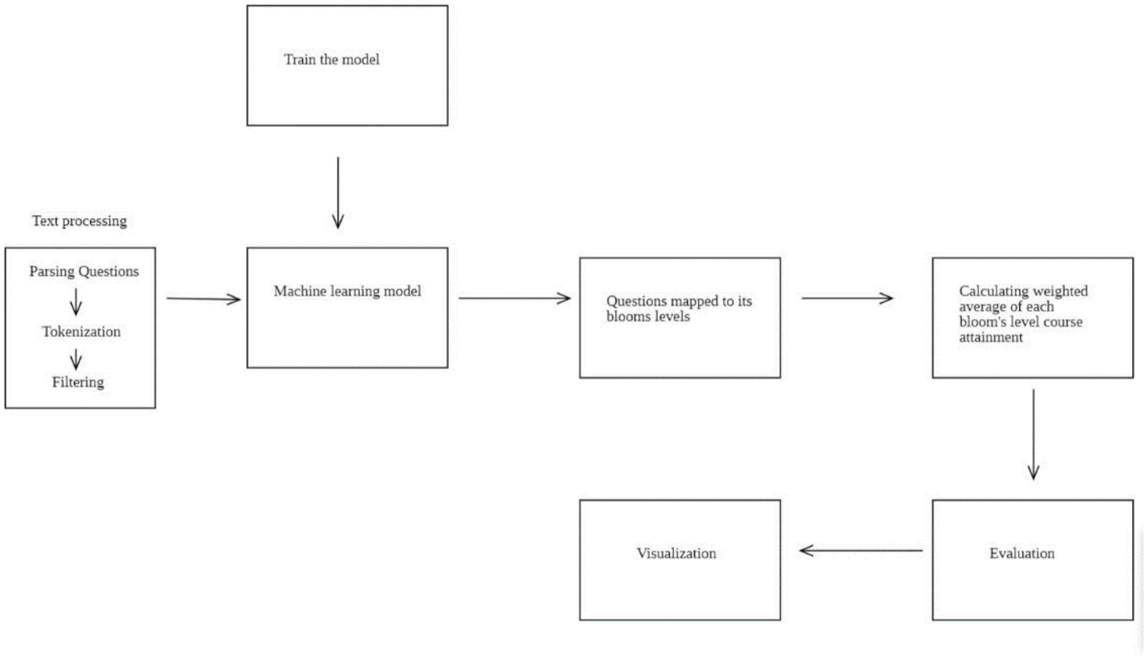


Figure 9: System Architecture

6. IMPLEMENTATION

6.1 Modules

To implement this project , the following modules are used.

Using Flask:

We utilized Flask, a Python web framework, to create a web application. Flask allowed us to build the user interface and handle the file input and submission.

Training the Model:

We trained a machine learning model using the Support Vector Classifier (SVC) algorithm. The purpose of training the model is to predict the Bloom's levels based on the input data.

User Interface:

Within the Flask application, We designed a user interface that allows users to upload a file. The interface provides a way where users can submit their files for analysis.

Updating the File with Predicted Bloom's Levels:

After receiving the file submission, I processed it using the trained SVC classifier. The classifier predicted the Bloom's levels for the given data. I then updated the original file with the predicted levels, making it more informative.

Visualizing the Bloom's Levels and Attainments:

Using the Flask application, We generated pie charts to visually represent the Bloom's levels. Additionally, We conducted an analysis on course attainments, likely using the predicted levels.

By implementing these modules in Flask and leveraging the SVC classifier, We created a web-based application that offers a user interface for file input, predicts Bloom's levels, updates the original file, and provides visualizations and analysis related to Bloom's levels and course attainments.

Modules used:

Numpy:

NumPy is the name of the primary Python library used for numerical computing.. It offers an effective N-dimensional array object that makes it possible to store and manipulate sizable, homogeneous datasets in an effective manner. Elements-wise calculations, linear algebra operations, and Fourier transforms are just a few examples of the mathematical and logical operations that NumPy allows you to carry out on arrays. It provides a wide range of mathematical operations for manipulating arrays and performing complex calculations. Because NumPy's array operations are significantly faster than those of standard Python lists, it is a crucial tool for data analysis and scientific computing. The library also has tools for sorting, array indexing, cutting, and reshaping. It offers thorough documentation and a thriving community, guaranteeing users a wealth of resources and support.

Matplotlib:

A well-liked Python plotting library called Matplotlib offers a variety of customizable visualisations. One can make many different kinds of plots with it, including line plots, scatter plots, bar plots, histograms, and more. For making static, animated, and interactive visualisations, An intuitive interface is provided by Matplotlib. There are numerous customising possibilities available , allowing you to manage the colours, labels, axes, and annotations as well as every other aspect of the plot's appearance. Data from these libraries can be quickly and easily visualised thanks to Matplotlib's seamless integration with NumPy and Pandas. The library is compatible with a number of output formats, including vector graphics, interactive displays, and image files. For the purposes of scientific research, data exploration, and presentations, Matplotlib enables you to produce visualisations of publication-quality. Its thorough documentation and a sizable collection of examples and tutorials, makes it accessible to users of all skill levels.

Pandas:

Python has a robust data manipulation library called Pandas. To manage and analyse structured data, it offers data structures like Series and Data Frame. Pandas is a go-to tool for data cleaning and pre-processing because it makes it simple to load, manipulate, and transform datasets. You can gain valuable insights from your data by using the library's many functions for data selection, filtering, and transformation. It can read and write data in a variety of file kinds, including CSV, Excel, and SQL databases. Pandas is a popular choice for data analysis and manipulation in projects involving data science since it interfaces with other libraries without much difficulty. Additionally, it offers tools for grouping, combining, and aggregating data from various datasets. Excellent time series handling support is provided by Pandas. With its built-in tools for date and time manipulation, resampling, and time-based indexing, Pandas provides excellent support for handling time series data. The library enables effective handling of duplicate values and missing data. Its indexing and slicing capabilities make data subsets simple to access and manipulate. Pandas has a sizable community and extensive documentation, making it simple to locate materials and request assistance when necessary.

Flask:

Python has a simple and adaptable web framework called Flask. It offers a straightforward yet effective method for creating web applications. With Flask, you can handle HTTP requests, render dynamic HTML templates, and quickly configure routes. It complies with the WSGI (Web Server Gateway Interface) standard, enabling seamless web server integration. Model-View-Controller (MVC) design pattern support in Flask makes it simple to divide concerns and keep a clean codebase. Because the framework includes a built-in development server, testing and debugging applications locally is simple. For purposes like authentication, database integration, and form handling, Flask offers a wide variety of extensions and plugins. It has a straightforward API that makes it simple for developers to get started and concentrate on creating their application's logic.

Docx:

A Python package called the docx library makes it possible to write, read, and work with Microsoft Word (.docx) files. It offers a practical and simple user interface for programmatically working with Word documents. With docx, it is possible to make new Word documents, edit ones that already exist, and add or remove paragraphs, tables, images, and other document components. Numerous formatting options, including font styles, colours, alignment, and page layout, are supported by the library. We can also use it to extract text, images, and other data from Word documents for use in other applications or in-depth research. When automated document generation or manipulation is required, such as when producing reports, making templates, or processing lots of documents, docx is frequently used.

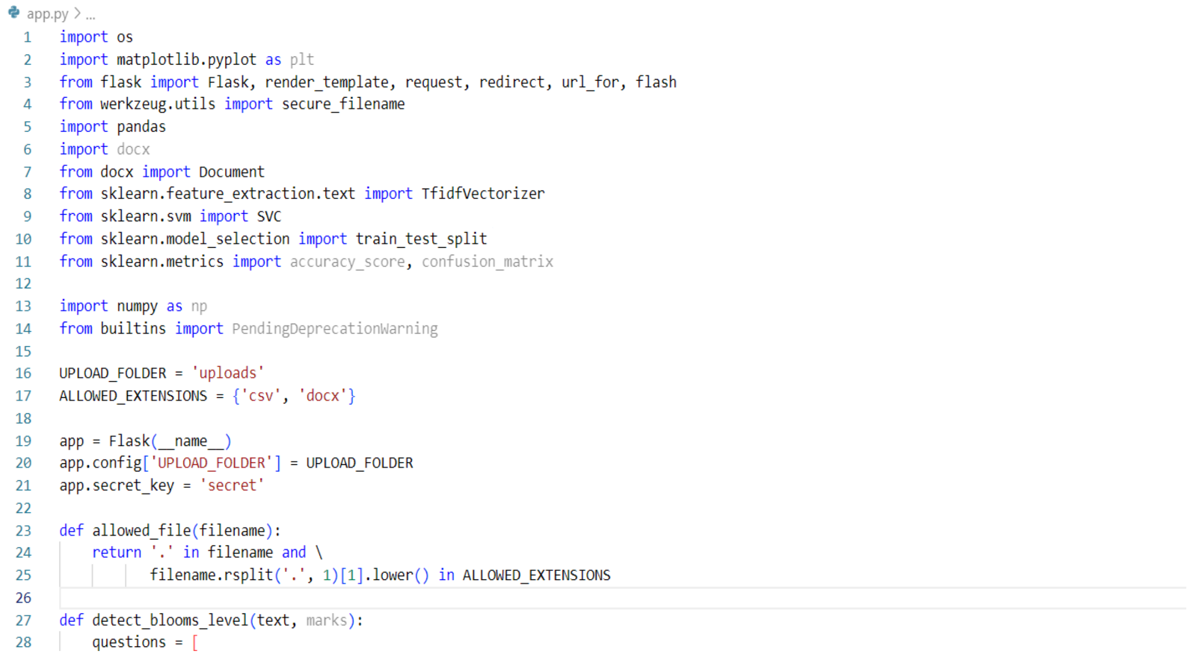
Sklearn:

