ABSTRACT

The Career Recommendation Website is an innovative platform designed to assist students and young professionals in making informed career choices by leveraging modern web technologies and machine learning. Unlike traditional career guidance systems that rely on outdated methods like Myers-Briggs or Holland Codes, which primarily focus on personality traits, this project emphasizes a user's skills and interests to recommend suitable career paths. Users interact with the system by inputting scores across eight key parameters, reflecting their proficiency in various skill areas relevant to today's job market. These inputs are processed by a machine learning model trained on the RandomForest algorithm, which analyses the data to predict and suggest the most appropriate career options tailored to the individual's strengths. Additionally, the platform integrates a dynamic dashboard built with PowerBI, providing users with a visually engaging representation of their skill profiles and career recommendations, making the decision-making process both intuitive and data-driven.

This project aims to overcome the shortcomings of current career guidance tools, which can be inefficient and often fail to connect with the real-world requirements of today's industries. By emphasizing skill-based assessments, the system delivers a more relevant and tailored experience, enabling users to align their career goals with their actual skills. The platform's web-based technology ensures that it is accessible from any device, while the machine learning model enhances its accuracy as it analyses more data. The Career Recommendation Website is especially beneficial for students who may feel daunted by the multitude of career choices or are uncertain about their professional direction. By offering clear, evidence-driven recommendations, this system not only saves time but also boosts users' confidence as they pursue a rewarding career that suits their individual skills and interests.

Keywords: – Machine Learning, RandomForest, Recommendation System, PowerBI, and Data Visualization.

Table of Contents

1. Introduction

1.1 Overview

The Career Recommendation Website is designed to assist students and young individuals in discovering their career paths in a highly personalized and user-friendly manner. Unlike traditional methods that rely on personality type assessments, this platform emphasizes your actual skills and interests. You provide scores for eight different abilities, and our system, utilizing a RandomForest algorithm, analyzes this data to recommend careers that suit you perfectly. Additionally, there's an engaging PowerBI dashboard that presents your results in a visual and interactive format, allowing you to clearly identify your strengths. The primary aim is to simplify career planning, making it more focused on your abilities, your interests.

1.2 Feasibility Study

Let's discuss whether this project is feasible or not? We evaluated several important factors to ensure the website is feasible. Technical feasibility: By utilizing reliable web technologies such as HTML, and CSS for the front end, along with Python for the RandomForest model. PowerBI is an excellent choice for the dashboard, and all these tools have strong support. Both students and career counselors will find it user-friendly, as not much technical knowledge is required. We also analyzed the market, and given that many students feel uncertain about their career paths, there is a significant demand for a solution like this. Additionally, the job market is constantly evolving, so a tool that adapts to these changes is definitely beneficial.

1.3 Existing System

Traditional approaches, like Myers-Briggs or Holland Codes, mostly focus on analyzing your personality, rather than the skills. That's interesting, but not the best for picking a career. These methods often drag on with lengthy questionnaires and overlook your actual skills. There are some online tools out there, but they tend to be either too simplistic like those websites which are just plain to use. Most of them don't tap into smart tech like machine learning for precise recommendations. In short, what's available now feels outdated, lacks personalization, and doesn't do a great job of linking your strengths to real-world career paths.

1.4 Proposed System

We've created a platform that's all about your skills and passions, not just what kind of personality you have. You'll enter scores for eight key areas, like problem-solving, tech, or soft skills and our RandomForest algorithm will process the numbers to suggest the career that fit you perfectly. It's fast, accurate, and stays in sync with what's trending in the job world. The PowerBI dashboard is a total highlight it gives you a clean, visual snapshot of your results, making it easy to see which careers match your strengths. Designed to work smoothly on your phone or laptop, this platform is user-friendly and modern. Unlike those old-school systems, ours is quick, data-driven, and focused on guiding you to a career, based on what makes you unique.

2. Literature Survey

A system showcased at the 2022 International Conference on Computational Collective Intelligence (ICCCI) focused on connecting underserved students in developing regions with industry mentors, using neural networks and LightFM. It nailed a 91% accuracy rate, which is impressive, but it also revealed a big issue: many students aren't even aware of the career paths available to them. This highlights the need for tools that consider diverse factors like personal values or location preferences, which our system tackles by incorporating an eight-parameter framework to capture a broader picture of each user. [1]

AI driven career tools are making waves, especially in developing countries where access to counselling is limited (Choi & Gupta, 2023). These systems use machine learning, fuzzy logic, collaborative filtering, and data mining to offer tailored advice based on personality, academic performance, and aptitudes. They're great at reducing biases like gender stereotypes and staying current with job market trends. But many lack a clear, engaging way to show results. Our PowerBI dashboard fixes that by presenting recommendations in a visual, easy-to-grasp format that users can actually connect with. [2]

Some systems target secondary school students to kick start early career planning. A 2021 project by Smith et al. used Decision Trees and K-Nearest Neighbors (KNN) with Python, Django, Scikit-learn, and PostgreSQL to build a career prediction website. It included skill assessments and result analysis but was limited to academic and skill-based inputs, missing out on personality or regional factors. Our platform goes wider, ensuring we account for more aspects of a user's profile. [3]

For engineering students, one system used Support Vector Machines (SVM), XGBoost, and Decision Trees to analyse academic records, technical skills, and psychometric data. It cut down biases compared to traditional counselling and delivered data-driven advice through a digital platform. However, it was super specific to engineering, which limits its reach. Our website, by contrast, is designed to work for users from all kinds of backgrounds. [4]

For fresh graduates, a job recommendation system used content-based filtering and the SVD++ algorithm, hitting 91% precision with a 0.9737 RMSE (Patel & Lee, 2020). Built with web frameworks, it had plans for an Android app and better UI, focusing on social factors. But it was more about matching people to immediate jobs than guiding long-term career paths, which is where our system shines by focusing on broader career planning for students. [5]

Research shows a big gap in career awareness, with many students stuck relying on outdated or generic career advice. [6]

Studies also highlight that students often struggle to recognize their true potential, and even when they do, it's sometimes too late. Plus, recommendation systems for career advice are pretty rare. [7-9]

Using supervised learning techniques like KNN, Random Forest Classifier, and Naïve Bayes helps make predictions more accurate. [10]

One study emphasized that user inputs like body language or general appearance are key to predicting the right career, alongside other attributes, which aligns with our eight parameter approach. [11]

Without proper career counselling, students often feel lost, picking courses they don't care about and avoiding career decision discussions altogether. [12-13]

Research also points out that when parents ignore a student's passions, it can sap their motivation, and lack of interest means even external encouragement won't unlock their potential. Our Career Recommendation Website tackles these challenges by letting users input scores across eight key areas, processed by a RandomForest algorithm to deliver personalized, data-driven career suggestions. With a slick PowerBI dashboard, we make it easy and engaging for students to discover careers that match their skills and passions, helping them take confident steps toward a future they're excited about. [14-16]

3. Hardware and Software Requirements

3.1 Hardware Interfaces:

Section	Requirements
Processor	AMD Ryzen 5 or higher
RAM	8.00 GB
Monitor	Minimum resolution 800x600 pixels (PC), 640x1136 pixels(Mobile)
CPU	3.00GHZ
Hard Disk	256GB of minimum free space
Keyboard	Normal or Multimedia (PC)
Internet	Ethernet (LAN), OR Wireless adapter (Wi-Fi)

Table 1: Hardware Requirements: Which indicate the requirements used in the project

3.2 Software Interfaces:

Front-end	HTML, CSS
Back-end	Flask
Database	MS Excel
Operating System	Windows 11 (PC)
Applications	VS Code, PowerBI, Anaconda

Table 2: Software Requirements: Which indicates the requirements used in the project

4. Software Requirement Specification

4.1 Introduction

4.1.1 Purpose

This SRS lays out the requirements for the Career Recommendation Website, version 1.0, a platform built to guide students and young professionals toward careers that match their skills and passions. It covers the full system web interface, machine learning model, and PowerBI dashboard for visualizing results. The document serves as a roadmap for developers, testers, and stakeholders to create a tool that delivers personalized, data-driven career advice.

4.1.2 Document Conventions

This SRS follows a standard IEEE format with Times New Roman font. All the headings are of size 16pts, sub-headings are of size 14pts and the inner contents are of 12pts. The headings and sub-headings are written in bold whereas the content has normal font. This SRS is straightforward, divided up into sections detailing an overall description, project requirements, system features, and other non-functional requirements.

4.1.3 Intended Audience and Reading Suggestions

This SRS is for users, developers, project managers, testers, documentation writers, faculty advisors and evaluators. Users can start with Section 4.1 for a quick overview, then hit Section 4.2 for context. Developers and testers should dive into Sections 4.3, 4. 4, and 4.5. Managers should focus on 4.1.4 and 4.2.7.

4.1.4 Product Scope

The Career Recommendation Website helps students find careers that fit their strengths by letting them score eight key areas (like Logical-Mathematical or Linguistic). A RandomForest algorithm processes these inputs to suggest career, and a PowerBI dashboard displays results in a visual way. Unlike old-school tools like Myers-Briggs, which focus on personality, this system prioritizes skills and job market trends. It aims to make career planning faster, more relevant, and empowering, aligning with the goal of supporting informed career decisions.

4.1.5 References

- Power BI Public Dashboard: https://app.powerbi.com/
- Flask Documentation: https://flask.palletsprojects.com/
- IEEE 830-1998 – IEEE Recommended Practice for Software Requirements Specifications

4.2 Overall Description

4.2.1 Product Perspective

The Career Recommendation Website is a standalone, web-based tool designed to replace outdated career counselling methods. Though it's not part of a larger product family but integrates with PowerBI for dashboards and Scikit-learn for machine learning. The system takes user inputs, processes them via a RandomForest model, and outputs career recommendations. It interfaces with a web browser for user access and a dataset for storing careers.

4.2.2 Product Functions

- Let users input scores for eight skill areas (such as Logical-Mathematical, Linguistic, etc.).
- Process inputs using a RandomForest algorithm to predict careers.
- Display results via an interactive PowerBI dashboard.
- Provide a responsive web interface for access on any device.

4.2.3 User Classes and Characteristics

- **Students**: High school or college students can be the frequent users, need clear career guidance.
- **Career Counsellors:** Professionals advising students based on the result of our recommendation system, these can be occasional users, high tech skills, need detailed reports.

4.2.4 Operating Environment

The system runs on a web server with Python 3.8+, Scikit-learn, and Django framework. The front end uses HTML, and CSS. PowerBI Embedded handles dashboards on the website. It's compatible with browsers like Chrome, Firefox, and Safari on Windows, macOS, iOS, and Android. The system needs a stable internet connection for cloud hosting.

4.2.5 Design and Implementation Constraints

- Use HTML and CSS for user-friendly and interactive user interface.
- Use Python and Scikit-learn for the RandomForest model.
- Integrate PowerBI Embedded for dashboards.
- Use Django for web framework to ensure scalability.

4.2.6 User Documentation

The following documentation will be provided:

- User Manual (PDF) – Detailed guide on how to use the application, upload data, and interpret graphs.
- System Overview Presentation – High-level description of system components and data flow.

