Employee Attrition Analysis

Group No: 2

Members:

Kubilay Kürtür
Ahmet Yiğit Özkoca
Beyza Akgün
İlayda Kasapçopur

CMPE 331 Final Report
2023 Fall

Employee Attrition Prediction Web Application Project

Introduction to the Project

In an era where human capital is as crucial as financial assets, the ability to understand and manage employee turnover becomes imperative for any forward-thinking organization. The project that is embarked upon addresses a fundamental yet often overlooked aspect of corporate health - employee attrition. With the evolving dynamics of the modern workplace, companies increasingly seek reliable methods to not only anticipate but also intelligently respond to workforce changes. This project steps into this space, offering a data-driven approach to predicting employee attrition. By harnessing the power of machine learning, a tool is provided that is not just reliable for the company in terms of strategic planning and cost management, but also beneficial for employees in fostering a more engaging and satisfying work environment. Choosing to focus on this problem reflects our commitment to enhancing organizational resilience and employee well-being, ultimately contributing to a more stable and productive workforce.

Objective of the Project

The phenomenon of employee attrition, characterized by the unforeseen departure of employees from an organization, poses a significant challenge in today's corporate landscape. This project addresses this challenge head-on by leveraging advanced machine learning algorithms to predict employee attrition within a few months. By meticulously analyzing a comprehensive dataset encompassing various employee attributes – including age, monthly income, work-life balance, business travel frequency, and tenure at the company – valuable insights and actionable feedback are provided to the human resources departments of companies and organizations. The aim is to transform this predictive analysis into a strategic tool, enhancing the efficacy of workforce management and planning.

Importance of the Project

In the complex realm of human resource management, understanding and mitigating employee turnover is more than just a tactical move – it's a strategic necessity. This project stands at the forefront of this endeavor, offering a crucial advantage for companies in optimizing their hiring strategies and managing turnovers. The financial implications of employee attrition are manifold, encompassing the costs of recruitment, training, and the loss of productivity. By identifying the risk factors associated with employee departure, this project empowers organizations to take preemptive actions. These actions are not just about retaining talent but also about fostering an environment where employees thrive. Through the analysis, companies gain the ability to pinpoint and address the underlying causes of attrition, be it through improving workplace conditions, offering career development opportunities, or other targeted retention strategies. In essence, this project is a pivotal step towards creating a more stable, satisfied, and productive workforce.

Roles

This project focuses on predicting employee attrition using machine learning algorithms and presenting the results through a user-friendly web interface. This endeavor required a coordinated effort across various domains, including idea conceptualization, data handling, machine learning implementation, API integration, front-end development, and rigorous testing.

Kubilay Kürtür (Project Conceptualization and Backend Development):

- Conceptualized the project idea, identifying the need for an advanced solution to predict employee attrition.
- Sourced and prepared the datasets necessary for machine learning analysis.
- Led the development of the machine learning model, applying sophisticated algorithms to ensure accurate predictions.
- Contributed significantly to the backend development, integrating the machine learning model into the server-side logic.

- Assisted in API integration, ensuring seamless communication between the backend and frontend.
- Conducted comprehensive testing to validate the functionality and reliability of the machine learning implementation and backend processes.

Ahmet Yiğit Özkoca (API Integration and Testing):

- Played a pivotal role in API development, facilitating effective data exchange between the backend and the frontend.
- Worked collaboratively to integrate the API, enabling the frontend application to access and display the processed data.
- Undertook extensive testing, focusing on both the API and the overall system functionality to ensure robust performance and reliability.

Beyza Akgün, İlayda Kasapçopur (Frontend Development and Design):

- Jointly responsible for the design and development of the frontend interface, crafting an intuitive and responsive web application.
- Implemented the frontend features, translating the project requirements into a visually appealing and user-friendly interface.
- Ensured the frontend seamlessly integrated with the backend API, providing a smooth user experience.
- Conducted thorough testing of the frontend components, focusing on usability, design consistency, and responsiveness across various devices.

- Jointly responsible for the design and development of the frontend interface, crafting an intuitive and responsive web application.
- Implemented the frontend features, translating the project requirements into a visually appealing and user-friendly interface.
- Ensured the frontend seamlessly integrated with the backend API, providing a smooth user experience.
- Conducted thorough testing of the frontend components, focusing on usability, design consistency, and responsiveness across various devices.

Timeline

- Finding the appropriate dataset
- Developing a machine learning model
- Optimizing the model
- Creating a Flask API
- Researching suitable designs and templates for the website
- Planning the necessary website pages for the project
- Frontend coding
- Integrating the frontend and backend with the API
- Backend for registration and login pages with a Spring project
- Testing the buttons and the website
- API bug fix
- Adding necessary new content for the website design
- Adding password pattern

Requirements

1. **Data Acquisition and Preparation**
    - The system should allow uploading Excel and CSV files.
    - The system should inform the user about the file format.
    - The system should warn the user if a wrongly formatted file is uploaded.
    - The system should be able to process and validate the data in uploaded files for compatibility with the machine learning model.

2. **Machine Learning and Data Processing**
    - The system should accurately perform employee attrition prediction using machine learning algorithms.
    - The backend should efficiently handle data processing and analysis to return meaningful insights.

3. **Results Generation and Dissemination**
    - The system should be able to write and return results in both CSV and Excel formats.
    - The system should provide a clear and concise presentation of the analysis results, highlighting key attrition risk factors.

4. **User Management and Accessibility**
    - The system should have a user-friendly interface, making it easy for users with varying levels of technical expertise to operate.

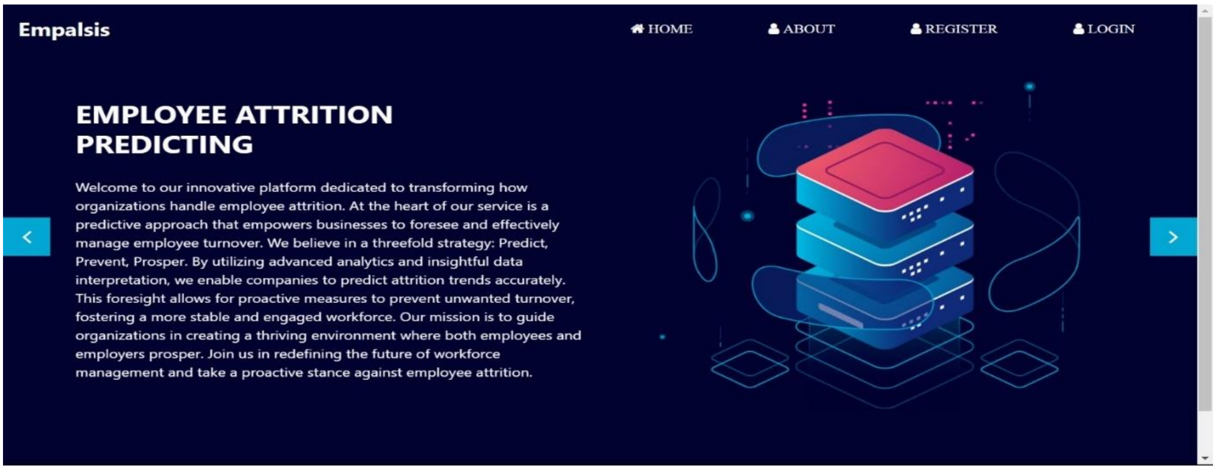
5. **System Performance and Security**
    - The system should ensure data security and confidentiality, especially considering the sensitive nature of employee data.
    - The system should be scalable to handle a large number of users and large datasets.
    - The system should have robust error handling and provide meaningful feedback for troubleshooting.