Popular machine learning library scikit-learn, sometimes referred to as sklearn, is written in Python. It provides a complete collection of tools and techniques for a range of machine learning problems, including classification, regression, clustering, dimensionality reduction, and model selection . Sklearn provides a unified and consistent API for machine learning tasks and is built on top of other numerical and scientific Python libraries like NumPy and SciPy. Implementations of well-known algorithms like Support Vector Machines (SVM), Random Forests, K-Means clustering, and others are available in the library. Additionally, it offers tools for feature extraction, model performance evaluation, and data pre-processing. Sklearn features thorough documentation, logically organised code examples, and is user-friendly and accessible. It is widely used for machine learning in academia and industry.

6.2 Dataset

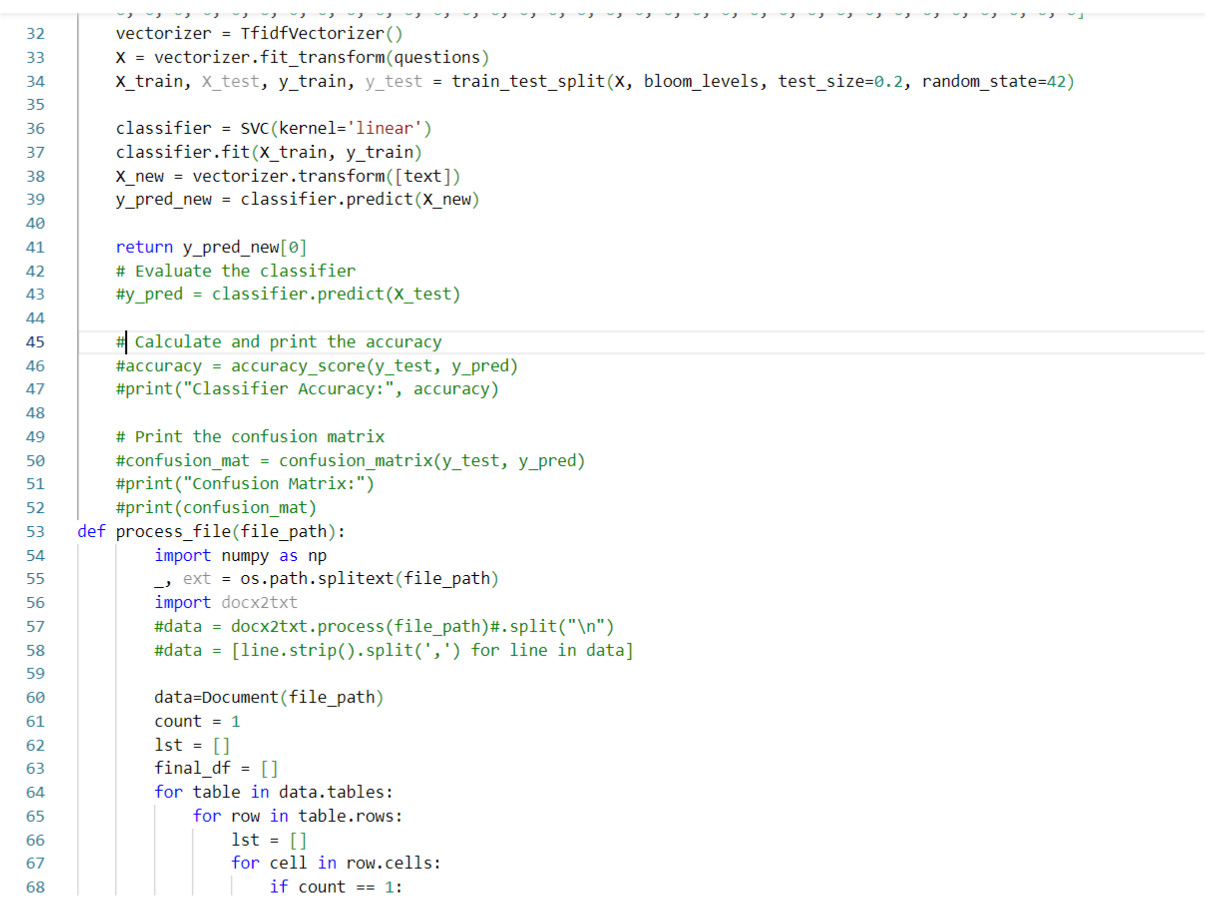
- https://www.researchgate.net/publication/303608228_Bloom's_Taxonomy_Cognitive_Levels_Data_Set.
- We also collected the data manually from the question papers of the end semester exams which we wrote.
- The data set consists of 700 questions ,which has the bloom’s levels mapped.

6.3 Sample code

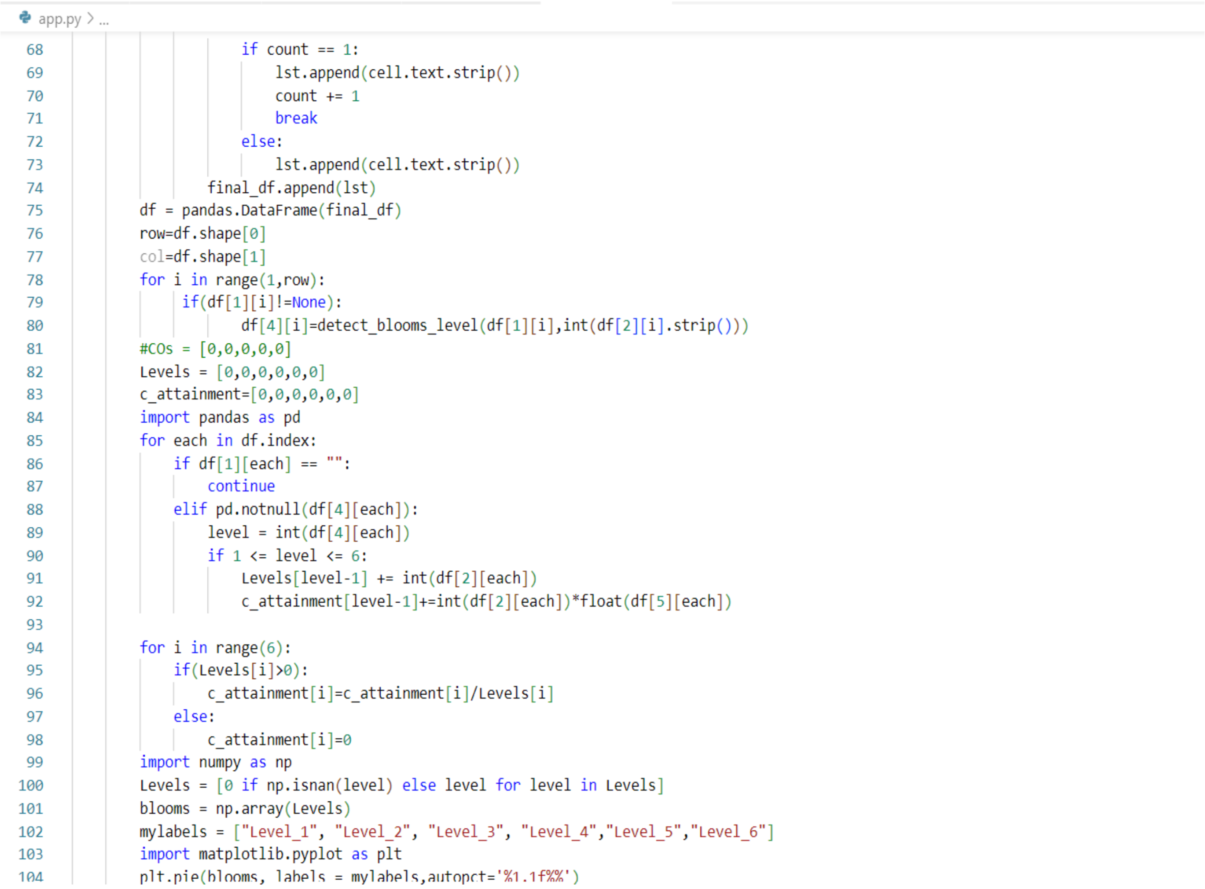


app.py > ...