4.2.7 Assumption and Dependencies

- Users have basic internet and browser access.
- Job market data is accurate, and students can self-assess skills.
- Relies on Scikit-learn for ML, PowerBI for dashboards.
- Updates to these tools may impact performance.
- A separate vision document details job market data sources.

4.3 External Interface Requirements

4.3.1 User Interfaces

The Career Recommendation Website features a responsive and interactive graphical user interface (GUI). It provides the following components:

- **Home Page:** It includes the brief description of the project and navigation menu.
- **Prediction Page:** It includes the input fields with eight skills which the user needs to fill out based on their skill set.
- **Result Page:** It displays the recommended career, and also consists of the PowerBI dashboard where the user can navigate through other career options.

4.3.2 Hardware Interfaces

No direct hardware interactions, as the system is web-based. However, the typical user environment includes:

- **Client:** Desktop or laptop with:
  - Minimum: 4 GB RAM, 2-core CPU, 50 GB free storage
  - Recommended: 8 GB RAM, 4-core CPU
- **Display:** 1280x720 pixel resolution or higher
- **Server** (for deployment):
  - OS: Ubuntu 20.04 or Windows 10 and above
  - Python 3.8+, 4 GB RAM, 20 GB storage

4.3.3 Software Interfaces

- **Operating System:** Windows/Linux/macOS
- **Back-end framework:** Flask 2.3+
- **Front-end:** HTML and CSS
- **Scikit-learn**: Processes RandomForest predictions

- **PowerBI Embedded**: Renders dashboards, receiving JSON data from the ML model

4.3.4 Communication Interfaces

- PowerBI dashboards load via iframe with API authentication
- Machine learning model integration via Flask
- Web browsers like Chrome, Firefox, Edge, Brave

4.4 System Features

4.4.1 Skill Input Collection

Description: The system lets users (students or career counsellors) enter scores for eight skills, like Linguistic or Logical-Mathematical, to figure out the best career match.

Functional Requirements:

- **REQ-1**: The system must display a form with eight input fields for skill scores (0–20) on the career page.
- **REQ-2**: The system must validate that each score is a number between 0 and 20, showing an error message if invalid (e.g., "Linguistic score must be between 0 and 20").
- **REQ-3**: The system must handle non-numerical inputs by displaying an error like "Please enter valid numerical values."

4.4.2 Career Prediction

Description: The system uses a smart model to analyze skill scores and recommend a career that fits the user's strengths.

Functional Requirements:

- **REQ-4**: The system must process skill scores using the trained RandomForest model (career.pkl) to predict a career.
- **REQ-5**: The system must convert the model's numerical prediction to a career name (e.g., "Engineer") using label_encoder.pkl.
- **REQ-6**: The system must display the predicted career on a results page.

4.4.3 Data Pre-processing

Description: The system cleans and prepares data from a dataset to train the career prediction model accurately.

Functional Requirements:

- **REQ-7**: The system must load Career_Dataset.xlsx and remove unnecessary columns (e.g., Sr.No., Student).
- **REQ-8**: The system must map skill ratings (P1 to P8) to categories: 0 (Poor, 1–10), 1 (Average, 11–14), or 2 (Best, 15–20), with missing/invalid values set to 0.
- **REQ-9**: The system must normalize numerical skill scores (e.g., Linguistic) using scaler.pkl for consistent model input.

4.4.4 Model Performance Visualization

Description: The system creates visuals to show how well the career prediction model works, viewable in a PowerBI dashboard.

Functional Requirements:

- **REQ-10**: The system must generate a confusion matrix chart and save it as confusion_matrix.png.
- **REQ-11**: The system must save model accuracy in model_accuracy.csv.
- **REQ-12**: The system must ensure PowerBI can load confusion_matrix.png and model_accuracy.csv to display performance metrics.

4.5 Other Functional Requirements

4.5.1 Performance Requirements

- The system shall load within 5 seconds on a standard broadband connection
- ML predictions and dashboard visuals render within 5 seconds

4.5.2 Software Quality Attributes

- **Availability:** 99% uptime (if hosted online)
- **Usability:** Designed for users such as students, and career counsellors
- **Portability:** Cross-platform, it can be deployed on Windows, Linux, or cloud services
- **Maintainability:** Modular Python scripts with comments and unit tests
- **Reusability:** Dashboard and prediction modules can be adapted in future enhancements of project

4.6 Other Requirements

- **Database Requirements:** All uploads and logs are stored in excel or csv file, future versions may use PostgreSQL.
- **Legal Compliance:** Data used must be from open-source or public datasets; citation is required for third-party data.
- **Versioning:** Current version is v1.0. Planned future versions to include user login, real-time analytics.

5. System Design Description

5.1 Introduction

5.1.1 Purpose

This SDD outlines the design of the Career Recommendation Website, version 1.0, a platform to help students and young professionals pick careers that align with their skills and interests. It's aimed at developers, architects, testers, and project managers, providing a clear guide to build and implement the system, from the web interface to the machine learning model and visualization tools.

5.1.2 Scope

The Career Recommendation Website lets *users* input scores for eight skill or interest areas (e.g., logical-mathematical, linguistic), processes them using a RandomForest algorithm, and suggests career paths via a PowerBI dashboard. It doesn't provide job listings or real-time counselling but focuses on personalized, data-driven career guidance. The system aims to make career planning accessible, reduce reliance on outdated methods like Myers-Briggs, and align recommendations with job market trends, offering a modern alternative for students.

5.1.3 References

- Software Requirements Specification (SRS) for Habitat Harmony
- Scikit-learn, Pandas, and NumPy documentation
- Flask documentation
- Power BI Embedded
- IEEE 1016-2009 Standard for SDD

5.2 System Overview

The Career Recommendation Website is a web-based platform that takes user inputs, processes them with machine learning, and displays career recommendations via interactive visuals. It's a new system, not tied to existing products, built to address gaps in traditional career counselling by focusing on skills and job trends. The project stems from the need to help students make informed career choices using modern tech.

5.2.1 System Architecture

The architecture consists of the user, which will interact with the front-end (user-interface) of the website, then after taking the user inputs, the data is sent to the recommender system where the ML model is trained on supervised learning, which will compare the input with the dataset, and based on the most suitable answer, it will recommend the career to the user, and display a PowerBI dashboard where the user can also navigate through different career options and have an easy understanding through the dashboard.

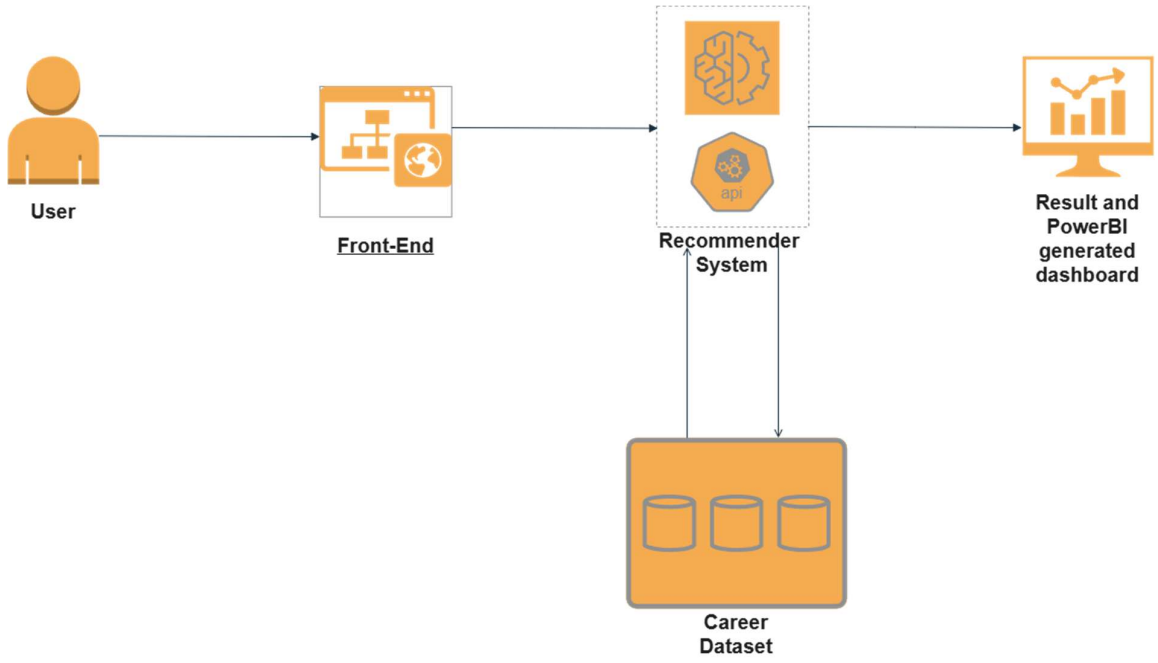

Fig. 1: System Architecture

The diagram illustrates the workflow of the Career Sync system, a user-friendly application designed to recommend careers based on individual skills and provide insightful visualizations. At its core, the process begins with the User, represented on the left, who interacts with the system to explore career options. The user starts by engaging with the **F**ront-End, depicted as a web interface (likely the Flask-based UI with pages like home.html and career.html). This front-end serves as the entry point, allowing users to input their skill scores for eight areas, such as Linguistic or Logical-Mathematical, in a simple form.

Once the user submits their scores, the data flows to the Recommender System, the brain of the application. This system, symbolized by a neural network icon, houses the trained RandomForest model (career.pkl). It processes the input data after normalization with scaler.pkl and augmentation with dummy ratings (P1 to P8) to predict a suitable career. The model leverages the Career Dataset, shown at the bottom, which contains historical skill data and corresponding professions from Career_Dataset.xlsx. This dataset, pre-processed to remove unnecessary columns and encode careers, ensures the model's predictions are accurate and relevant.

After the recommendation is generated, the system produces two outcomes. First, it sends the predicted career back to the front-end, which displays it on the result.html page for the user to see. Second, it feeds data into the Result and PowerBI Generated Dashboard, represented by a monitor with charts. This dashboard uses outputs like model_accuracy.csv to visualize the model's performance, offering users and administrators insights into accuracy. The entire flow user input, data processing, career recommendation, and visualization works seamlessly, making Career Sync a powerful tool for career guidance, enhanced by interactive and informative dashboards. This design ensures a smooth, end-to-end experience, bridging user interaction with data-driven insights.

5.3 Data Dictionary

A data dictionary for the dataset is a structured reference that provides detailed metadata about each data element within the dataset. It serves as a guide to help users, developers, and analysts understand the structure, meaning, and usage of the data, ensuring consistency and accuracy in its handling. For this specific dataset, which contains information about students' skills and recommended careers based on multiple intelligences (e.g., Linguistic, Musical, Bodily), the data dictionary acts as a blueprint to document the dataset's columns, their purposes, and technical specifications.

This data dictionary ensures that anyone working with a dataset from data scientists training the RandomForest model to UI developers building the Flask app, understands each field's role and limitations. It supports data integrity by enforcing bounds and mandatory fields, aids in troubleshooting (In this case, identifying why a score exceeds 20), and facilitates integration with tools like PowerBI for visualization.