6. **Testing and Compliance**
    - The system should undergo thorough testing, including functionality, performance, and security testing.

Design Overview

In this section, the design aspects of the application are presented, which seamlessly blend aesthetics with functionality. The accompanying screenshots illustrate the user-friendly interface, highlighting how the process of analyzing employee attrition is effectively simplified. Each design element, from the layout to the color scheme, has been carefully chosen to ensure an intuitive and engaging user experience. These visuals demonstrate the commitment to making complex data analysis both accessible and insightful for the users.

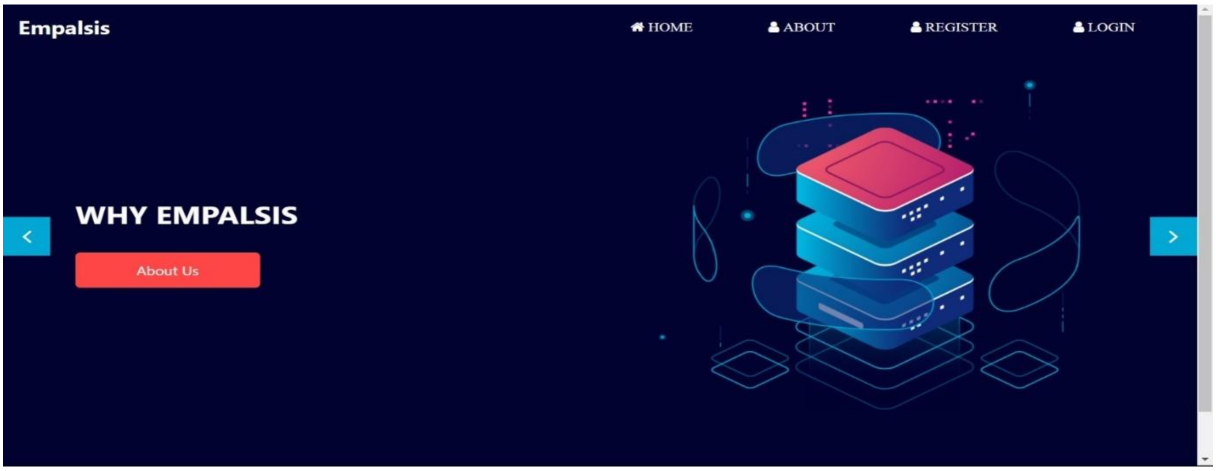
Home Page Design



EMPLOYEE ATTRITION PREDICTING

Welcome to our innovative platform dedicated to transforming how organizations handle employee attrition. At the heart of our service is a predictive approach that empowers businesses to foresee and effectively manage employee turnover. We believe in a threefold strategy: Predict, Prevent, Prosper. By utilizing advanced analytics and insightful data interpretation, we enable companies to predict attrition trends accurately. This foresight allows for proactive measures to prevent unwanted turnover, fostering a more stable and engaged workforce. Our mission is to guide organizations in creating a thriving environment where both employees and employers prosper. Join us in redefining the future of workforce management and take a proactive stance against employee attrition.

Figure 1: Home Page



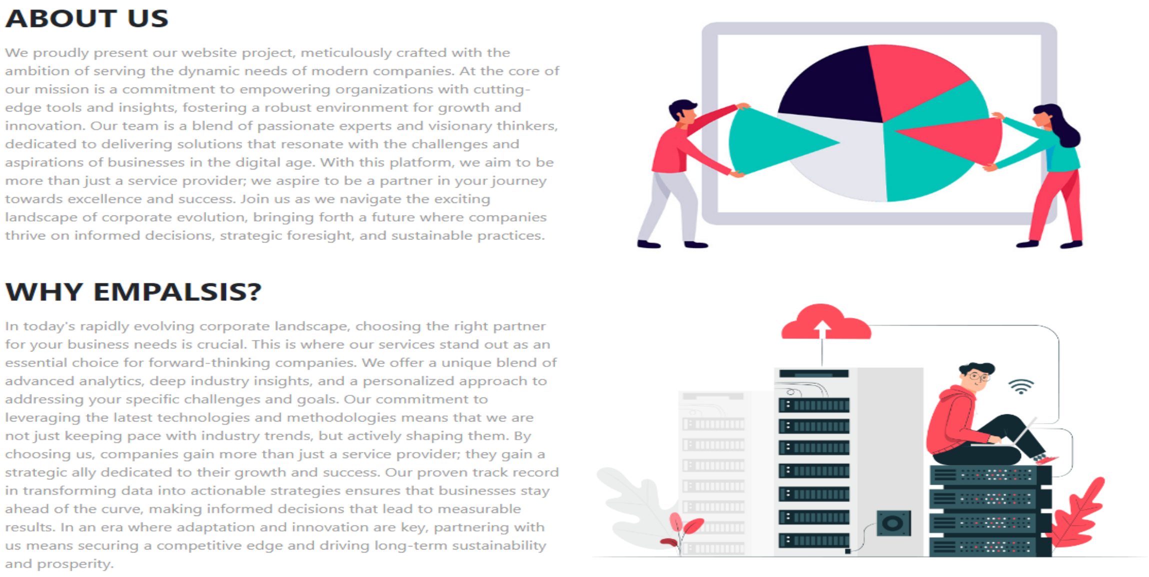
WHY EMPALSIS

Figure 2: Second Slide of the Home Page

The first home page of the application (*Figure 1*) sets the stage for user interaction with a sleek and modern design. The central theme revolves around a dark color palette, which not only emphasizes the tech-centric nature of the platform but also allows for the vibrant colors of the graphics to stand out. The use of abstract and dynamic illustrations signifies the innovative and fluid approach that is taken in handling employee attrition. The bold, capitalized typography of the header 'EMPLOYEE ATTRITION PREDICTING' is deliberate, ensuring immediate attention and delivering the site's purpose with clarity. The navigation is made intuitive through minimalist arrow buttons, guiding the user seamlessly through the informational content.

This second page (*Figure 2*), titled 'WHY EMPALSIS', is designed to communicate the value proposition of the service. It follows a consistent theme with the home page to maintain a coherent user experience. The 'About Us' call-to-action button is prominently placed against the dark background, offering a clear pathway for users seeking more detailed information. The design stays faithful to minimalism, reducing distractions and allowing users to focus on the message.

- **About Page Design**



ABOUT US

We proudly present our website project, meticulously crafted with the ambition of serving the dynamic needs of modern companies. At the core of our mission is a commitment to empowering organizations with cutting-edge tools and insights, fostering a robust environment for growth and innovation. Our team is a blend of passionate experts and visionary thinkers, dedicated to delivering solutions that resonate with the challenges and aspirations of businesses in the digital age. With this platform, we aim to be more than just a service provider; we aspire to be a partner in your journey towards excellence and success. Join us as we navigate the exciting landscape of corporate evolution, bringing forth a future where companies thrive on informed decisions, strategic foresight, and sustainable practices.