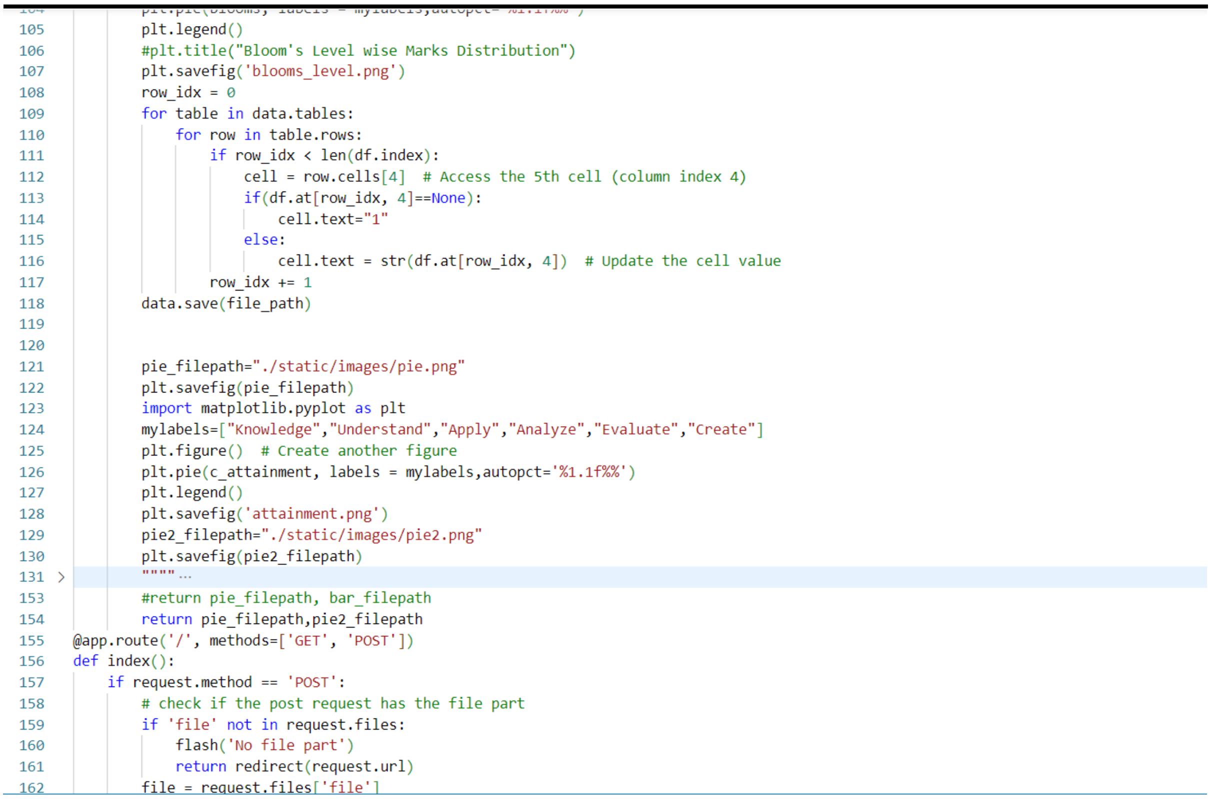
```python
import os
import matplotlib.pyplot as plt
from flask import Flask, render_template, request, redirect, url_for, flash
from werkzeug.utils import secure_filename
import pandas
import docx
from docx import Document
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix

import numpy as np
from builtins import PendingDeprecationWarning

UPLOAD_FOLDER = 'uploads'
ALLOWED_EXTENSIONS = {'csv', 'docx'}

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.secret_key = 'secret'

def allowed_file(filename):
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

def detect_blooms_level(text, marks):
    questions = [
```



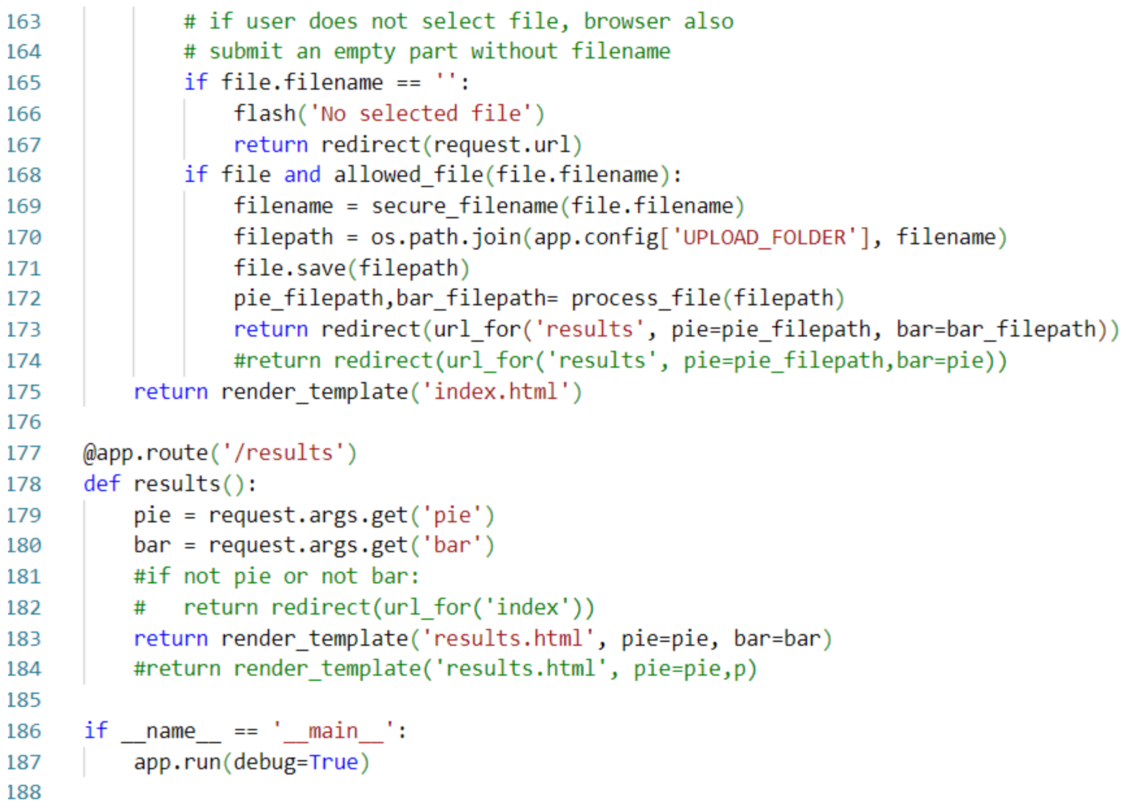
```python
    vectorizer = TfidfVectorizer()
    X = vectorizer.fit_transform(questions)
    X_train, X_test, y_train, y_test = train_test_split(X, bloom_levels, test_size=0.2, random_state=42)

    classifier = SVC(kernel='linear')
    classifier.fit(X_train, y_train)
    X_new = vectorizer.transform([text])
    y_pred_new = classifier.predict(X_new)

    return y_pred_new[0]
    # Evaluate the classifier
    #y_pred = classifier.predict(X_test)

    # Calculate and print the accuracy
    #accuracy = accuracy_score(y_test, y_pred)
    #print("Classifier Accuracy:", accuracy)

    # Print the confusion matrix
    #confusion_mat = confusion_matrix(y_test, y_pred)
    #print("Confusion Matrix:")
    #print(confusion_mat)
def process_file(file_path):
        import numpy as np
        _, ext = os.path.splitext(file_path)
        import docx2txt
        #data = docx2txt.process(file_path)#.split("\n")
        #data = [line.strip().split(',') for line in data]

        data=Document(file_path)
        count = 1
        lst = []
        final_df = []
        for table in data.tables:
            for row in table.rows:
                lst = []
                for cell in row.cells:
                    if count == 1:
```