Table 3: Data Dictionary

Entity Name	Data Element	Definition	Data Type	Storage Format	Scale	Bounds	Number	Mandatory Entry	Default Value	Create / Modify Functions	Functions to Read	Constraints
Career_Dataset	Sr.No.	Unique identifier for each record	Integer	int	Whole	1 to 3601	Number	Required	None	Data entry, preprocessing script	Data retrieval, model prediction	None
Career_Dataset	Course	Name of the career or job profession	Text	varchar	N/A	Any valid text	Text	Optional	None	Data entry, manual update	Dashboard display, report generation	None
Career_Dataset	Job profession	Specific job title or profession	Text	varchar	N/A	Any valid text	Text	Optional	None	Data entry, manual update	Dashboard display, report generation	None
Career_Dataset	Student	Identifier or name of the student	Text	varchar	N/A	S1 to S3601	Alphanumeric	Required	None	Data entry, manual update	Prediction engine, UI display	Protected
Career_Dataset	Linguistic	Score for linguistic skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	Musical	Score for musical skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	Bodily	Score for bodily skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	Logical-Mathematical	Score for logial-mathematical skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	Spatial-Visualization	Score for spatial-visualization skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	Interpersonal	Score for interpersonal skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	Intrapersonal	Score for intrapersonal skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	Naturalist	Score for naturalist skills	Integer	int	Whole	0 to 20	Number	Required	0	user input, preprocessing script	Model training, visualization	Range 0 to 20
Career_Dataset	s/p	Status indicating student / professional	Text	char	N/A	"s" or "p"	s/p	Required	"s"	mapping function	Prediction engine, UI display	Valid values: s,p

Career_Dataset	P1	Rating for linguistic skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best
Career_Dataset	P2	Rating for musical skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best
Career_Dataset	P3	Rating for bodily skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best
Career_Dataset	P4	Rating for logial-mathematical skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best
Career_Dataset	P5	Rating for spatial-visualization skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best
Career_Dataset	P6	Rating for interpersonal skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best
Career_Dataset	P7	Rating for intrapersonal skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best
Career_Dataset	P8	Rating for naturalist skills	Text	varchar	N/A	Poor, Avg, Best	Text	Required	None	mapping function	model input, dashboard display	Values: Poor, Avg, Best

5.4 Functional Design

5.4.1 Functionalities

The system lets users input eight skill scores (such as Linguistic, Logical-Mathematical, etc.), and predicts careers using RandomForest, and displays results via PowerBI.

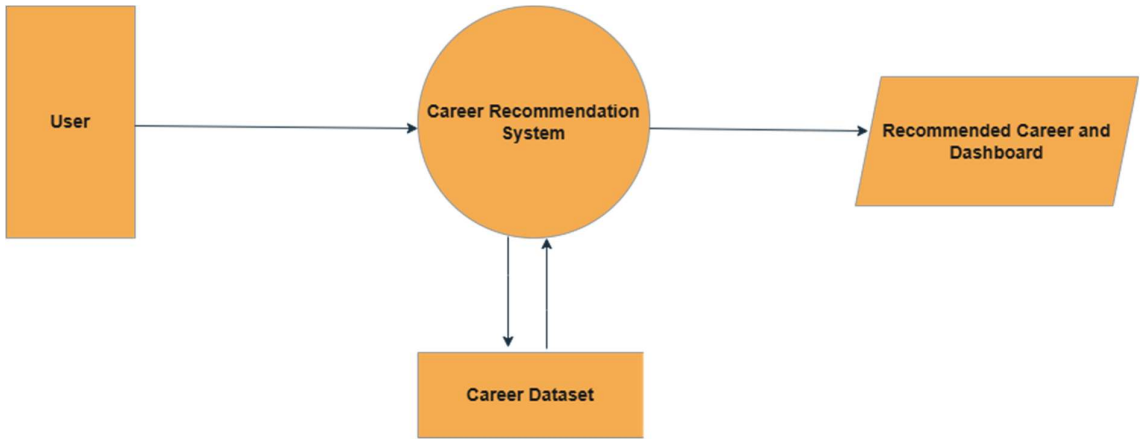

Fig. 2: Level – 0 Data Flow Diagram showing the flow of data between the components

The diagram represents a Level 0 Data Flow Diagram (DFD) for the Career Sync system, providing a high-level, simplified overview of how data moves through the system to deliver career recommendations and visualizations. At the top left, the User initiates the process by inputting skill scores (e.g., Linguistic, Logical-Mathematical) into the system, marking the starting point of data flow. This input is directed to the central process, the Career Recommendation System, depicted as the core circle, which serves as the primary processing unit. The system relies on the Career Dataset, shown at the bottom, a static data store containing pre-processed historical data from Career_Dataset.xlsx, including skill ratings and career mappings, to inform its analysis. Using a trained RandomForest model (career.pkl) and tools like scaler.pkl, the system processes the user's input to generate a career recommendation.

The data then splits into a single output stream leading to the Recommended Career and Dashboard**,** represented on the right. This external entity receives the processed output, where the predicted career (e.g., "Engineer") is presented to the user via the Flask app's result.html, and performance metrics (e.g., model_accuracy.csv, confusion_matrix.png) are visualized through a PowerBI dashboard. This Level 0 DFD abstracts the internal complexities, focusing on the main data flows: from user input to the recommendation system, supported by the dataset, and out to the user and dashboard. As of 11:26 PM IST on Wednesday, June 18, 2025, this diagram effectively captures the end-to-end data movement, ensuring a clear understanding of how Career Sync transforms user data into actionable career insights and visual analytics.

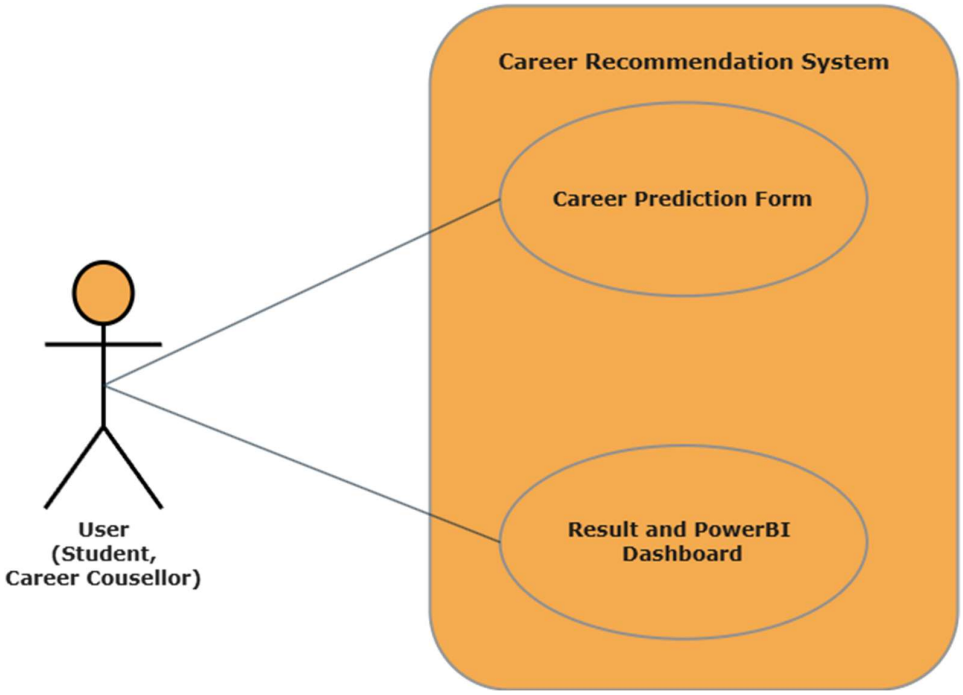

Fig. 3: Use Case Diagram for the Career Recommendation System

The diagram represents a Use Case Diagram for the Career Sync system, illustrating the interactions between the User (who can be a student or career counsellor) and the system's key functionalities. At the centre is the Career Recommendation System, which serves as the main actor facilitating the user's experience. The User, depicted on the left, initiates two primary interactions with the system. First, the Career Prediction Form use case allows the user to input their skill scores such as Linguistic, Logical-Mathematical, or Musical through a simple form (likely the Flask app's career.html). This action triggers the system to process the data using a trained RandomForest model (career.pkl) and recommend a suitable career based on the pre-processed Career Dataset from Career_Dataset.xlsx.

Second, the Result and PowerBI Dashboard use case enables the user to receive the predicted career, displayed on the result.html page, and explore performance insights through a PowerBI dashboard loaded with model_accuracy.csv and confusion_matrix.png. This dual-purpose use case ensures the user not only gets a personalized career suggestion but also understands the model's reliability.

6. Implementation

6.1 Machine Learning Implementation Algorithm

Start Algorithm

- **Step 1:** Load the libraries needed, like pandas for handling data, scikit-learn for machine learning, and matplotlib/seaborn for visuals.
- **Step 2:** Load the Career_Dataset.xlsx file, which contains data about people's skills and their matching careers.
- **Step 3:** Check all the column names in the dataset to make sure everything's there, such as skill ratings and job professions.
- **Step 4:** Remove columns like "Sr.No.", "Student", or "s/p" that are not needed for picking careers.
- **Step 5:** Turn skill ratings (i.e. Linguistic or Logical-Mathematical) into three categories: Poor (0) for scores 1–10, Average (1) for 11–14, and Best (2) for 15–20. If a score is missing, mark it as Poor (0).
- **Step 6:** Convert career names (like "Engineer" or "Teacher") into numbers using a LabelEncoder tool, so the app can work with them easily. Save this tool as label_encoder.pkl for later.
- **Step 7:** Separate the data into skills (like Linguistic, Musical, etc.) as features (X) and careers as the target (y).
- **Step 8:** Use a StandardScaler tool to make skill scores comparable by putting them on the same scale. Save this tool as scaler.pkl for future use.
- **Step 9:** Divide the data: 80% to train the app and 20% to test it.
- **Step 10:** Use a RandomForestClassifier to learn how skills lead to careers based on the training data.
- **Step 11:** Have the app predict careers for the test data. Check its accuracy and print a report with details like precision and a confusion matrix showing any mix-ups.
- **Step 12:** Store the app's accuracy score in a file called model_accuracy.csv to keep a record of how well it did.
- **Step 13:** Create a confusion matrix chart with seaborn to show which careers the app got right or wrong. Save it as confusion_matrix.png.
- **Step 14:** Store the trained RandomForest model as career.pkl so it is ready to recommend careers for users.
- **Step 15:** If anything goes wrong (such as a missing file), handle it gracefully to keep the process smooth.