WHY EMPALSIS?

In today's rapidly evolving corporate landscape, choosing the right partner for your business needs is crucial. This is where our services stand out as an essential choice for forward-thinking companies. We offer a unique blend of advanced analytics, deep industry insights, and a personalized approach to addressing your specific challenges and goals. Our commitment to leveraging the latest technologies and methodologies means that we are not just keeping pace with industry trends, but actively shaping them. By choosing us, companies gain more than just a service provider; they gain a strategic ally dedicated to their growth and success. Our proven track record in transforming data into actionable strategies ensures that businesses stay ahead of the curve, making informed decisions that lead to measurable results. In an era where adaptation and innovation are key, partnering with us means securing a competitive edge and driving long-term sustainability and prosperity.

Figure 3: About Page

This page (*Figure 3*) is an essential part of any project, as it establishes the ethos of the company and communicates the core values and mission to the audience. The "About" page is designed with clarity and user engagement in mind. It employs a clean and organized layout that facilitates easy reading and comprehension. Using a pie chart graphic as a central visual metaphor cleverly represents the analytical nature of the services, while human figures actively engaging with the chart symbolize the collaborative approach to problem-solving.

- **Register Page Design**

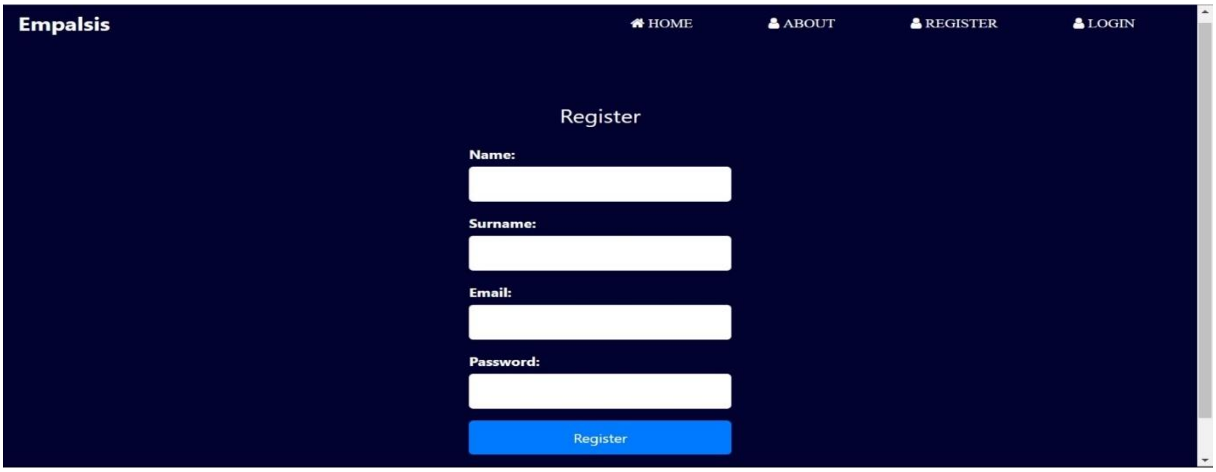


Figure 4: Register Page

The registration page (*Figure 4*) of the application is the gateway for new users, designed with a focus on simplicity and ease of use. The page features a minimalistic layout that centers on a clean form against a deep blue background, reducing visual clutter and directing the user's attention to the task of registration. Fields for the user's name, surname, email, and password are laid out in a logical sequence, promoting a straightforward and undistracted registration process.

- **Login Page Design**

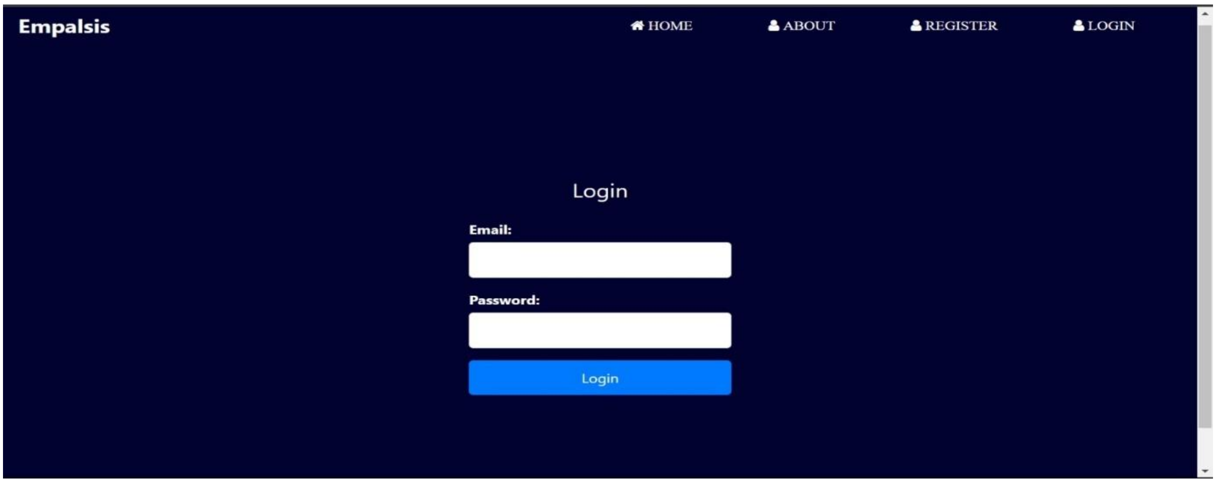
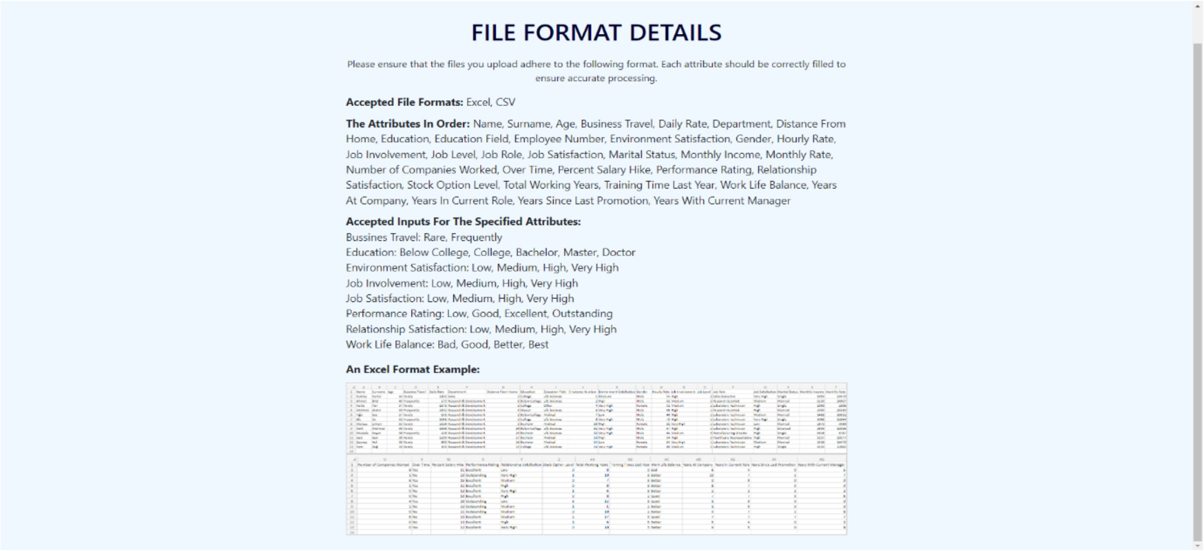


Figure 5: Login Page

The login page (*Figure 5*) maintains the design continuity of the registration page (*Figure 4*), echoing the same color palette and simplicity to provide a familiar experience for returning users. It is crafted to facilitate quick and secure access, featuring only the essential fields required for entry: email and password.