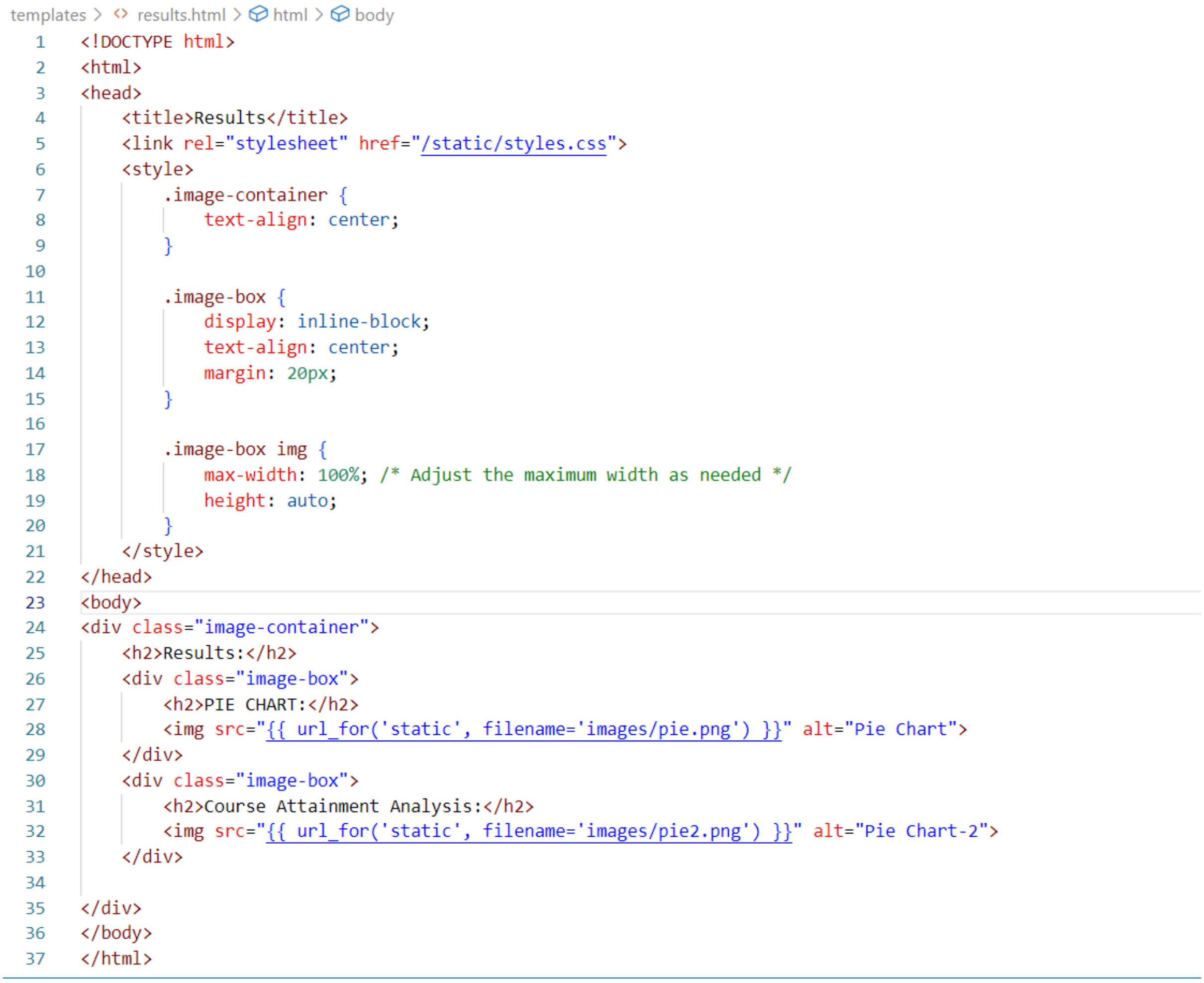
```
                    if count == 1:
                        lst.append(cell.text.strip())
                        count += 1
                        break
                    else:
                        lst.append(cell.text.strip())
                final_df.append(lst)
        df = pandas.DataFrame(final_df)
        row=df.shape[0]
        col=df.shape[1]
        for i in range(1,row):
            if(df[1][i]!=None):
                    df[4][i]=detect_blooms_level(df[1][i],int(df[2][i].strip()))
        #COs = [0,0,0,0,0]
        Levels = [0,0,0,0,0,0]
        c_attainment=[0,0,0,0,0,0]
        import pandas as pd
        for each in df.index:
            if df[1][each] == "":
                continue
            elif pd.notnull(df[4][each]):
                level = int(df[4][each])
                if 1 <= level <= 6:
                    Levels[level-1] += int(df[2][each])
                    c_attainment[level-1]+=int(df[2][each])*float(df[5][each])

        for i in range(6):
            if(Levels[i]>0):
                c_attainment[i]=c_attainment[i]/Levels[i]
            else:
                c_attainment[i]=0
        import numpy as np
        Levels = [0 if np.isnan(level) else level for level in Levels]
        blooms = np.array(Levels)
        mylabels = ["Level_1", "Level_2", "Level_3", "Level_4","Level_5","Level_6"]
        import matplotlib.pyplot as plt
        plt.pie(blooms, labels = mylabels,autopct='%1.1f%%')
```



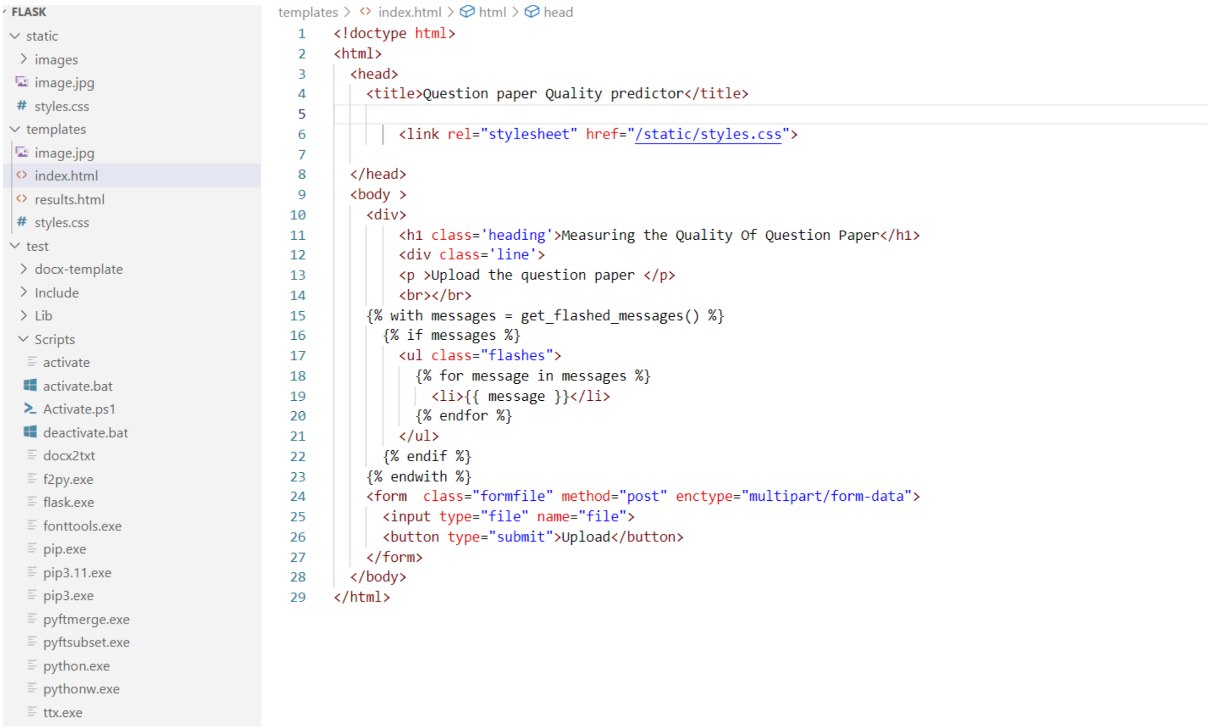
```
        plt.legend()
        #plt.title("Bloom's Level wise Marks Distribution")
        plt.savefig('blooms_level.png')
        row_idx = 0
        for table in data.tables:
            for row in table.rows:
                if row_idx < len(df.index):
                    cell = row.cells[4]  # Access the 5th cell (column index 4)
                    if(df.at[row_idx, 4]==None):
                        cell.text="1"
                    else:
                        cell.text = str(df.at[row_idx, 4])  # Update the cell value
                row_idx += 1
        data.save(file_path)


        pie_filepath="./static/images/pie.png"
        plt.savefig(pie_filepath)
        import matplotlib.pyplot as plt
        mylabels=["Knowledge","Understand","Apply","Analyze","Evaluate","Create"]
        plt.figure()  # Create another figure
        plt.pie(c_attainment, labels = mylabels,autopct='%1.1f%%')
        plt.legend()
        plt.savefig('attainment.png')
        pie2_filepath="./static/images/pie2.png"
        plt.savefig(pie2_filepath)
        """"...
        #return pie_filepath, bar_filepath
        return pie_filepath,pie2_filepath
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']
```