End Algorithm

6.2 Career Sync Webpage Algorithm

Start Algorithm

- **Step 1:** Launch the Career Sync app using Flask, a tool that creates a website, and load libraries like pandas and joblib to handle data and models.
- **Step 2:** Bring in the pre-trained career recommendation model (career.pkl), the scaler for normalizing scores (scaler.pkl), and the label encoder for career names (label_encoder.pkl).
- **Step 3:** When someone visits the app's main page, display a welcoming homepage (home.html) to introduce Career Sync.
- **Step 4:** If the user goes to the prediction page (career.html), it will show a form where they can enter scores for eight skills, like Linguistic or Logical-Mathematical.
- **Step 5:** When the user submits the form on the prediction page, fetch their scores for the eight skills (Linguistic, Musical, Bodily, Logical-Mathematical, Spatial-Visualization, Interpersonal, Intrapersonal, and Naturalist) from the form.
- **Step 6:** Make sure each skill score is a number between 0 and 20. If any score is invalid, send the user back to the form with a message explaining which score needs fixing.
- **Step 7:** Put the user's skill scores into a DataFrame to prepare them for the career recommendation model.
- **Step 8:** Use the saved scaler to adjust the skill scores so they're on the same scale as the data used to train the model, ensuring fair predictions.
- **Step 9:** Add eight dummy columns (P1 to P8) set to "Average" (1) to match the model's expected format.
- **Step 10:** Feed the prepared data into the RandomForest model to predict a career number, then use the label encoder to turn that number back into a career name (like "Engineer" or "Teacher").
- **Step 11:** Display the predicted career on a results page (result.html) so the user can see their recommended career.
- **Step 12:** If the user enters non-numerical values, show the form again with a message asking for valid numbers.
- **Step 13:** Run the app in debug mode, so it's easy to test and fix any issues while it's live, waiting for users to explore their career options.

End Algorithm

7. Testing

7.1 Description of Testing

Unit Testing: It focuses on testing the individual components of the system in isolation. In this project, unit tests are applied to key Python functions, ML modules, and PowerBI dashboard such as:

- Data pre-processing functions (e.g., handling missing values, feature scaling)
- Individual machine learning models (e.g., validating prediction output of Random Forest on sample inputs)
- Testing the Flask interface
- Testing the PowerBI dashboard working

These tests ensures that each unit works correctly and returns the expected output without considering the interaction with other units.

Integration Testing: It ensures that different modules work correctly when integrated together. In the context of the project, it includes the following:

- Checking if the cleaned data from the pre-processing module correctly flows into the ML models
- Verifying that Flask UI interacts properly with the backend (Python scripts and ML models)
- Checking if the PowerBI Embedded works properly and displays the dashboard

System Testing: It evaluates the entire project as a complete solution from the user's perspective. It includes the following:

- Run the pre-processing script on Career_Dataset.xlsx to confirm it removes extra columns, maps skill ratings to 0/1/2, normalizes skills, and encodes careers as numbers.
- Feed a sample of cleaned data to the career.pkl model, ensuring it predicts a valid career number, which label_encoder.pkl converts to a correct career name.
- Start the Flask app, verify the homepage (home.html) and career form (career.html) load, and ensure the form accepts eight skill inputs (1–20).
- Submit valid and invalid (e.g., 25 or "abc") scores via the form. Confirm valid inputs yield a career result, invalid ones show error messages, and predictions match pre-processing outputs.
- Ensure the PowerBI dashboard loads model_accuracy.csv displaying accuracy of the model clearly. Verify error handling for missing files.

7.2 Test Cases

System Testing

Table 4: System Testing Table

Test Case #	Test Case Name	Test Case Description	Inputs	Expected Output	Actual Output	Status
1	Working of Predict Now button	Validating the working of the Predict Now button	Clicking the Predict Now button	It should redirect to the prediction page	It should redirect to the prediction page	Pass
2	Valid skill rating input	Validating the skill input rating fields by giving valid inputs	Values ranging between 1 – 20	It should not display any error message	It should not display any error message	Pass
3	Invalid skill rating input	Validating the skill input rating fields by giving invalid inputs	Values out of range (e.g. 0 or > 20)	It should display error message if invalid value is identified	It should display error message if invalid value is identified	Pass
4	Missing skill rating input	Validating the skill input rating fields by leaving the inputs blank	None	It should display message telling user to enter a value	It should display message telling user to enter a value	Pass
5	Working of Predict Career button	Validating the working of the Predict Career button	Click on the Predict Career button	It should redirect to results page if inputs are valid, else stay on page	It should redirect to results page if inputs are valid, else stay on page	Pass
6	Dashboard Visualization Display	Validating if PowerBI dashboard loads correctly after prediction	Valid skill scores followed by clicking Predict Career	Dashboard with graphs of skill scores and recommended careers	Dashboard with graphs of skill scores and recommende d careers	Pass
7	Recommendation Accuracy	Checking if relevant careers are shown for given skills	High scores in logical, math, programming skills	Careers like Data Scientist, Software Engineer, etc.	Careers like Data Scientist, Software Engineer, etc.	Pass

8	Partial Input Handling	Checking behaviour when only some skill values are entered	4 out of 8 fields filled	Prompt user to complete all inputs	Prompt user to complete all inputs	Pass
9	Non-numeric Input Validation	Checking system response to alphabetic or symbol input	Input: "ten", "@", etc. in fields	Show validation error message	Show validation error message	Pass
11	Responsiveness Across Devices	Verifying UI display on mobile/tablet	Access site via mobile/tablet	Responsive layout without breaking elements	Responsive layout without breaking elements	Pass
12	Invalid URL Navigation	Behaviour when accessing a wrong or non-existent page	Enter URL `/career/resultxyz`	Show 404 or redirect to homepage	Custom 404 message shown	Pass
13	Model Processing Time	Measuring time taken to return recommendations	Submit valid input	Results within 3 seconds	Results returned in 2.5 seconds	Pass
14	Skill Score Edge Case (Minimum Boundary)	Inputting all skill scores as 1	All inputs = 1	Recommend beginner-level careers	Beginner-level careers shown	Pass
15	Skill Score Edge Case (Maximum Boundary)	Inputting all skill scores as 20	All inputs = 20	Recommend advanced-level careers	Advanced-level careers shown	Pass

8. Reports

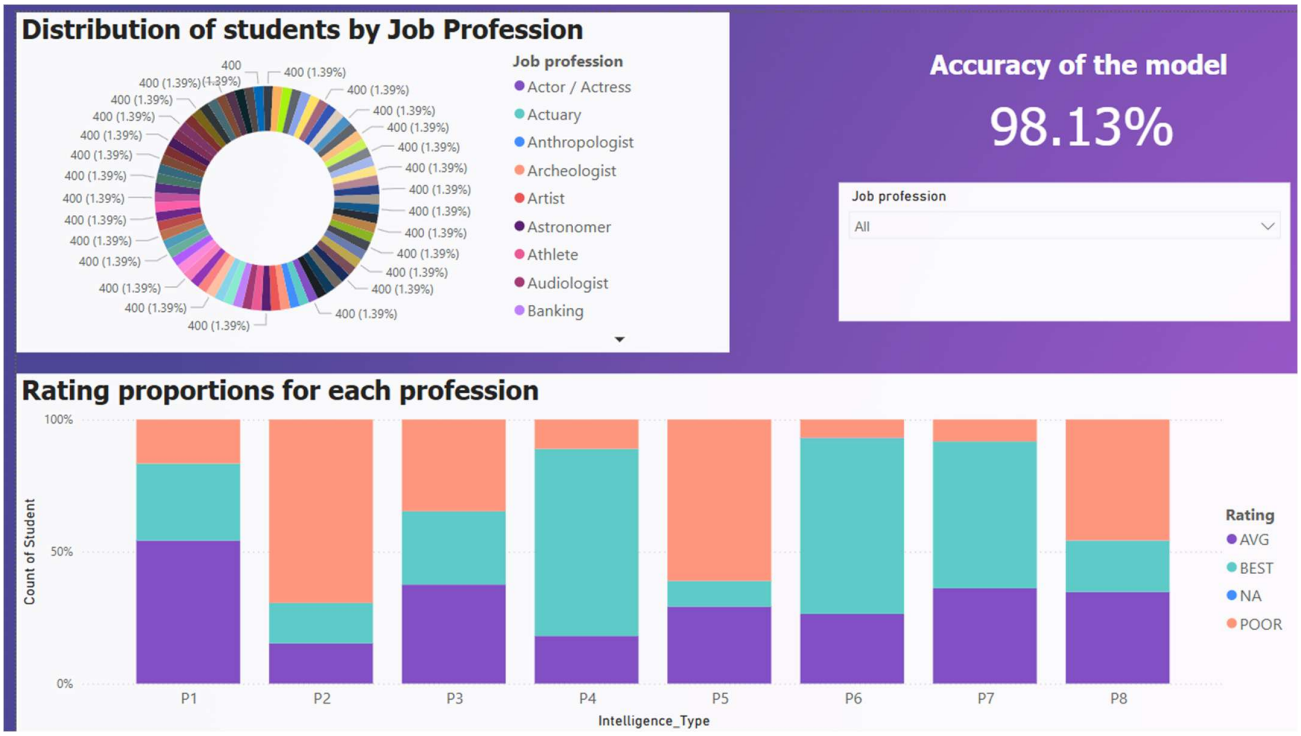

Fig. 4: Career Recommendation Dashboard which works on RandomForest model

The Career Recommendation Dashboard (shown in Fig. 4) that uses the RandomForest model to help guide students toward the perfect career! This dashboard is like a friendly career counsellor packed with colourful charts and helpful insights. It's designed to break down how students are distributed across different job professions and show how their skill ratings stack up, all while showing an impressive accuracy of 98.13%. Here's a simple and detailed explanation of what you're seeing:

Top Left: Distribution of Students by Job Profession

Imagine a big, colorful pie chart that shows where all the students in the dataset (3,601) might end up based on their skills. This section is called the "Distribution of Students by Job Profession." Each slice of the pie represents a different career, and every slice is labelled with a job title like "Actor / Actress," "Astronomer," "Athlete," or even "Banking." The chart tells us that each profession has an equal share about 400 students (1.39% each) because the data is evenly spread across 72 unique careers.

Top Right: Accuracy of the Model

Next to the pie chart, you'll see a bold purple box proudly displaying the "Accuracy of the model" at 98.13%**.** This is like a grade for the RandomForest model, showing how good it is at predicting the right career for each student based on their skills. An accuracy this high means that nearly 98 out of every 100 predictions are accurate.

Bottom: Rating Proportions for Each Profession

Below these, there's a set of bar charts under "Rating Proportions for Each Profession." These charts are like a snapshot of how students' skill ratings (POOR, AVG, BEST, or NA) are distributed across the eight intelligence types (P1 to P8, corresponding to Linguistic, Musical, Bodily, etc.). Each bar represents one intelligence type, and its split into different color sections:

- Purple (AVG): Shows the percentage of students with an average rating.
- Teal (BEST): Highlights those with the best skills in that area.
- Orange (POOR): Indicates students with lower skills.
- Blue (NA): Marks any missing or not applicable ratings.

For example, the bar for P1 (Linguistic) might show a mix of purple, teal, and orange, telling you how many students rate average, best, or poor in linguistic skills. The height of each section shows the count of students, with 100% at the top, so you can see at a glance which skills are strong or need work across the dataset. This helps identify patterns like if most students have strong Bodily skills (P3) for careers like Athlete, or if Musical skills (P2) lean toward Recording Engineer.