- **Details Page Design**



FILE FORMAT DETAILS

Please ensure that the files you upload adhere to the following format. Each attribute should be correctly filled to ensure accurate processing.

Accepted File Formats: Excel, CSV

The Attributes In Order: Name, Surname, Age, Business Travel, Daily Rate, Department, Distance From Home, Education, Education Field, Employee Number, Environment Satisfaction, Gender, Hourly Rate, Job Involvement, Job Level, Job Role, Job Satisfaction, Marital Status, Monthly Income, Monthly Rate, Number of Companies Worked, Over Time, Percent Salary Hike, Performance Rating, Relationship Satisfaction, Stock Option Level, Total Working Years, Training Time Last Year, Work Life Balance, Years At Company, Years In Current Role, Years Since Last Promotion, Years With Current Manager

Accepted Inputs For The Specified Attributes:
Bussines Travel: Rare, Frequently
Education: Below College, College, Bachelor, Master, Doctor
Environment Satisfaction: Low, Medium, High, Very High
Job Involvement: Low, Medium, High, Very High
Job Satisfaction: Low, Medium, High, Very High
Performance Rating: Low, Good, Excellent, Outstanding
Relationship Satisfaction: Low, Medium, High, Very High
Work Life Balance: Bad, Good, Better, Best

An Excel Format Example:

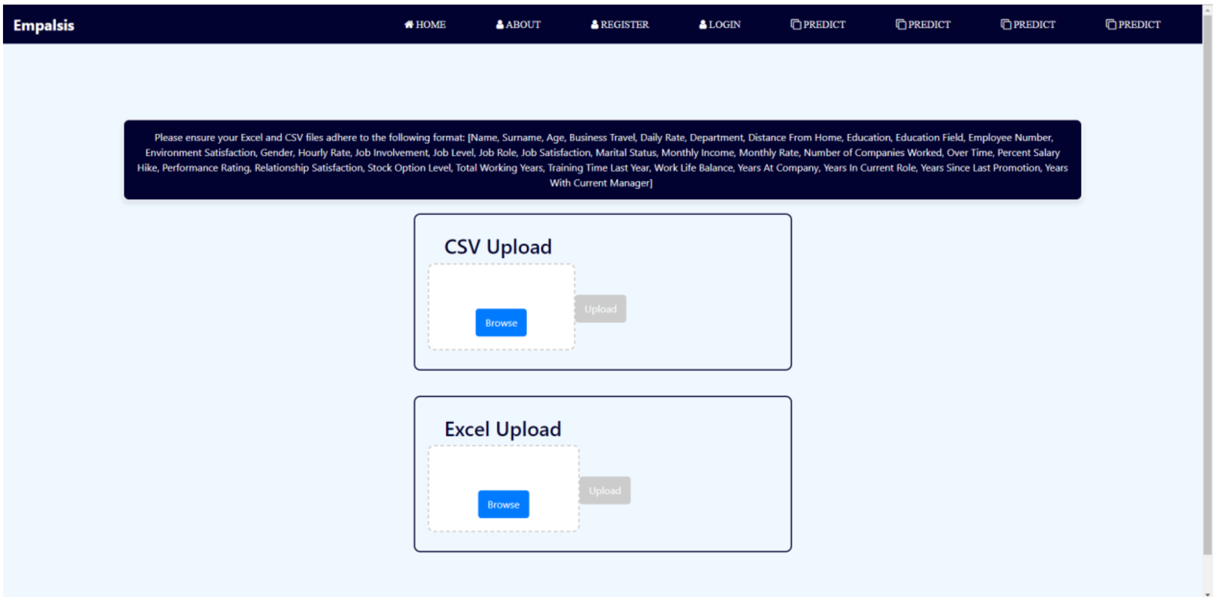
Figure 6: Details Page

This page (*Figure 6*) serves as an informative guide to ensure users upload their data correctly. This page is meticulously designed to be both informational and easy to navigate. It begins with a clear and concise explanation of the accepted file formats and a detailed list of the attributes required for the dataset in the order they should appear. This is crucial in setting users up for success when interfacing with the machine-learning platform.

Directly beneath the textual instructions, users find an illustrative example in the form of a screenshot displaying an Excel spreadsheet. This visual example aligns precisely with the specified attributes, providing a real-world reference that users can emulate. The design of this page aligns with the overall theme of the application, featuring a clean, straightforward layout that prioritizes content comprehension and user guidance.

The inclusion of an actual Excel format snapshot serves as a practical example, reinforcing the textual content and ensuring users have a visual cue to follow. The goal of this page is not only to educate but also to reduce the margin of error during the file upload process, which in turn streamlines the user experience and enhances the efficiency of the data analysis process.

- **File Uploading Page Design**



Please ensure your Excel and CSV files adhere to the following format: [Name, Surname, Age, Business Travel, Daily Rate, Department, Distance From Home, Education, Education Field, Employee Number, Environment Satisfaction, Gender, Hourly Rate, Job Involvement, Job Level, Job Role, Job Satisfaction, Marital Status, Monthly Income, Monthly Rate, Number of Companies Worked, Over Time, Percent Salary Hike, Performance Rating, Relationship Satisfaction, Stock Option Level, Total Working Years, Training Time Last Year, Work Life Balance, Years At Company, Years In Current Role, Years Since Last Promotion, Years With Current Manager]

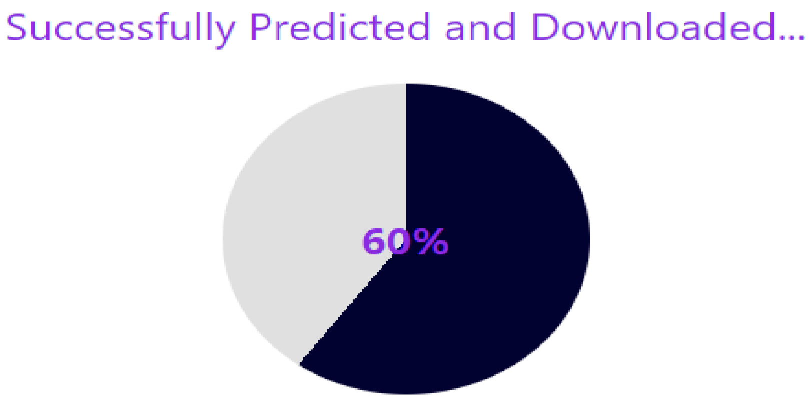
Figure 7: File Uploading Page

This page (*Figure 7*) is a critical component of the application, designed to facilitate the seamless submission of data files in CSV and Excel formats. This page adheres to a minimalist design philosophy, where clarity and ease of use take center stage. The interface is divided into two distinct sections, each dedicated to one of the accepted file types, ensuring users can quickly identify where and how to upload their data.