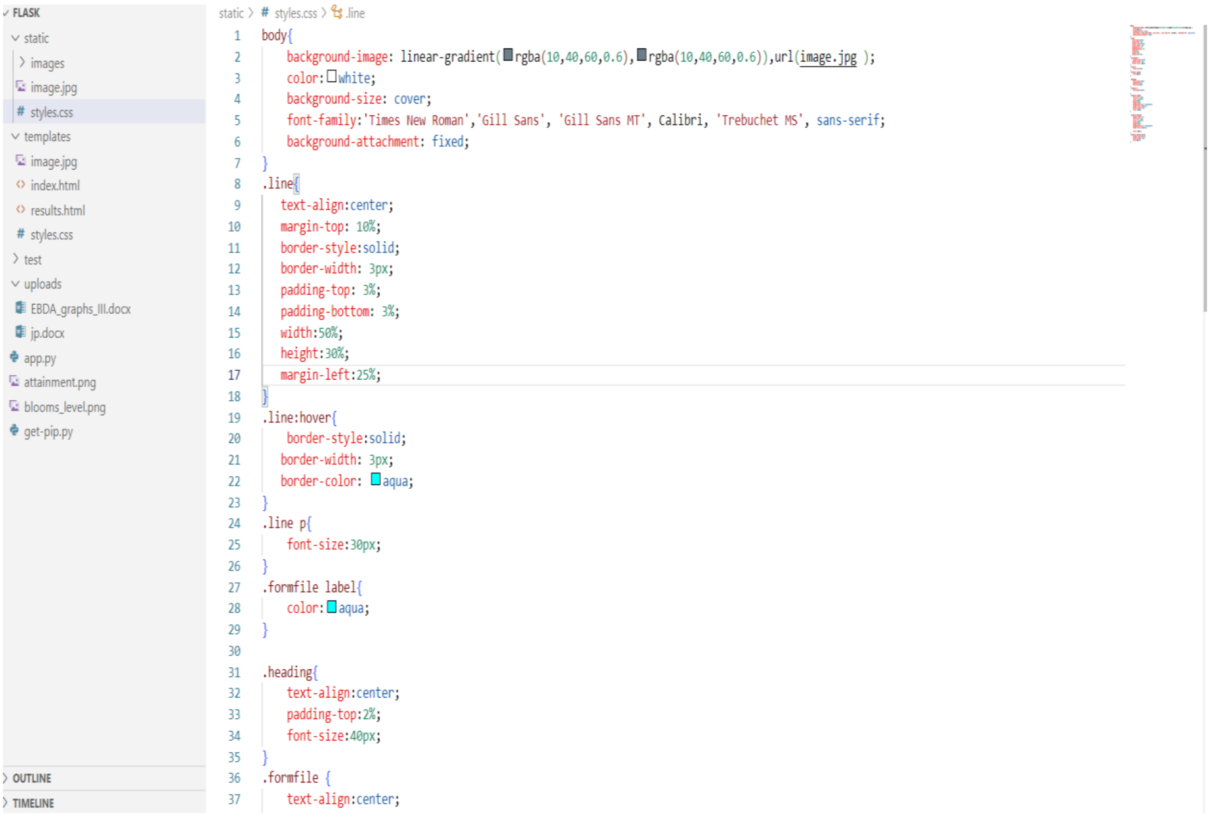
```
        # if user does not select file, browser also
        # submit an empty part without filename
        if file.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
            file.save(filepath)
            pie_filepath,bar_filepath= process_file(filepath)
            return redirect(url_for('results', pie=pie_filepath, bar=bar_filepath))
            #return redirect(url_for('results', pie=pie_filepath,bar=pie))
    return render_template('index.html')

@app.route('/results')
def results():
    pie = request.args.get('pie')
    bar = request.args.get('bar')
    #if not pie or not bar:
    #    return redirect(url_for('index'))
    return render_template('results.html', pie=pie, bar=bar)
    #return render_template('results.html', pie=pie,p)

if __name__ == '__main__':
    app.run(debug=True)

```



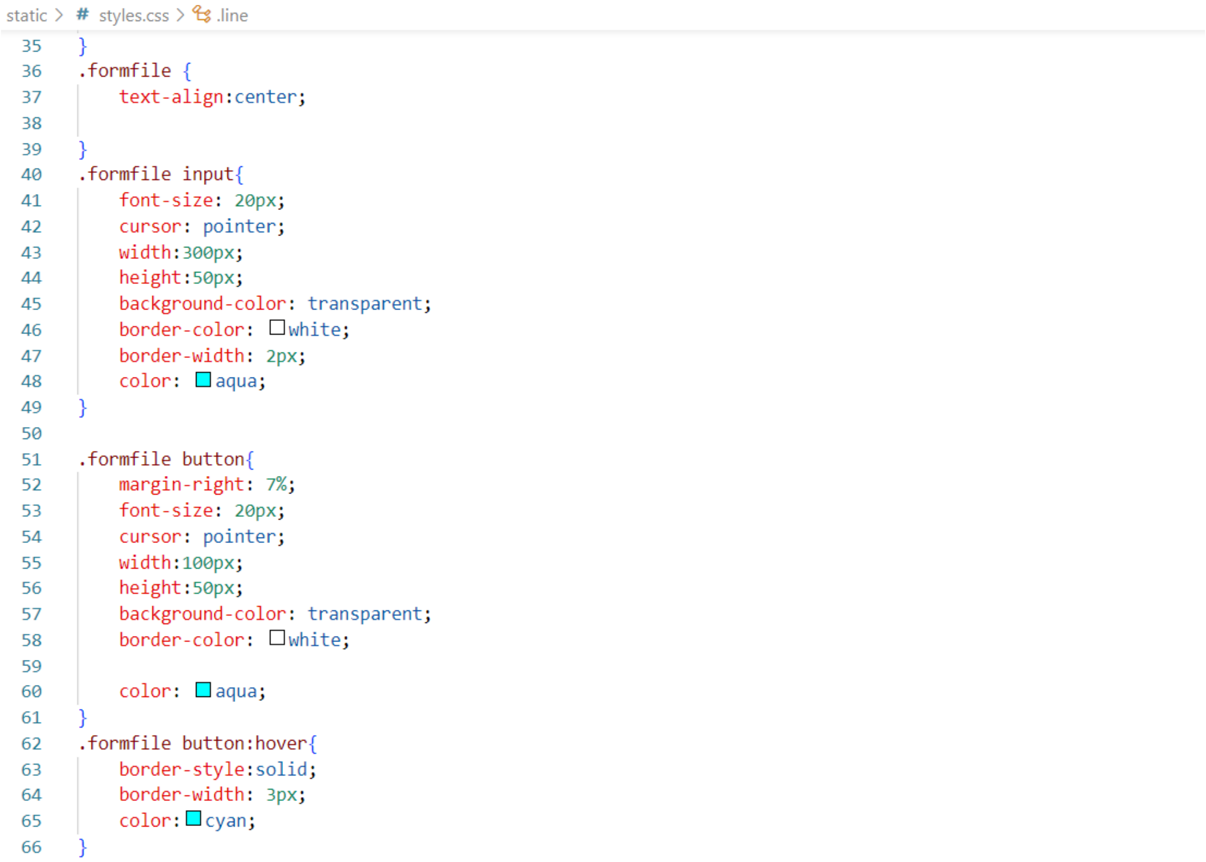
```
<!DOCTYPE html>
<html>
<head>
    <title>Results</title>
    <link rel="stylesheet" href="/static/styles.css">
    <style>
        .image-container {
            text-align: center;
        }

        .image-box {
            display: inline-block;
            text-align: center;
            margin: 20px;
        }

        .image-box img {
            max-width: 100%; /* Adjust the maximum width as needed */
            height: auto;
        }
    </style>
</head>
<body>
<div class="image-container">
    <h2>Results:</h2>
    <div class="image-box">
        <h2>PIE CHART:</h2>
        <img src="{{ url_for('static', filename='images/pie.png') }}" alt="Pie Chart">
    </div>
    <div class="image-box">
        <h2>Course Attainment Analysis:</h2>
        <img src="{{ url_for('static', filename='images/pie2.png') }}" alt="Pie Chart-2">
    </div>

</div>
</body>
</html>
```



```
<!doctype html>
<html>
  <head>
    <title>Question paper Quality predictor</title>

        <link rel="stylesheet" href="/static/styles.css">

  </head>
  <body >
    <div>
        <h1 class='heading'>Measuring the Quality Of Question Paper</h1>
        <div class='line'>
        <p >Upload the question paper </p>
        <br></br>
    {% with messages = get_flashed_messages() %}
      {% if messages %}
        <ul class="flashes">
          {% for message in messages %}
            <li>{{ message }}</li>
          {% endfor %}
        </ul>
      {% endif %}
    {% endwith %}
    <form  class="formfile" method="post" enctype="multipart/form-data">
      <input type="file" name="file">
      <button type="submit">Upload</button>
    </form>
  </body>
</html>
```



```css
body{
    background-image: linear-gradient(rgba(10,40,60,0.6),rgba(10,40,60,0.6)),url(image.jpg );
    color:white;
    background-size: cover;
    font-family:'Times New Roman','Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
    background-attachment: fixed;
}
.line{
   text-align:center;
   margin-top: 10%;
   border-style:solid;
   border-width: 3px;
   padding-top: 3%;
   padding-bottom: 3%;
   width:50%;
   height:30%;
   margin-left:25%;
}
.line:hover{
    border-style:solid;
   border-width: 3px;
   border-color: aqua;
}
.line p{
    font-size:30px;
}
.formfile label{
    color:aqua;
}

.heading{
    text-align:center;
    padding-top:2%;
    font-size:40px;
}
.formfile {
    text-align:center;
```



```css
}
.formfile {
    text-align:center;

}
.formfile input{
    font-size: 20px;
    cursor: pointer;
    width:300px;
    height:50px;
    background-color: transparent;
    border-color: white;
    border-width: 2px;
    color: aqua;
}

.formfile button{
    margin-right: 7%;
    font-size: 20px;
    cursor: pointer;
    width:100px;
    height:50px;
    background-color: transparent;
    border-color: white;

    color: aqua;
}
.formfile button:hover{
    border-style:solid;
    border-width: 3px;
    color:cyan;
}
```

7.SOFTWARE TESTING

What is software testing:

Software testing is a procedure that determines whether a piece of software is right by taking into account all of its characteristics and assessing how well its various parts operate in order to detect any bugs, mistakes, or flaws.