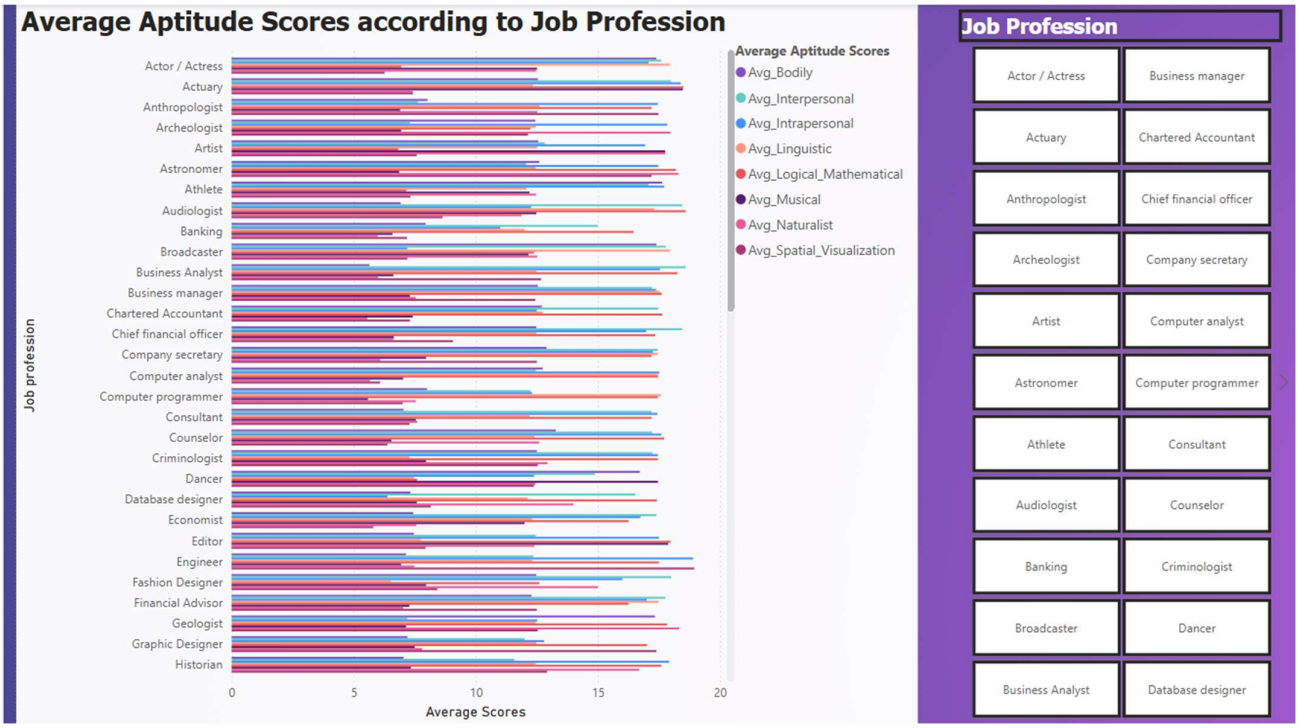

Fig. 5: Career Recommendation Dashboard showing the average ratings for skills based on the career

The Career Recommendation Dashboard (shown in Fig. 5) highlights the average aptitude scores for various skills across different job professions. This dashboard is like a personalized guide that helps you see how your skills align with different careers. It's built using data from the Career_Dataset.xlsx and presented in a clear, colourful way. Here's a simple and detailed breakdown of what this report tells us:

Main Section: Average Aptitude Scores according to Job Profession

Picture a series of horizontal bar charts, where each bar represents a job profession, like "Actor / Actress," "Astronomer," "Engineer," or "Dancer." These bars stretch across the chart to show the average scores (out of 20) for eight different skill types, measured through the multiple intelligences from the dataset. The skills are color-coded for easy reading:

- Purple (Avg_Bodily): Average score for bodily-kinesthetic intelligence (e.g., physical coordination).
- Teal (Avg_Interpersonal): Average score for interpersonal skills (e.g., working with others).
- Blue (Avg_Intrapersonal): Average score for intrapersonal skills (e.g., self-awareness).
- Orange (Avg_Linguistic): Average score for linguistic skills (e.g., language ability).
- Red (Avg_Logical Mathematical): Average score for logical-mathematical skills (e.g., problem-solving).
- Pink (Avg_Musical): Average score for musical intelligence (e.g., rhythm and sound).
- Magenta (Avg_Naturalist): Average score for naturalist skills (e.g., nature understanding).

- Light Purple (Avg_Spatial Visualization): Average score for spatial-visualization skills (e.g., visualizing objects).

Each bar is stacked with these colours, showing how the average scores for each skill contribute to the overall profile of that job. For example:

- An "Astronomer" bar might show a high red (Logical-Mathematical) score and a decent light purple (Spatial-Visualization) score, reflecting the need for math and spatial skills.
- A "Dancer" bar might have a taller purple (bodily) section, emphasizing physical talent.

The x-axis runs from 0 to 20, so you can see at a glance which skills are strongest for each profession. Some bars might reach higher averages (closer to 20), while others stay lower, depending on the career's demands.

Right Side: Job Profession List

On the right, there's a handy dropdown menu under "Job Profession," listing all 72 careers from the dataset, such as "Actor / Actress," "Chartered Accountant," "Graphic Designer," and "Veterinarian." This menu pairs each profession with a related job (e.g., "Actor / Actress" with "Business manager"), possibly suggesting complementary career paths. You can scroll through or select a specific job to focus the chart on that profession's skill averages, making it easy to zoom in on what interests you.

9. Conclusion

The Career Sync system leverages a RandomForest model as its core predictive engine, and when pitted against alternative models like K-Nearest Neighbors (KNN) and Decision Tree, it stands out with an impressive test set accuracy of 98%. This high accuracy reflects the model's ability to correctly predict career recommendations based on user-provided skill scores, such as Linguistic, Logical-Mathematical, and Musical. Beyond accuracy, the RandomForest model excels with an average precision, recall, and F1-score of 0.99 across various career categories, indicating its reliability in minimizing false positives and effectively identifying true positives, ensuring balanced performance. This robustness is particularly evident in its ability to recommend careers tailored to diverse user profiles successfully matching technically skilled individuals (e.g., to roles like engineers) and those with creative interests (e.g., artists or writers) with precision.

The system's effectiveness shines through its handling of a wide range of inputs, from varying skill levels to different user backgrounds, demonstrating adaptability and consistency. Complementing this, the PowerBI dashboard enhances the user experience by providing a powerful visualization and analysis tool. It displays key metrics like accuracy and confusion matrix data from model_accuracy.csv and confusion_matrix.png, making the model's performance transparent. A standout feature is the slicer, which allows users to filter results, enabling them to focus on specific recommended careers or analyses trends for particular skill sets. This combination of a high-performing RandomForest model and an interactive PowerBI dashboard positions Career Sync as a robust, user-friendly solution for personalized career guidance.

Table 5: Table comparing accuracy metrics of different machine learning models

Algorithm	Accuracy	Precision	F1-score
Random Forest	98	99	99
Decision Tree	97	97	97
KNN	91	91	91

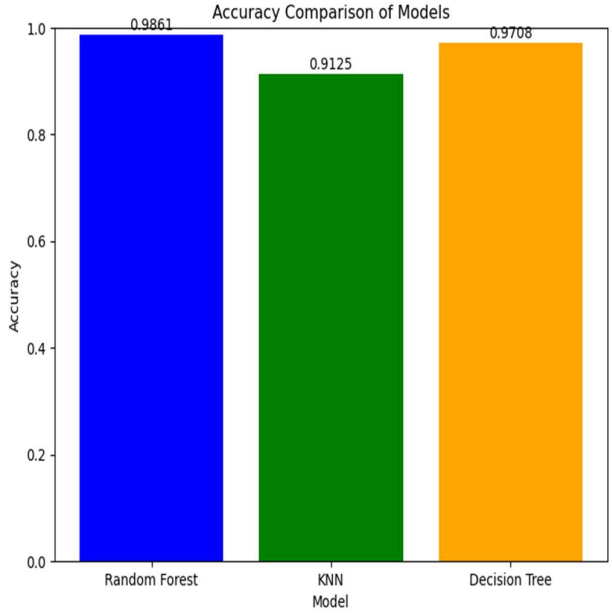

Fig. 6: Graph displaying accuracy scores of different machine learning models

10. Future Enhancements

- **Interactive Career Suggestions**: Getting more than just one career idea! After you submit your skill scores, the result.html page will show a list of your top 3 to 5 career options, like "Engineer," "Teacher," or "Artist." Each suggestion comes with a short description of it as a quick peek into what the job's like and a list of skills you'll need. It's like having a career buffet to choose from, making it easier to find the perfect fit for you.
- **Skill Development Recommendation**: If there's a gap between your current skills and what the job needs, the system will suggest fun ways to improve, like online courses on platforms like Coursera or even books to read. It's like a roadmap to help you grow into the career you prefer.
- **User Profile and History**: With a simple login system, you can save your profile and all your past predictions. This lets you look back at your journey seeing how your skills have changed or which careers you explored before. It's a handy way to track your progress and revisit ideas whenever you want.
- **Feedback Form**: After you see your recommended career on result.html, you can fill out a quick feedback form to rate how accurate it feels. Your input will be used to fine-tune the model, making it smarter for you and future users.
- **Personalized Email Notifications**: After submitting your form, you'll receive an email with your career recommendation and a link to the PowerBI dashboard. You can also opt-in to get periodic career tips, like job market updates or skill-building ideas, delivered right to you.
- **Predictive Trend Analysis**: PowerBI dashboard will now show predictions about which careers might grow in demand, using past data to spot trends. Whether its tech jobs or creative roles, this feature helps you pick a career that's not just a good fit now but will thrive later.

11. Bibliography

[1] Dutta, Ishaan, et al. "Building a recommendation system for career advice for students from professionals." 2022 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2022.

[2] Shah, Arya, et al. "Review of Approaches Towards Building AI Based Career Recommender & Guidance Systems." ScienceOpen Preprints (2024).

[3] M. Rane, S. Kalal, J. Chandegara, T. Kakkad, V. Jain and S. Jagtap, "Career Prediction Website using Machine Learning," 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2023

[4] Padma, E., et al. "Career guidance system for students using machine learning." Challenges in Information, Communication and Computing Technology. CRC Press, 2025.

[5] S. K. A. S, S. S, A. S. M and S. K. S, "Machine Learning based Ideal Job Role Fit and Career Recommendation System," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023.

[6] "93% of Indian students are aware of just seven career options" India Today. https://www.indiatoday.in/education-today/news/story/93- indian-students-aware-of-just-seven-career-options-what-areparents-doing-wrong-1446205-2019-02-04.

[7] Shahab, Shehba, "NEXT LEVEL: A COURSE RECOMMENDER SYSTEM BASED ON CAREER INTERESTS" (2019). Master's Projects. 684

[8] Student Career Guidance System for Recommendation of Relevant Course Selection by E K Subramanian, Ramachandran, International Journal of Recent Technology and Engineering (Volume-7, Issue-6S4, April 2019)

[9] V, S. & S, S. (2019). Data-Science Recommendation System using Semantic Technology. International Journal of Engineering and Advanced Technology Regular Issue, 9(1):2592–2599.

[10] Adapa, S., Gandi, N., Alekya, V., Lakshmi Durga, G. (2021). Learning Style Recommender System Using VAK Technique and Machine Learning. In: Satapathy, S.C., Bhateja, V., Ramakrishna Murty, M., Gia Nhu, N., Jayasri Kotti (eds) Communication Software and Networks. Lecture Notes in Networks and Systems, vol 134. Springer, Singapore.