The page features prominently labeled areas for 'CSV Upload' and 'Excel Upload,' with dotted borders subtly indicating the drag-and-drop functionality, complemented by 'Browse' and 'Upload' buttons. This dual-option layout caters to user preference, accommodating those who favor traditional file selection dialogues as well as those who prefer the convenience of drag-and-drop operations.

Above the upload areas, we provide concise instructions that reiterate the required format for the uploaded files. This ensures users are well-informed of the specifications before initiating the upload process, minimizing the potential for errors.

- **Successfully Downloaded Page Design**



Successfully Predicted and Downloaded...

Figure 8: Download Page with Attrition Percentage

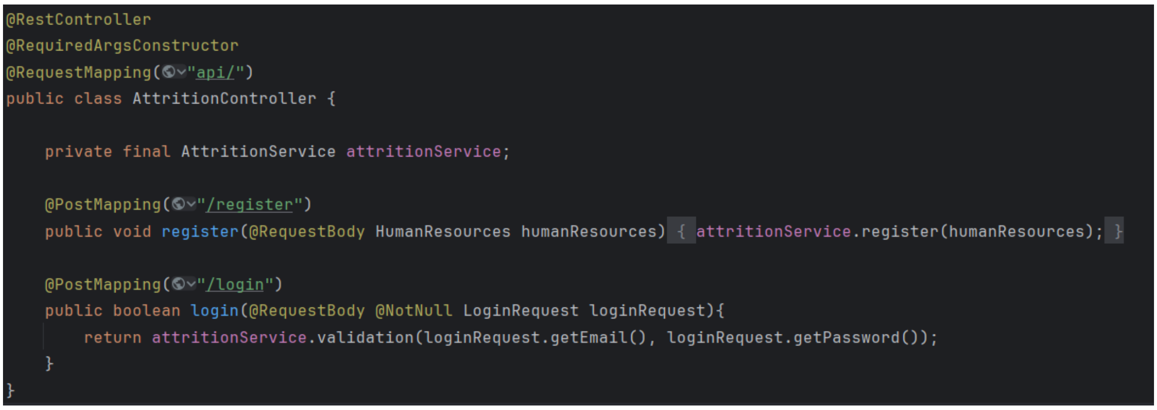
This page (*Figure 8*) is designed to deliver the result of the attrition prediction in a visually impactful and digestible manner. At the center of the page is a pie chart that graphically represents the attrition percentage, providing an at-a-glance understanding of the analysis outcome.

The use of a pie chart here is deliberate; it transcends numerical data, offering an intuitive visual representation of the proportion of employees predicted to leave. The color contrast in the chart is thoughtfully chosen to highlight the significant segment, thus drawing immediate attention to the critical data point – the attrition risk percentage.

Construction of the Project

Java Spring Back-End :

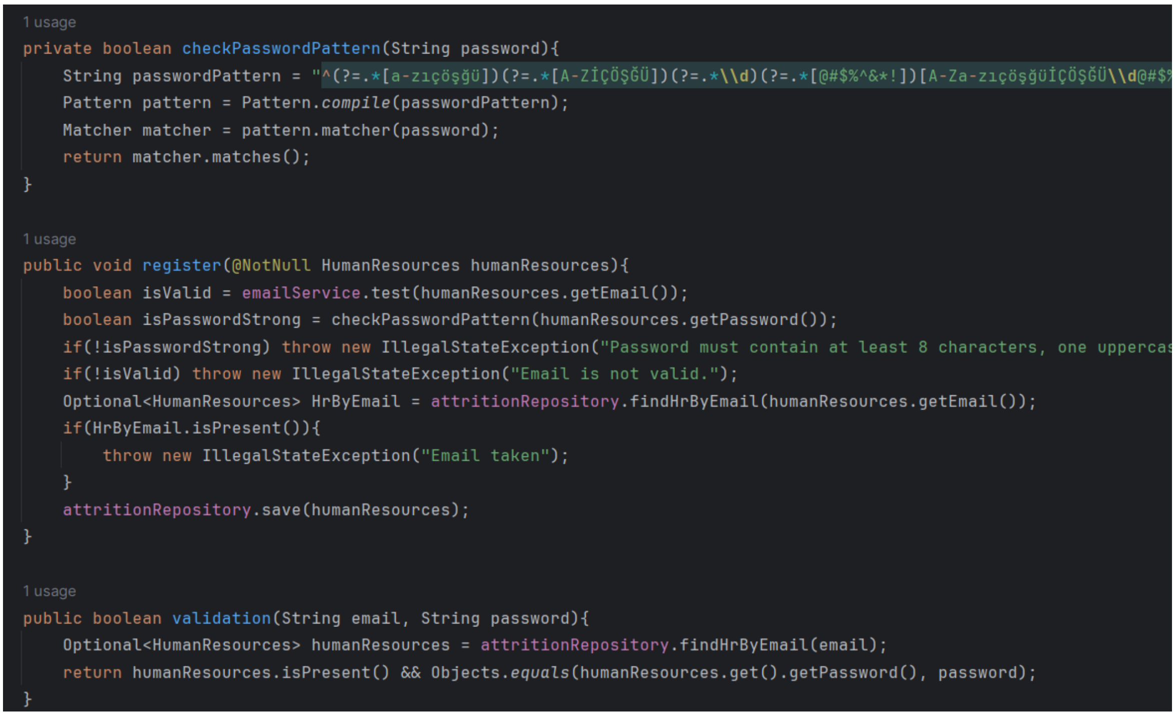
Figure 9: Rest Controller



```java
@RestController
@RequiredArgsConstructor
@RequestMapping("api/")
public class AttritionController {

    private final AttritionService attritionService;

    @PostMapping("/register")
    public void register(@RequestBody HumanResources humanResources) { attritionService.register(humanResources); }

    @PostMapping("/login")
    public boolean login(@RequestBody @NotNull LoginRequest loginRequest){
        return attritionService.validation(loginRequest.getEmail(), loginRequest.getPassword());
    }
}
```

Figure 9 shows the Rest APIs that is used in Java Spring project that front-end calls.