Software testing offers an impartial, unbiased perspective on the programme and ensures that it is fit for use. It entails testing each component that makes up the necessary services to determine whether or not they meet the requirements that have been set down. The procedure additionally gives the client knowledge about the calibre of the software.

Testing is required because failure of the programme at any point owing to a lack of testing would be dangerous. Software cannot be released to the end user without being tested.

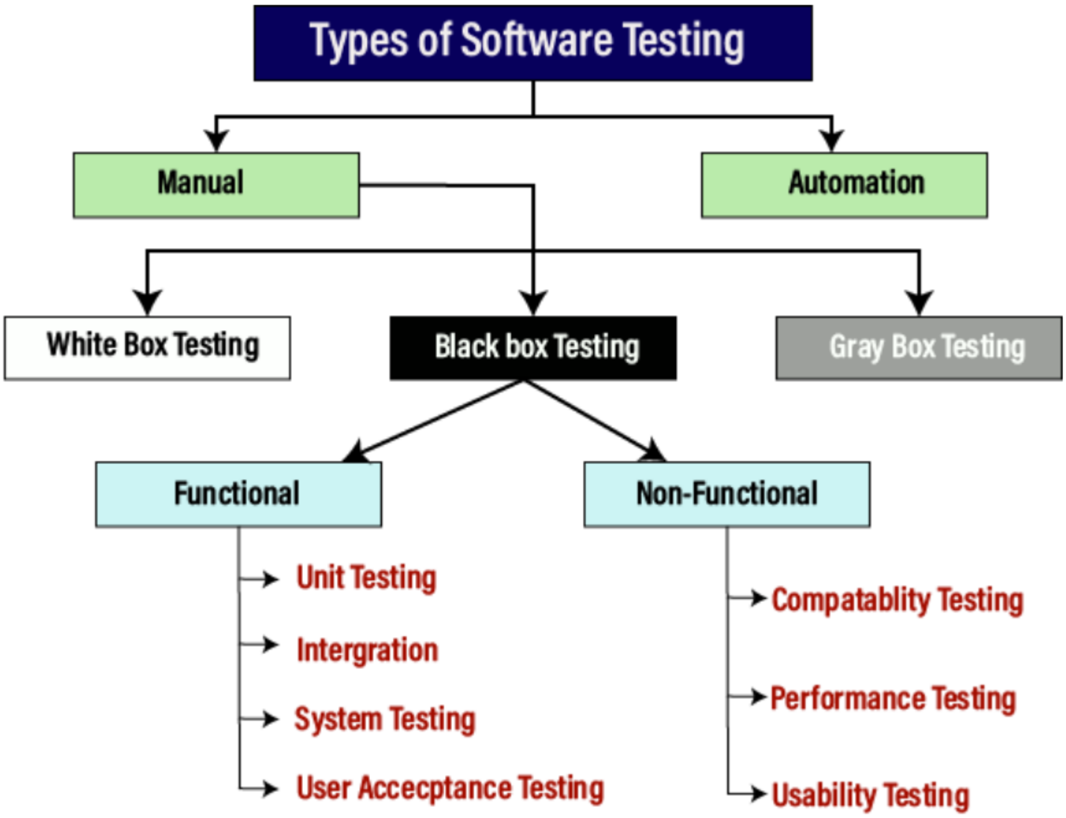


Figure 10: Types of Software Testing

<u>Manual testing:</u>

Without the use of any automation tools, the process of manually testing an application's functionality in accordance with user needs is carried out. It is not necessary to have in-depth knowledge of any testing tool to perform manual testing on any application; rather, in order to quickly create the test document, we should have a thorough understanding of the product.

Three different types of manual testing exist, and they are as follows

- **White box testing**
- **Black box testing**
- **Gray box testing**

7.1 White box testing:

A software testing technique called white box testing, also referred to as clear box testing, structural testing, or glass box testing, looks at the internal organisation and implementation specifics of a software application. The source code, architecture, and design documents of the system under test are all accessible to the tester during white box testing.

The main objective of white box testing is to evaluate the internal logic, control flow, and data flow of the software application. This kind of testing is concerned with confirming the accuracy of each component, module, or unit as well as the way in which they are integrated into the overall system. It looks for bugs or errors in the code, makes sure all possible paths and conditions have been tested, and checks that the software performs as expected.

Early defect detection, better code quality, a better understanding of the internal structure, and increased trust in the behaviour of the software are all benefits of white box testing. However, especially for larger applications, it can be time-consuming and requires a thorough understanding of the codebase.

7.2 Black box testing:

A software testing technique called "black box testing" focuses on evaluating a software application's functionality without having any knowledge of its underlying structure or details of its implementation. When testing software using a "black box", the tester only has access to the inputs, anticipated outputs, and behaviour of the system.

Black box testing's main goal is to confirm whether the software satisfies the requirements and works properly from the end-user's perspective. It looks for mistakes, flaws, or differences between the software's expected behaviour and actual behaviour.

Black box testing has benefits such as its independence from the internal architecture, which makes it appropriate for testing applications created by various teams or outside vendors. Because it doesn't require programming expertise, non-technical testers can conduct the testing. Black box testing also aids in finding problems that programmers might overlook.

Black box testing does, however, have its limitations. Due to its reliance on predefined inputs and anticipated outcomes, it may not offer comprehensive coverage of all potential scenarios. Additionally, there is a chance of missing specific code paths or internal flaws that can only be found using white box testing or other methods.

Black box testing can be further divided into two types of testing, which are as follows:

- **Functional testing**
- **Non-functional testing**

Functional testing:

To ensure that every software application function operates according to the requirements as stated in the requirement document. Testing all of the functionalities by supplying the proper input will allow you to determine whether the actual output matches the anticipated output or not. The testers don't need to worry about the application's source code because it falls under the purview of black-box testing.

The following are the two categories into which black box testing can be further divided:

- **Unit testing**
- **Integration testing**
- **System testing**
- **User acceptance testing**

Unit testing:

To ensure that the individual source code modules are functioning properly, unit testing is done. By testing each component of the application separately in the developer's environment. It is also known as module testing or component testing.

Integration testing:

Integration testing is the process of evaluating the connectivity or data transfer between a few modules that have undergone unit testing. It is also referred to as I&T testing or string testing. Top-Down, Bottom-Up, and Sandwich (a combination of Top-Down and Bottom-Up) are the three categories of approaches.

Automation testing:

By using automation tools or any programming language, automation testing is the process of turning any manual test cases into test scripts. The use of automation testing allows us to run tests more quickly because no human labour is required. Test scripts must be written and executed.

Non-Functional testing:

Non-functional software testing is focused on evaluating a software system's features and attributes outside of its functional needs. It entails testing non-functional elements that affect the software's overall quality, including performance, reliability, usability, security, scalability, and others.

Non-functional testing is essential for making sure that the software not only satisfies functional requirements but also offers end users a high-quality experience in terms of performance, security, usability, and other crucial areas. It enables the necessary improvements or adjustments to be made before deployment by assisting in the identification of potential problems or risks that may affect the software's usability, dependability, or security.

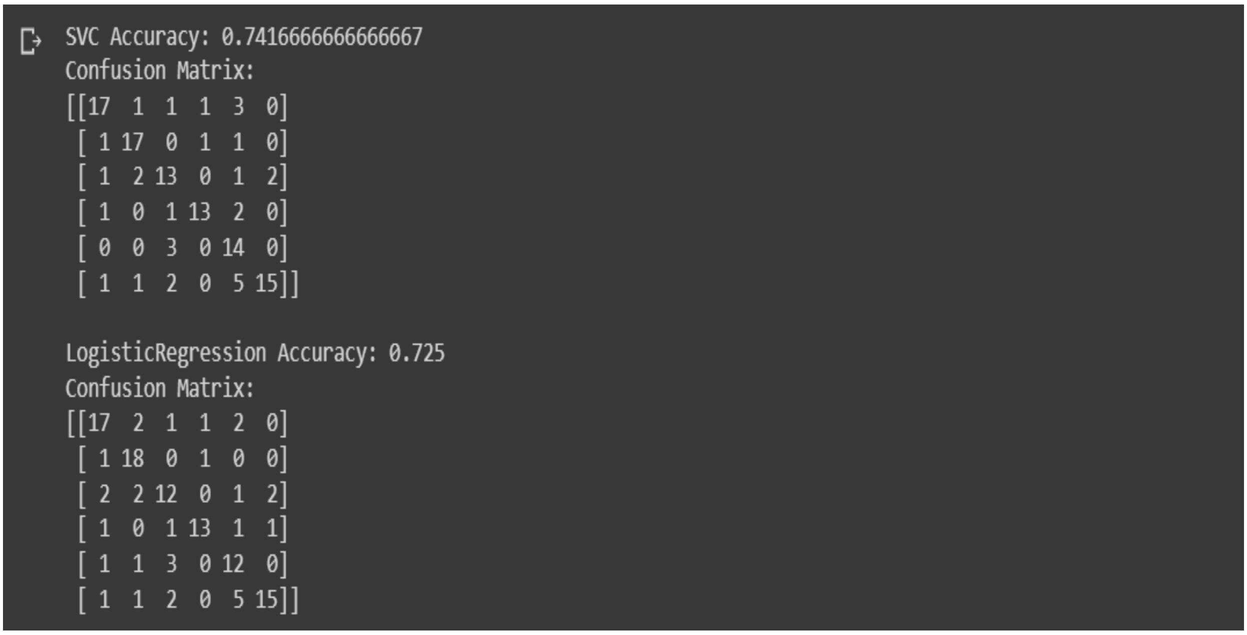
7.3 Gray box testing:

Gray box testing is a software testing technique that combines elements of black box testing and white box testing. The tester's understanding of the internal workings and implementation specifics of the software application under test is limited in gray box testing.