[11] C. D. Casuat and E. D. Festijo, "Identifying the Most Predictive Attributes Among Employability Signals of Undergraduate Students," 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Langkawi, Malaysia, 2020.

[12] K. O. Akyina, G. Oduro-Okyireh, and B. Osei-Owusu, "Assessment of the Rationality of Senior High School students' Choices of Academic Programmes in Kwabre East District of Ghana," Journal of Edu. and Practice. 2014.

[13] B. Redmond, S. Quin, C. Devitt, and J. Archbold, "A Qualitative Investigation into the Reasons Why Students Exit From the First Year of Their Programme and UCD," University College Dublin, School of Applied Social Science, October, 2011.

[14] M. D. Eremie, "Comparative Analysis of Factors Influencing CareerChoices among Senior Secondary School Students in Rivers State, Nigeria," Arabian Journal of Bus. and Mgt Review.

[15] A. F. Egunjobi, T. M. Salisu, and O. I. Ogunkeye, "Academic profile and career choice of fresh undergraduates of library and information science in a Nigerian University of Education," Annals of Library and Information Studies, 2013.

[16] I. A. Durosaro, and M. A. Nuhu, "An evaluation of the relevance of career choice to school Subject selection among school going adolescents in Ondo state," Asian Journal of Mgt Sci. and Edu. 2012.

12. User Manual

Introduction

The Career Recommendation System is a fantastic tool designed to guide you toward a suitable career by analyzing your self-assessed ratings across eight key skill areas, such as Logical-Mathematical, Linguistic, and Bodily, based on your strengths and weaknesses. Users start by rating yourself on these skills (e.g., 1 to 20), and the system sends your input to a highly accurate RandomForest machine learning model, trained on a dataset like Career_Dataset.xlsx having 3601 records, achieving an impressive 98.13% accuracy. Once processed, the model recommends a career tailored to your profile, and an interactive PowerBI dashboard pops up, letting you explore your suggested job, compare skill averages, and navigate other career options with engaging visuals making your career exploration reliable.

Getting Started

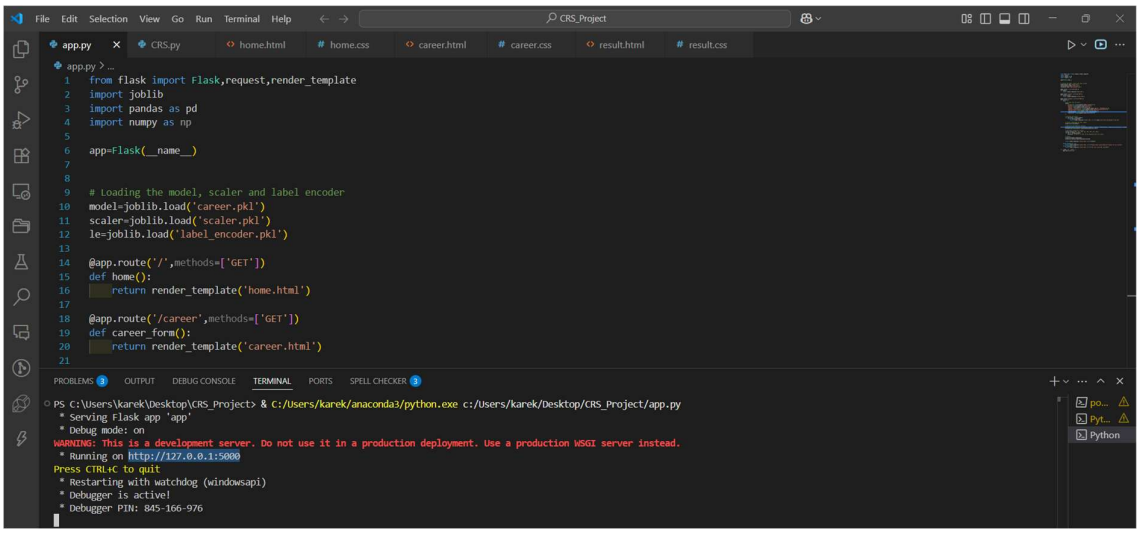

Fig. 7: Starting the localhost and setting the environment for website to run

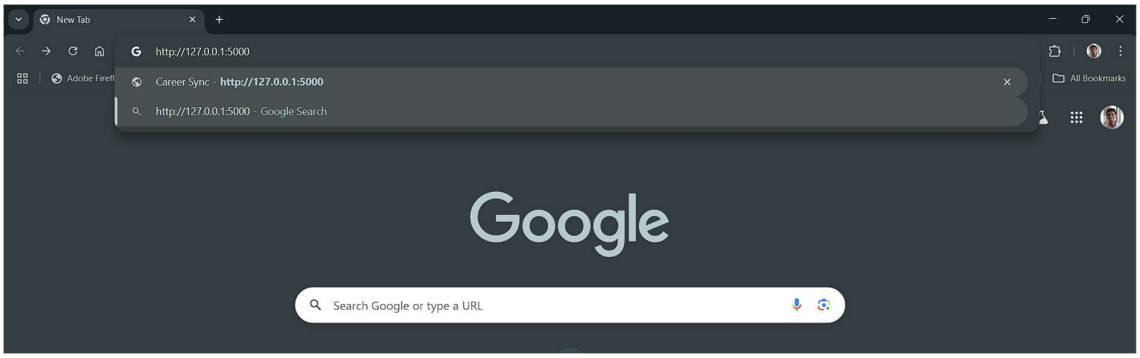

Fig. 8: Copy the localhost address and paste it in the browser

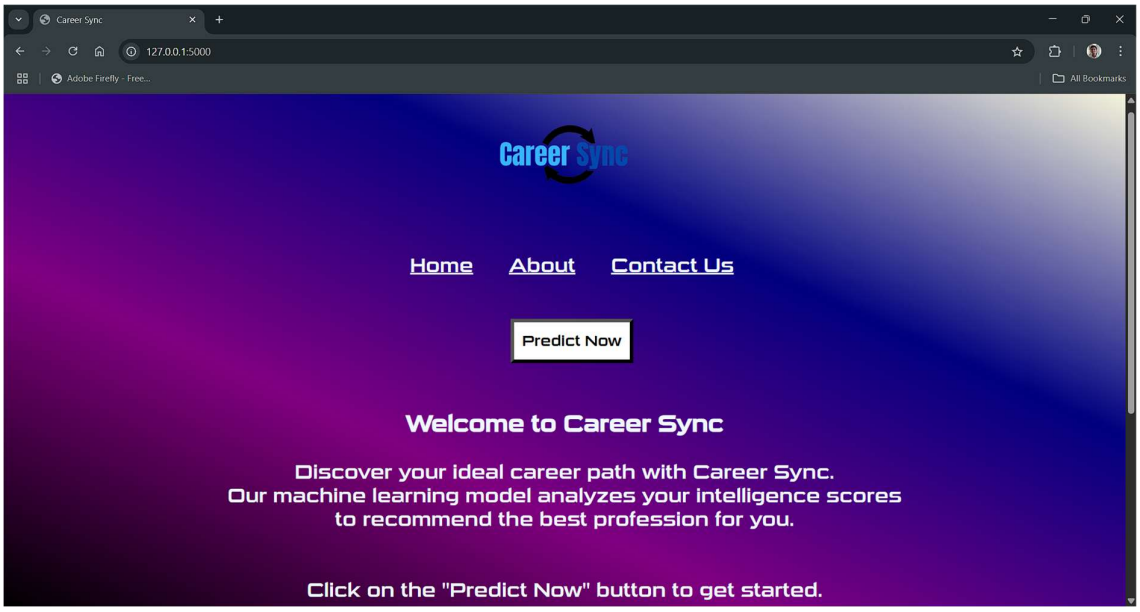

Fig. 9: The landing page of the website

- First, we have to run the app.py file which is based on the Flask framework and has all the web-pages routed. It will also start the localhost to run the application.
- After running in the terminal localhost IP address will be displayed, copy and paste it into the browser, now the application is ready to use.

Career Prediction Page

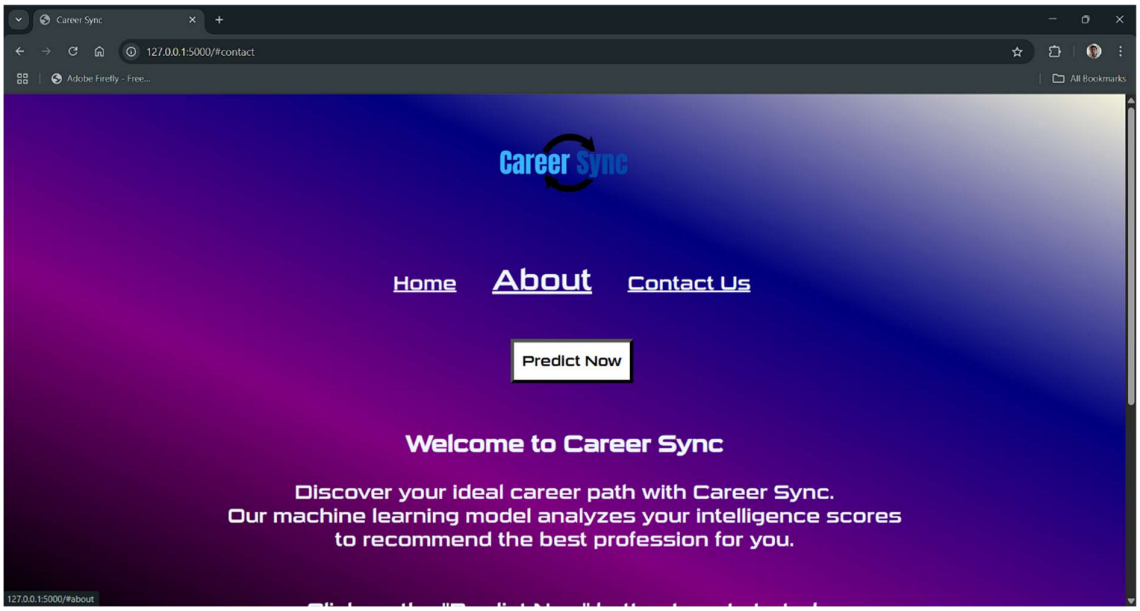

Fig. 10: Navigating through different sections of the web page

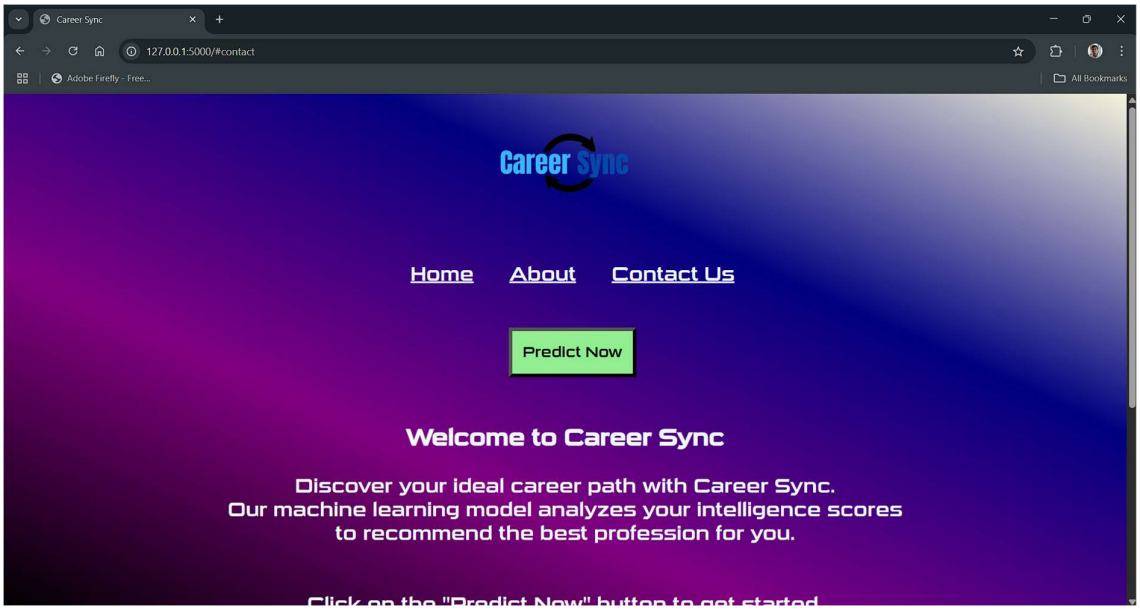

Fig. 11: Click on the Predict Now button to redirect to career prediction page

- The User can navigate through different sections of the by clicking on their section names.
- By clicking on the “Predict Now” button, the users will be redirected to the rating form page.