Figure 10: Service

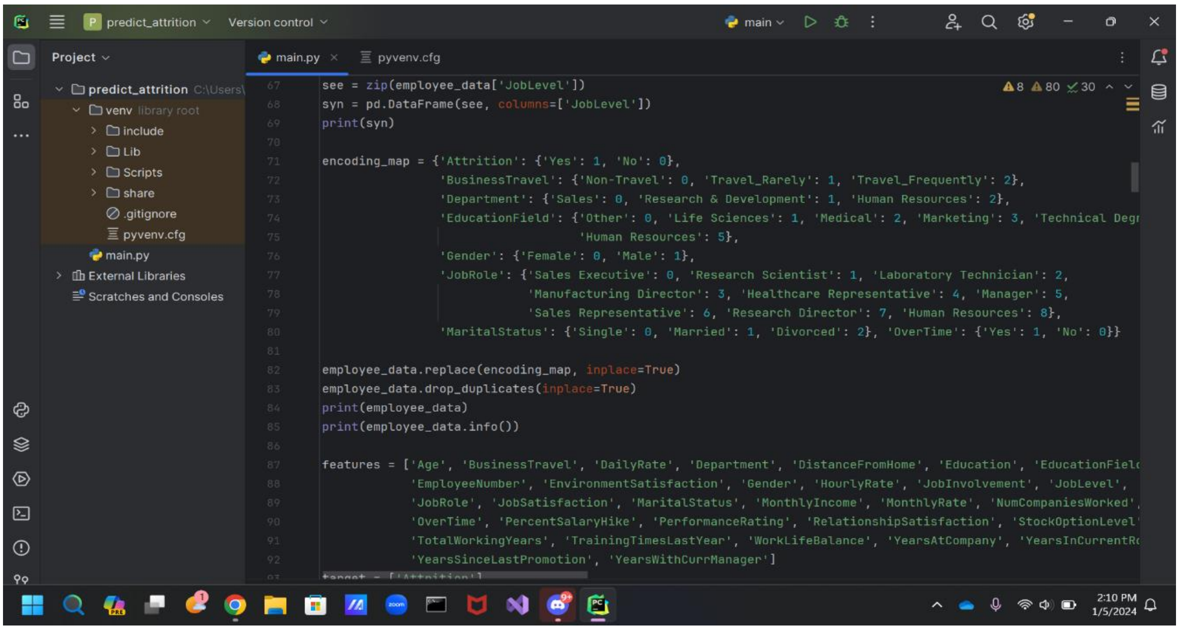


```java
1 usage
private boolean checkPasswordPattern(String password){
    String passwordPattern = "^(?=.*[a-zıçöşğü])(?=.*[A-ZİÇÖŞĞÜ])(?=.*\\d)(?=.*[@#$%^&*!])[A-Za-zıçöşğüİÇÖŞĞÜ\\d@#$
    Pattern pattern = Pattern.compile(passwordPattern);
    Matcher matcher = pattern.matcher(password);
    return matcher.matches();
}

1 usage
public void register(@NotNull HumanResources humanResources){
    boolean isValid = emailService.test(humanResources.getEmail());
    boolean isPasswordStrong = checkPasswordPattern(humanResources.getPassword());
    if(!isPasswordStrong) throw new IllegalStateException("Password must contain at least 8 characters, one uppercas
    if(!isValid) throw new IllegalStateException("Email is not valid.");
    Optional<HumanResources> HrByEmail = attritionRepository.findHrByEmail(humanResources.getEmail());
    if(HrByEmail.isPresent()){
        throw new IllegalStateException("Email taken");
    }
    attritionRepository.save(humanResources);
}

1 usage
public boolean validation(String email, String password){
    Optional<HumanResources> humanResources = attritionRepository.findHrByEmail(email);
    return humanResources.isPresent() && Objects.equals(humanResources.get().getPassword(), password);
}
```

The methods in figure 10 are the actual logic behind the API which it is called as “Services” in MVC design. From the top down, a password pattern is contained so that accounts are secured better, there is a registration service that goes to the database after checking the constraints and then saves. There is also a login service which goes to the database to validate the given inputs.

Python Flask Back-End:

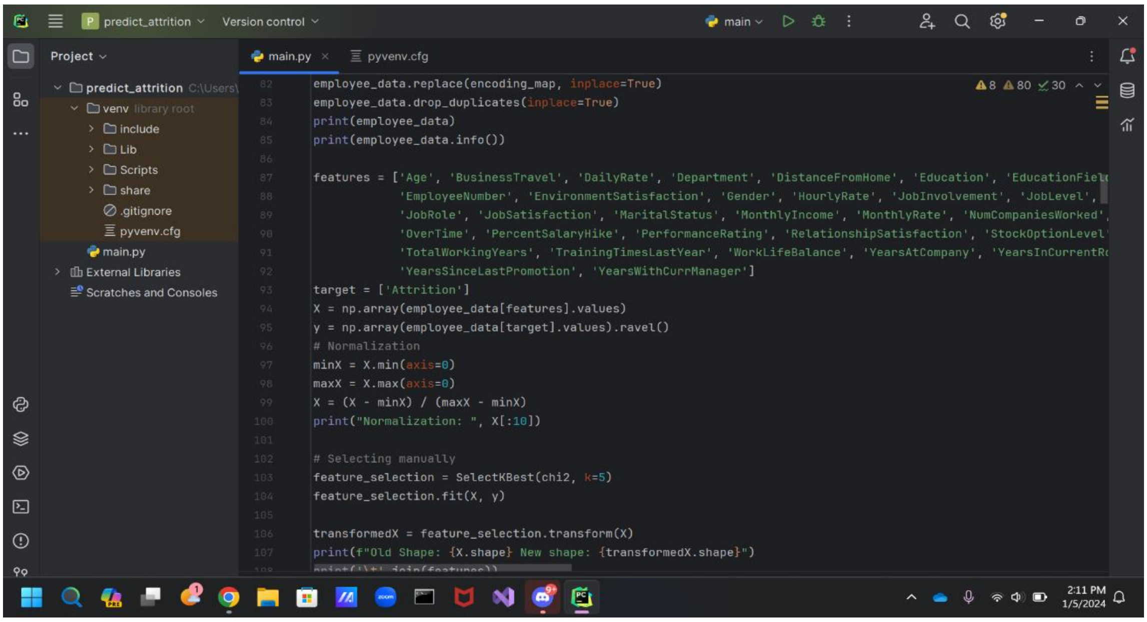
Figure 11: Encoding Map and Features



```
see = zip(employee_data['JobLevel'])
syn = pd.DataFrame(see, columns=['JobLevel'])
print(syn)

encoding_map = {'Attrition': {'Yes': 1, 'No': 0},
                'BusinessTravel': {'Non-Travel': 0, 'Travel_Rarely': 1, 'Travel_Frequently': 2},
                'Department': {'Sales': 0, 'Research & Development': 1, 'Human Resources': 2},
                'EducationField': {'Other': 0, 'Life Sciences': 1, 'Medical': 2, 'Marketing': 3, 'Technical Degr
                                   'Human Resources': 5},
                'Gender': {'Female': 0, 'Male': 1},
                'JobRole': {'Sales Executive': 0, 'Research Scientist': 1, 'Laboratory Technician': 2,
                            'Manufacturing Director': 3, 'Healthcare Representative': 4, 'Manager': 5,
                            'Sales Representative': 6, 'Research Director': 7, 'Human Resources': 8},
                'MaritalStatus': {'Single': 0, 'Married': 1, 'Divorced': 2}, 'OverTime': {'Yes': 1, 'No': 0}}

employee_data.replace(encoding_map, inplace=True)
employee_data.drop_duplicates(inplace=True)
print(employee_data)
print(employee_data.info())

features = ['Age', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome', 'Education', 'EducationFiel
            'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement', 'JobLevel',
            'JobRole', 'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
            'OverTime', 'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StockOptionLevel'
            'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRo
            'YearsSinceLastPromotion', 'YearsWithCurrManager']
target = ['Attrition']
```

Figure 11 shows that the values of the features in the dataset are encoded to numeric values to be able to train the model correctly.