Access to some incomplete system information, such as high-level design documents, database schemas, or a limited understanding of the code, is typically required for gray box testing. With the aid of this knowledge, the tester is able to comprehend the inner workings of the software to a greater extent than a black box tester but not as thoroughly as a white box tester.

Gray box testing's goal is to strike a balance between white box testing's depth of testing and black box testing's external perspective. It aims to test the software's features, inputs, and outputs while also utilising the internal knowledge to create more focused and effective test cases.

TESTING FOR THIS PROJECT:



```
SVC Accuracy: 0.7416666666666667
Confusion Matrix:
[[17  1  1  1  3  0]
 [ 1 17  0  1  1  0]
 [ 1  2 13  0  1  2]
 [ 1  0  1 13  2  0]
 [ 0  0  3  0 14  0]
 [ 1  1  2  0  5 15]]

LogisticRegression Accuracy: 0.725
Confusion Matrix:
[[17  2  1  1  2  0]
 [ 1 18  0  1  0  0]
 [ 2  2 12  0  1  2]
 [ 1  0  1 13  1  1]
 [ 1  1  3  0 12  0]
 [ 1  1  2  0  5 15]]
```

Figure 11 : Testing

The accuracy we achieved using the SUPPORT VECTOR CLASSIFIER IS 74.166% and using the Logistic Regression model is 72.5%.

8. RESULTS

8.1 Interface

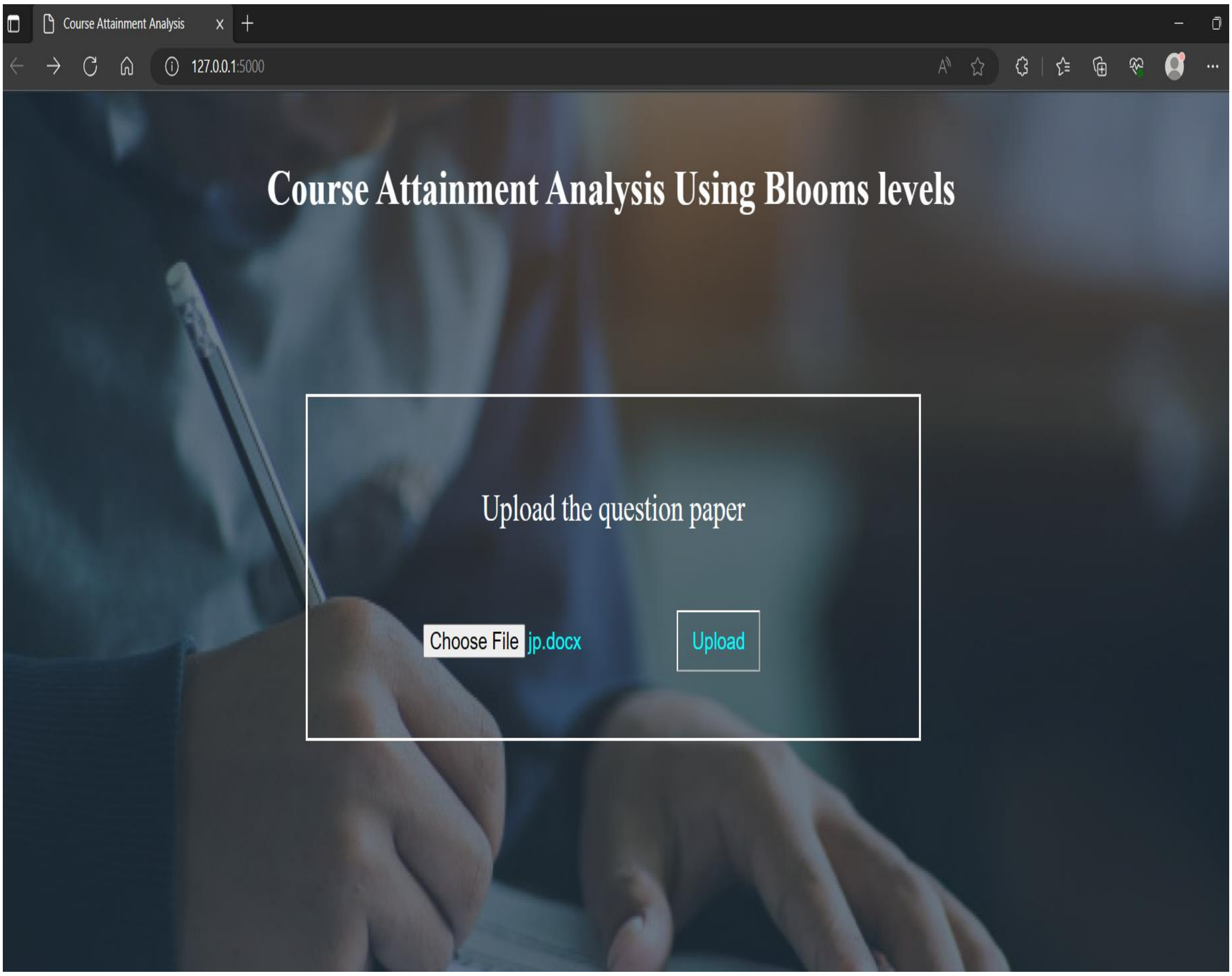
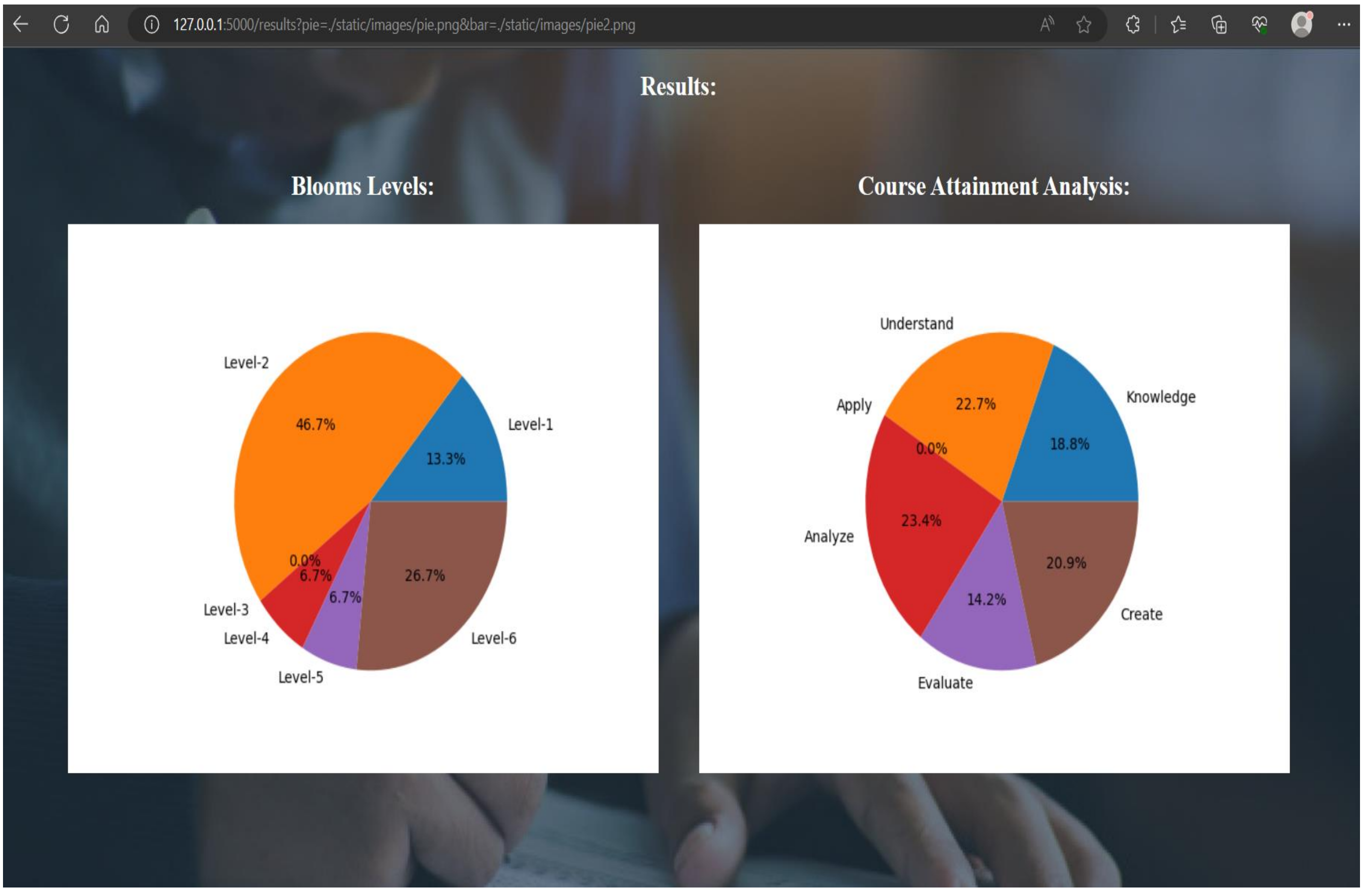