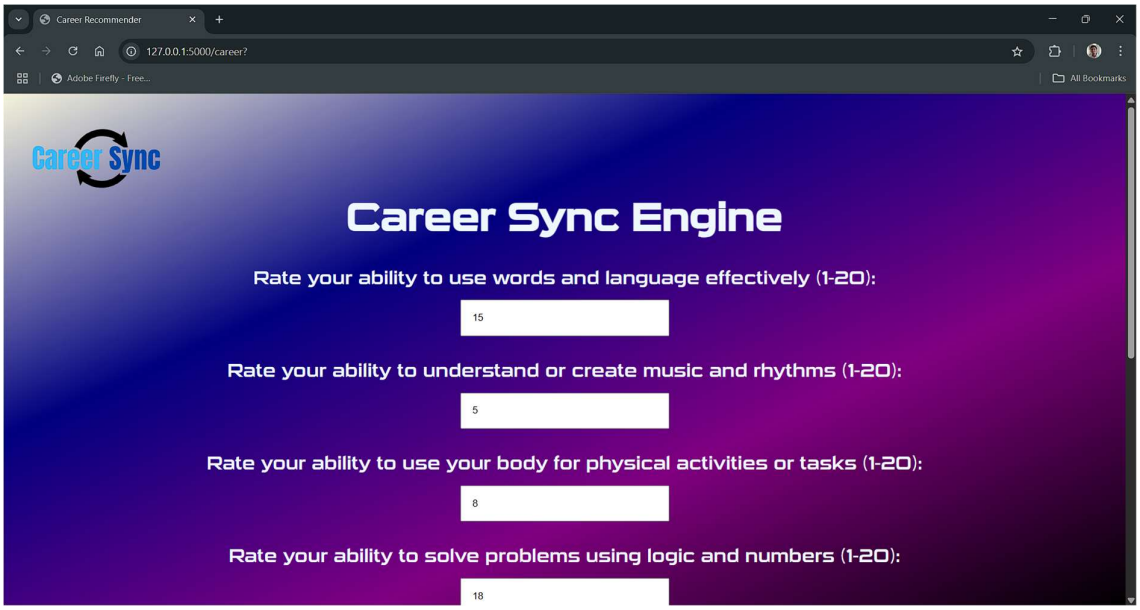

Fig. 12: User giving ratings as input based on their knowledge of the skills

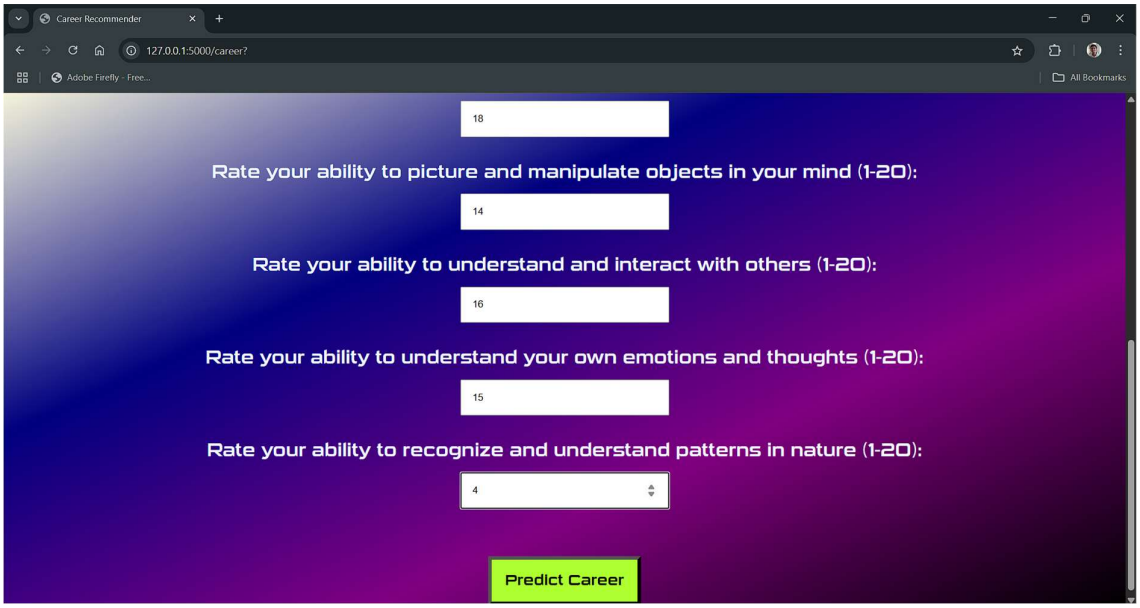

Fig. 13: Click on the Predict Career button to get the result

- In this page, Users have to rate themselves on eight parameters (such as Linguistic, Bodily, etc.) in between 1 to 20, and click on the “Predict Career” button.

Result Page

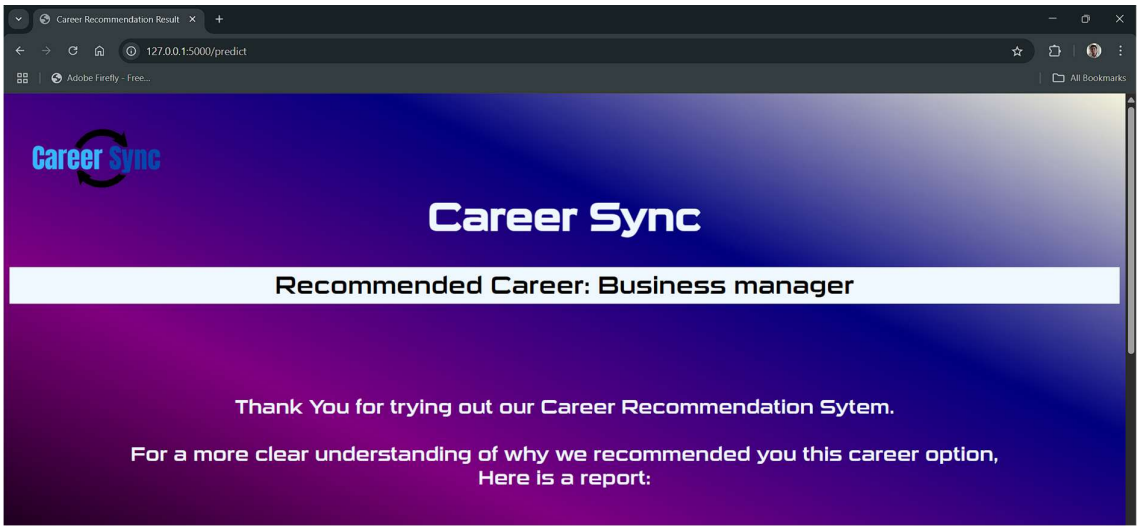

Fig. 14: Result Page displaying the recommended career

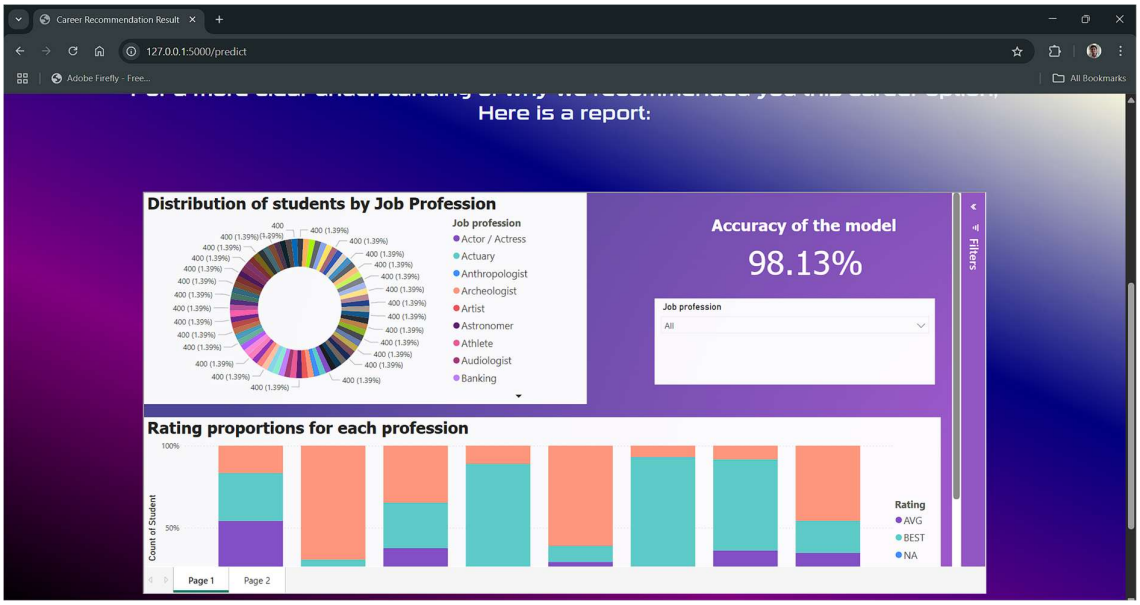

Fig. 15: Dashboard displaying the career distribution, accuracy of the model and skill ratings in terms of POOR, AVG, and BEST