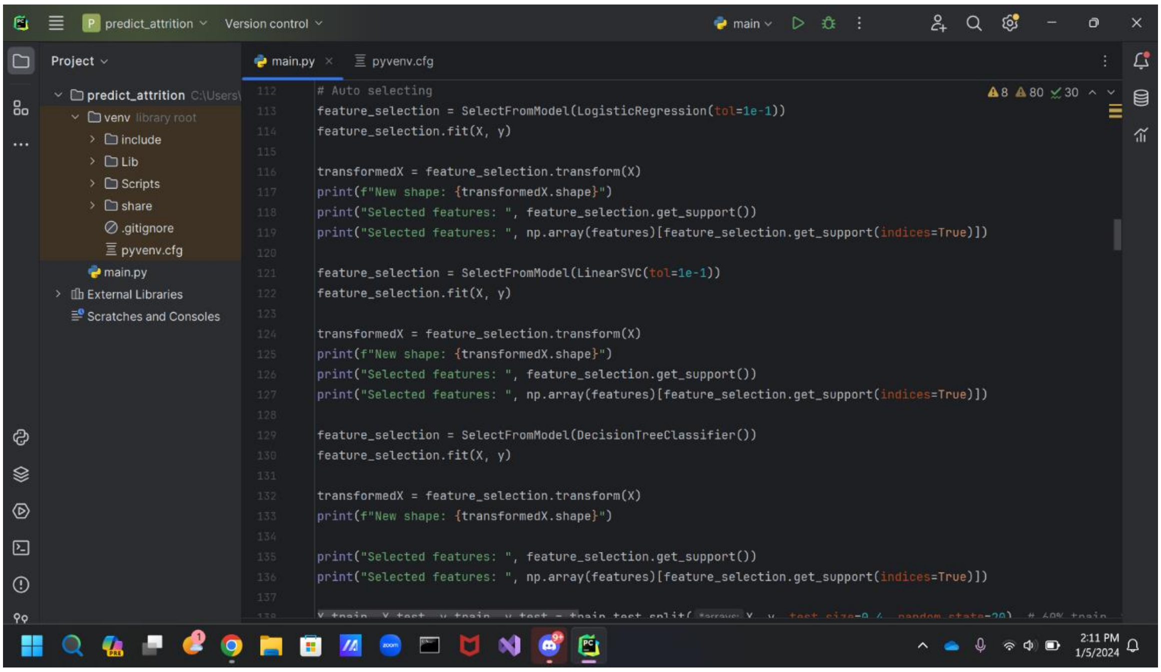
Figure 12: Normalization



```
employee_data.replace(encoding_map, inplace=True)
employee_data.drop_duplicates(inplace=True)
print(employee_data)
print(employee_data.info())

features = ['Age', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome', 'Education', 'EducationFiel
            'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement', 'JobLevel',
            'JobRole', 'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
            'OverTime', 'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StockOptionLevel'
            'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRo
            'YearsSinceLastPromotion', 'YearsWithCurrManager']
target = ['Attrition']
X = np.array(employee_data[features].values)
y = np.array(employee_data[target].values).ravel()
# Normalization
minX = X.min(axis=0)
maxX = X.max(axis=0)
X = (X - minX) / (maxX - minX)
print("Normalization: ", X[:10])

# Selecting manually
feature_selection = SelectKBest(chi2, k=5)
feature_selection.fit(X, y)

transformedX = feature_selection.transform(X)
print(f"Old Shape: {X.shape} New shape: {transformedX.shape}")
print('\t'.join(features))
```

Figure 12 shows the normalization of the dataset. Normalization is the process of scaling input features within a similar range (usually between 0 and 1). The goal of the normalization is to bring all the features to a similar scale to prevent certain features from dominating and having undue influence on the model's learning process due to differences in their magnitude.

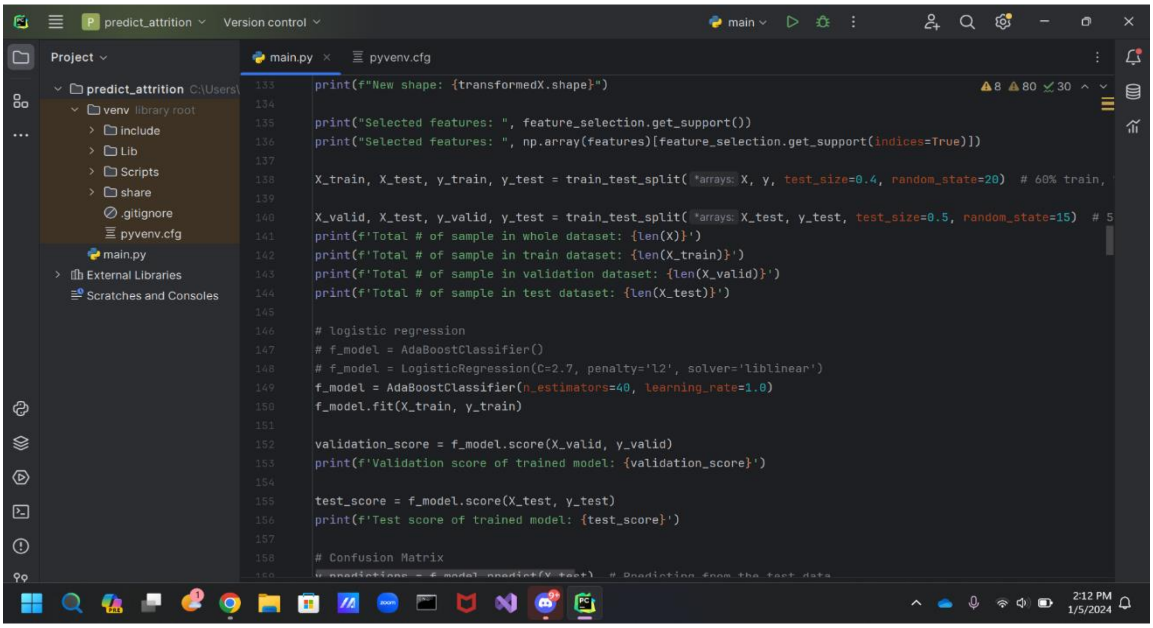
Figure 13: Feature Selection



```python
# Auto selecting
feature_selection = SelectFromModel(LogisticRegression(tol=1e-1))
feature_selection.fit(X, y)

transformedX = feature_selection.transform(X)
print(f"New shape: {transformedX.shape}")
print("Selected features: ", feature_selection.get_support())
print("Selected features: ", np.array(features)[feature_selection.get_support(indices=True)])

feature_selection = SelectFromModel(LinearSVC(tol=1e-1))
feature_selection.fit(X, y)

transformedX = feature_selection.transform(X)
print(f"New shape: {transformedX.shape}")
print("Selected features: ", feature_selection.get_support())
print("Selected features: ", np.array(features)[feature_selection.get_support(indices=True)])

feature_selection = SelectFromModel(DecisionTreeClassifier())
feature_selection.fit(X, y)

transformedX = feature_selection.transform(X)
print(f"New shape: {transformedX.shape}")

print("Selected features: ", feature_selection.get_support())
print("Selected features: ", np.array(features)[feature_selection.get_support(indices=True)])
```

Figure 13 shows the feature selection which is the process of choosing a subset of relevant and important features from the original dataset. Three algorithms are involved: Logistic Regression, Linear SVC and Decision Tree Classifier.