Course Attainment Analysis
127.0.0.1:5000
Course Attainment Analysis Using Blooms levels
Upload the question paper
Choose File
jp.docx
Upload

Figure 12 : User Interface



127.0.0.1:5000/results?pie=./static/images/pie.png&bar=./static/images/pie2.png

Results:

Blooms Levels:

Course Attainment Analysis:

Figure 13 : Pie-Chart

Our project's findings are as follows:

Classification of Bloom's Levels:

Using machine learning algorithms, particularly Support Vector Classification (SVC) and Logistic Regression, we were able to successfully classify the Bloom's levels of each question in the exam papers. With SVC, the accuracy is 74.1667%, and with Logistic Regression, it is 72.5%. The cognitive complexity of the questions can be better understood using this.

Pie Chart Visualisation:

We created a pie chart to show the distribution of Bloom's levels across the exam questions. This chart gives a clear summary of the percentage of questions falling into each Bloom's category, making it easy to determine the exam's overall cognitive demands.

Weighted Average Percentage of Course Attainment:

For each Bloom's level, we calculated the weighted average percentage of course attainment. This metric offers information about the students general performance and comprehension across various cognitive domains. We were able to obtain an extensive measure of student achievement by giving each Bloom's level the proper weights.

Visualisation of Course Attainment:

We represented the weighted average percentage of each Bloom's level's course attainment. With the aid of this visualisation, it is simple to understand and evaluate student performance at various cognitive levels. Educators and stakeholders can quickly identify areas of strength and areas that may need more attention by presenting the data in a visually appealing format.

Updated Question Paper:

As part of the results, we added the Bloom's levels of each question to the question paper file. This improvement offers useful data for upcoming analysis, curriculum development, and lesson planning.

Overall, the outcomes of our project show how well machine learning algorithms can categorise Bloom's levels, and the visualisations provide insightful information about how cognitive complexity and student achievement are distributed. The revised question paper with annotations for Bloom's levels makes it easier to conduct future research and develop educational materials.

9.CONCLUSION AND FUTURE ENHANCEMENTS

9.1 Conclusion

In this study, this work categorizes educational goals into various cognitive levels, A well-known taxonomy is Bloom's and significant framework. It was created by Benjamin Bloom and his associates in the 1950s and has been used as a manual for educators to plan lessons and evaluate the learning results of their pupils. Bloom's Taxonomy offers a thorough framework for comprehending the complexity of cognitive processes involved in education by classifying learning objectives into six hierarchical levels.

Using Support Vector Machine(SVM) and Logistic Regression algorithms, this project aimed to automate the classification of Bloom's taxonomy levels for questions in exam papers. SVC had an accuracy of 74.1667%, whereas Logistic Regression had an accuracy of 72.5%.

A pie chart is created to show the distribution of question classifications across various Bloom's levels in order to analyse the results. The project also computed and graphically displayed the course attainment weighted average percentage for each Bloom's level. According to the results, using SVC to classify data performed better than using the standard keyword mapping technique to determine Bloom's taxonomy levels. This suggests that classification of questions according to their cognitive complexity can be accomplished by machine learning algorithms like SVC.

Additionally, this work displays the course completion rates for each Bloom's level. This analysis sheds light on the students abilities and comprehension at various cognitive levels. This work evaluates the overall proficiency of students in each category by calculating the course attainment weighted average percentage for each Bloom's level. Then, in order to facilitate comparison and interpretation, this information is represented using the pie chart. Students learning outcomes and their command of various cognitive skills are comprehensively viewed through the visualisation of course attainment for each Bloom's level. It can assist teachers and curriculum developers in determining their strengths and potential areas for improvement.

Overall, this project shows the potential of automated systems and machine learning methods for precisely determining the cognitive levels of questions in educational assessments. By automating the classification process, educators and trainers can improve educational practises while saving time and effort when evaluating test questions.

9.2 Future Enhancements

User Feedback and Refinement: Include a mechanism for users to provide feedback on the anticipated Bloom's levels, such as teachers or students.The performance of the model can be gradually improved and refined using this feedback.

Integration with Learning Analytics: Integrate the system with well-known LMS platforms used in educational institutions. Learning Management System (LMS). Within the framework of the current educational workflows, this integration can enable scam less question paper analysis, course attainment assessment, and visualisation.

Automated Question Generation: Expand the system's ability to generate questions automatically.This would make it possible to generate fresh questions with predetermined Bloom's levels, giving teachers the ability to design unique question papers that are in line with particu lar learning objectives.

Adaptive Assessments: Examine the potential for adaptive assessments, where the system varies the level of difficulty or Bloom's classification of the questions based on the performance and academic development of each individual student. This individualised strategy can deliver custom assessments that successfully challenge and engage students.

10.BIBILOGRAPHY

[1] Manisha Dubey : Classifying Stack Overflow Questions Based on Bloom’s Taxonomy.

[2] Dr. G. N. R. Prasad: Identification of Bloom’s Taxonomy level for the given Question paper using NLP Tokenization technique.

[3]Sabine Ullrich Michaela Geierhos:Using Bloom’s Taxonomy to Classify Question Complexity.

[4] Kristopher J. Kopp1, Amy M. Johnson1, Scott A. Crossley2 and Danielle S. McNamara1: Assessing Question Quality using NLP.

[5] McNamara, D. S.: Self-explanation and reading strategy training (SERT) improves low-knowledge students’ science course performance. Discourse Processes (2015).

[6] McNamara, D. S., O'Reilly, T. P., Best, R. M., & Ozuru, Y.: Improving adolescent learners' reading comprehension with iSTART. J. of Educ. Comp. Res., 34(2), 147-171(2006).

[7] Cerdán, R., Vidal-Abarca, E., Martínez, T., Gilabert, R., & Gil, L.: Impact of question-answering tasks on search processes and reading comprehension. Learning and Instruction, 19(1), 13-27 (2009).

[8] Graesser, A. C., Jackson, G. T., Mathews, E. C., Mitchell, H. H., Olney, A., Ventura, M& Person, N.K : Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialog. In Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society (pp. 1-6) (2003)

[9]. Bullington, J. Endres, I. and Rahman, M(2007), Open-ended Question Classification Using Support Vector Machines, MAICS 2007.

[10] Bush, HM Daddysman, J and Charnigo, R.(2014) Improving Outcomes with Bloom‘s Taxonomy: From Statistics Education to Research Partnerships, Journal of Biometrics & Biostatistics, Volume 5, Issue 4.

[11] Cutrone L & Chang, M. (2010). Automarking: Automatic Assessment of Open Questions, 2010 10th IEEE International Conference on Advance Learning Technologies. 143-147.

[12] Johnson C.G. & U. Fuller. (2006). Is Bloom's Taxonomy is Appropriate for Computer Science, Koli Calling 2006, Koli National Park, Finland. 115-118.

[13] Jones K. O Harland, J Reid, J. M.V., Barlett, R (2009) . Relationship Between Examination Questions and Bloom's Taxonomy, 39th.

[14] McNamara, D. S., Allen, L. K., Crossley, S. A., Dascalu, M., & Perret, C. A. (in press). Natural language processing and learning analytics. In G. Siemens & C. Lang (Eds.), Handbook of Learning Analytics and Educational Data Mining.

[15] Bloom, Benjamin Samuel. "Taxonomy of educational objectives: The classification of educational goals (1956).

C1

ORIGINALITY REPORT

23%	15%	8%	18%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Educational Service District 105 Student Paper	2%
2	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	2%
3	files.eric.ed.gov Internet Source	1%
4	www.coursehero.com Internet Source	1%
5	Submitted to Coventry University Student Paper	1%
6	Submitted to University of Leicester Student Paper	1%
7	Submitted to Kingston University Student Paper	1%
8	Submitted to University of Wales Institute, Cardiff Student Paper	1%
9	www.ijke.org	