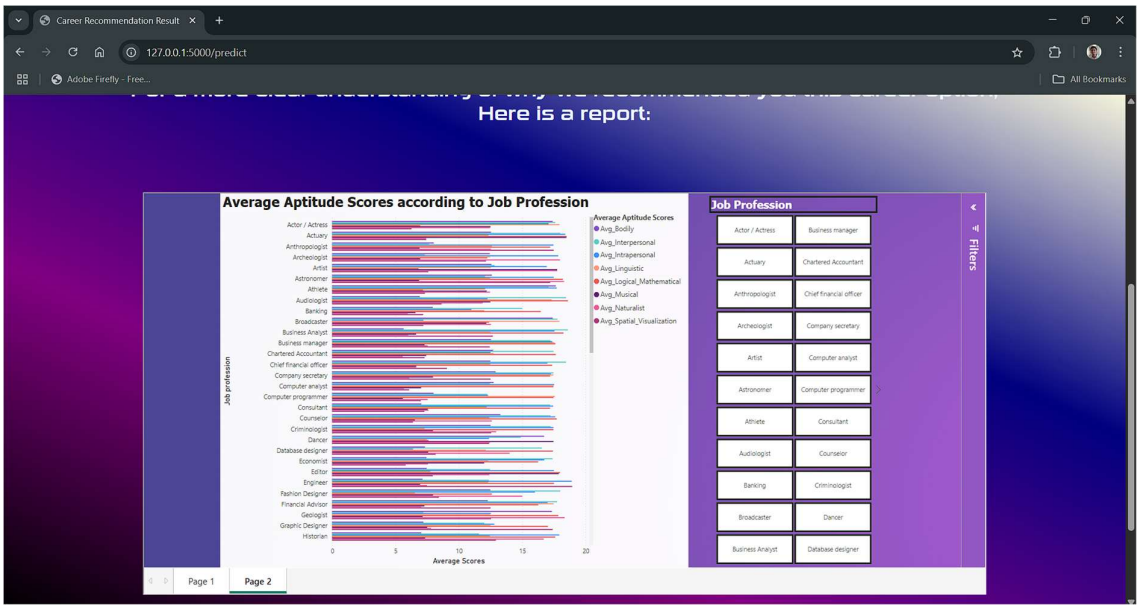

Fig. 16: Dashboard displaying the average skill ratings depending on the career

- The result page displays the resultant career, along with an interactive PowerBI dashboard which will help the users to understand more clearly and effectively about the skills required for the recommended career and also navigate through other career options visually.

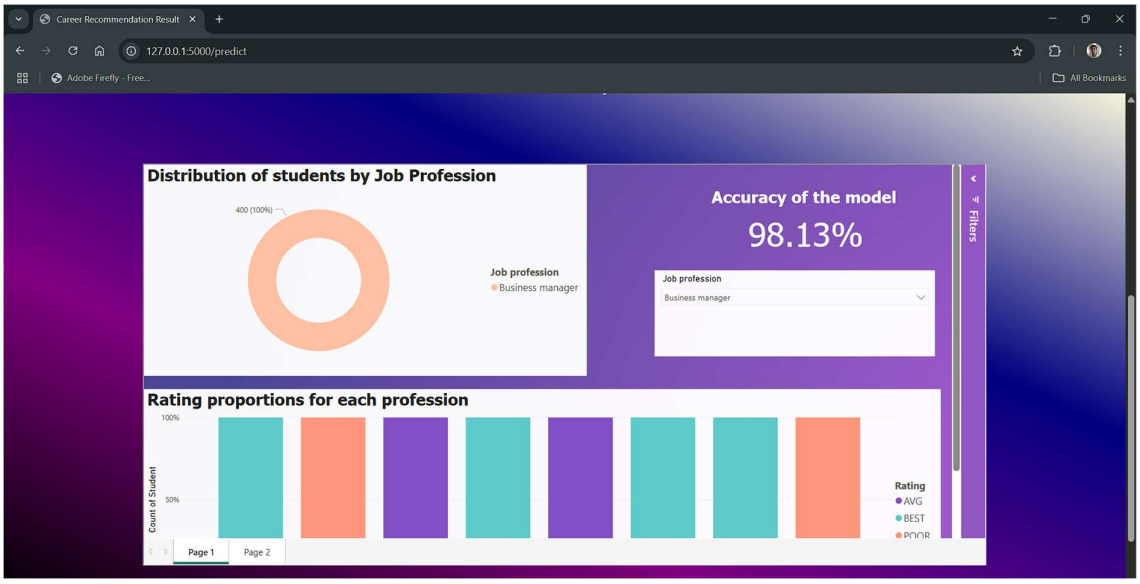

Fig. 17: User selecting the career through slicer and analyse the results

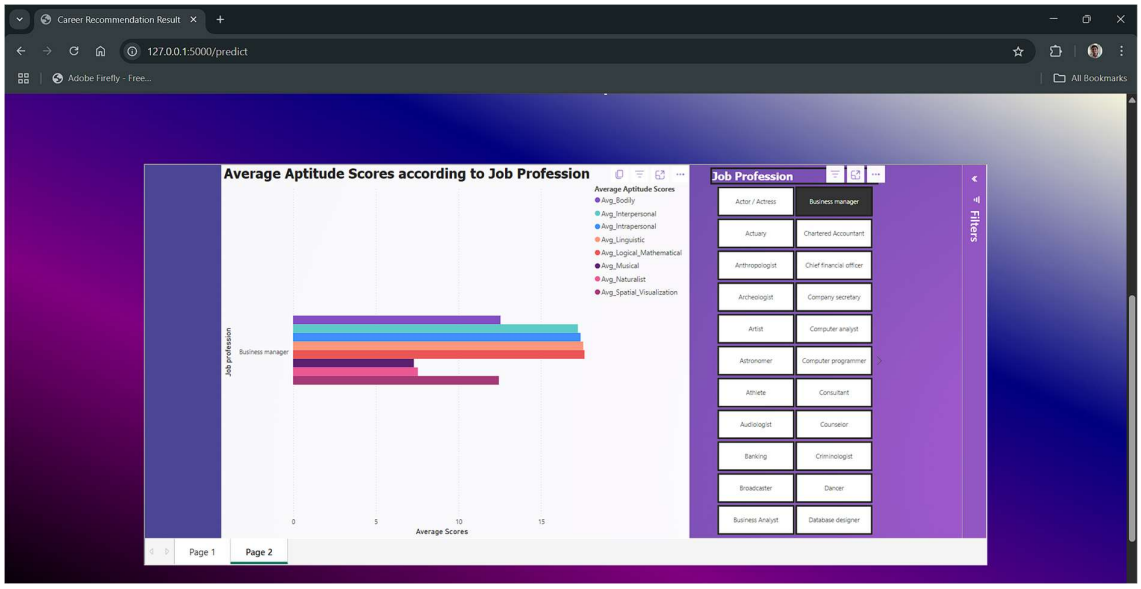

Fig. 18: User selecting career through slicer and getting the average skill scores

- The slicer in the dashboard helps the user to navigate through their recommended career and explore other career options as well.
- In the 1st dashboard, the slicer helps the users to check what percentage of people are in that career, and their skill rating for each parameter.
- In the 2nd dashboard, the slicer helps the user to check the average skill ratings for all parameters based on the chosen career.

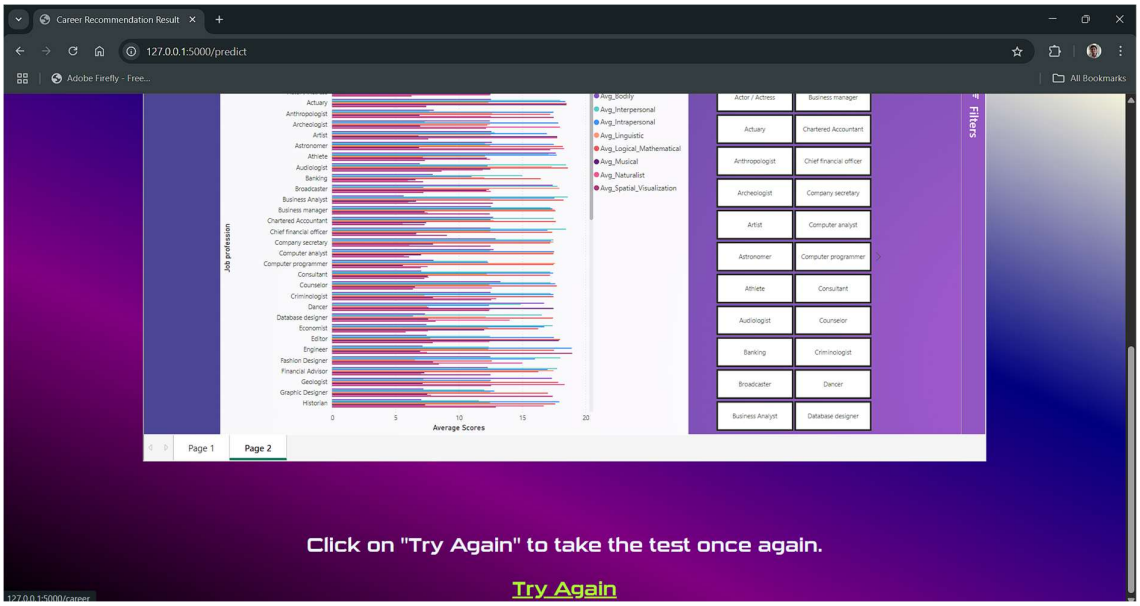

Fig. 19: User clicks on Try Again to retake the quiz

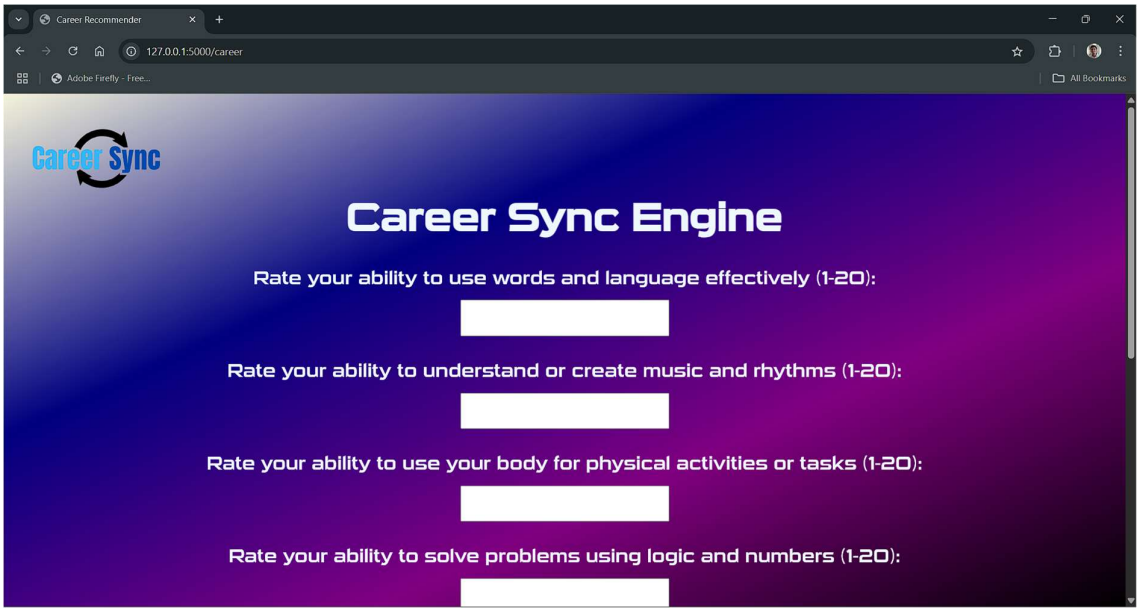

Fig. 20: A fresh new skill rating quiz page

- In case if user wants to retake the quiz, he has to click on “Try Again”, which will redirect them to fresh skill form.