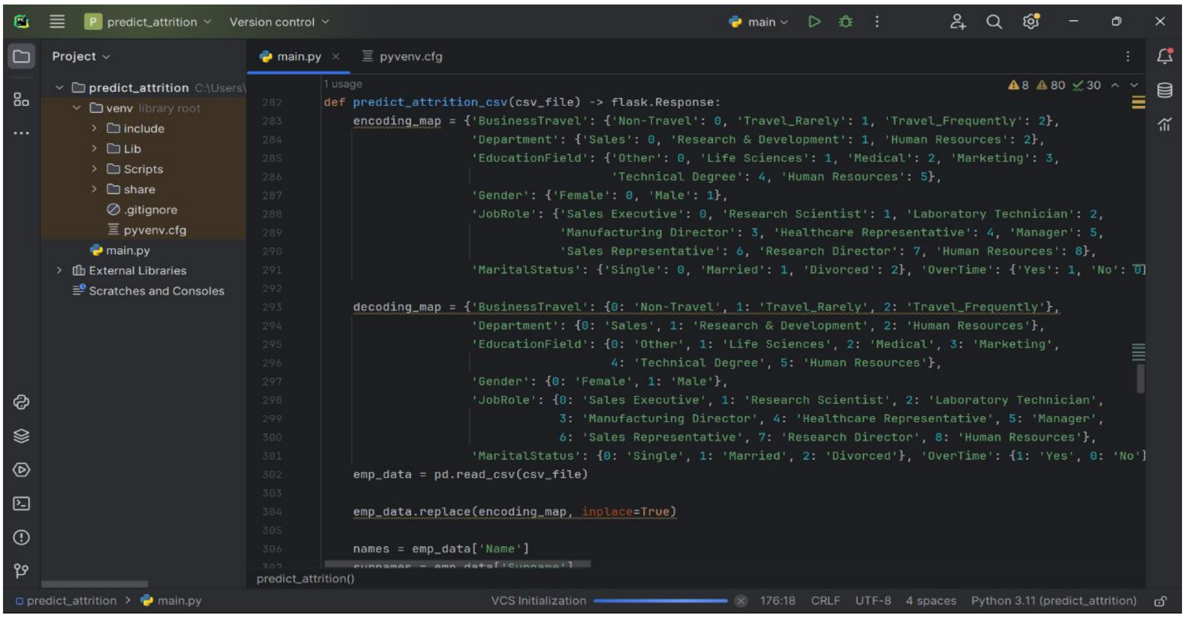
Figure 14: Train, Test, Validate and the Machine Learning Model



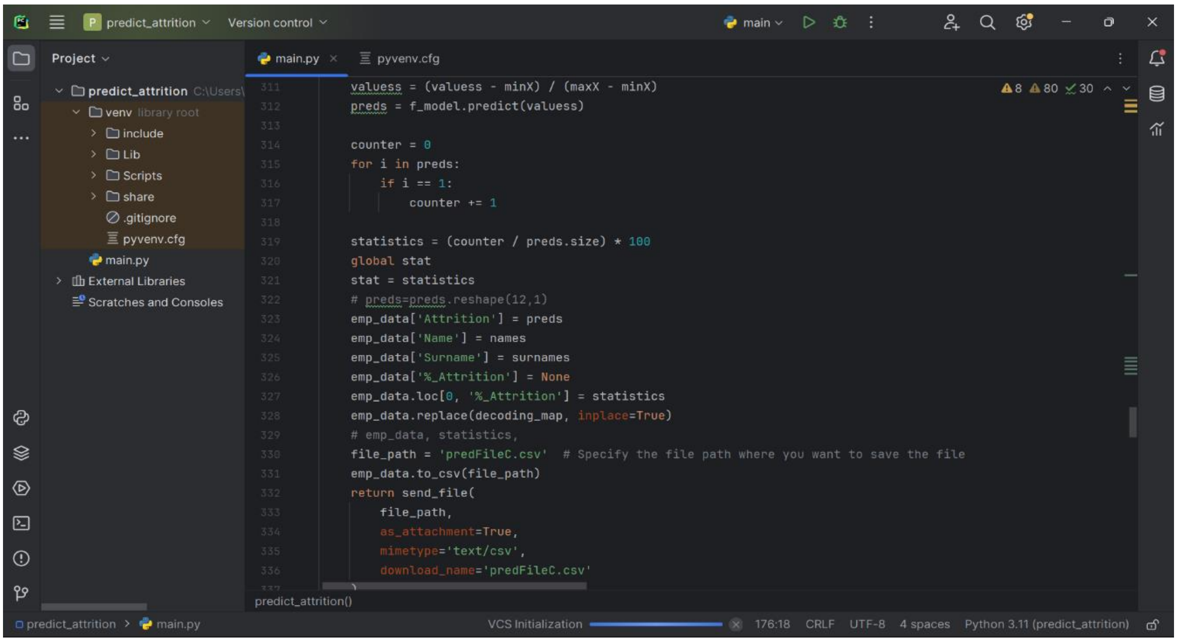
```python
print(f"New shape: {transformedX.shape}")

print("Selected features: ", feature_selection.get_support())
print("Selected features: ", np.array(features)[feature_selection.get_support(indices=True)])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=20)  # 60% train,

X_valid, X_test, y_valid, y_test = train_test_split(X_test, y_test, test_size=0.5, random_state=15)  # 5
print(f'Total # of sample in whole dataset: {len(X)}')
print(f'Total # of sample in train dataset: {len(X_train)}')
print(f'Total # of sample in validation dataset: {len(X_valid)}')
print(f'Total # of sample in test dataset: {len(X_test)}')

# logistic regression
# f_model = AdaBoostClassifier()
# f_model = LogisticRegression(C=2.7, penalty='l2', solver='liblinear')
f_model = AdaBoostClassifier(n_estimators=40, learning_rate=1.0)
f_model.fit(X_train, y_train)

validation_score = f_model.score(X_valid, y_valid)
print(f'Validation score of trained model: {validation_score}')

test_score = f_model.score(X_test, y_test)
print(f'Test score of trained model: {test_score}')

# Confusion Matrix
```

In figure 14, after the feature selection, the dataset is splited into 3 parts: testing data, training data and validation data. Different machine learning algorithms are tested, optimized and finally the optimal model is found which is the AdaBoost Classifier. Hyperparameters of the model are optimized. The training data is fitted, results and scores are taken.

Figure 15-16: Predicting CSV File Method



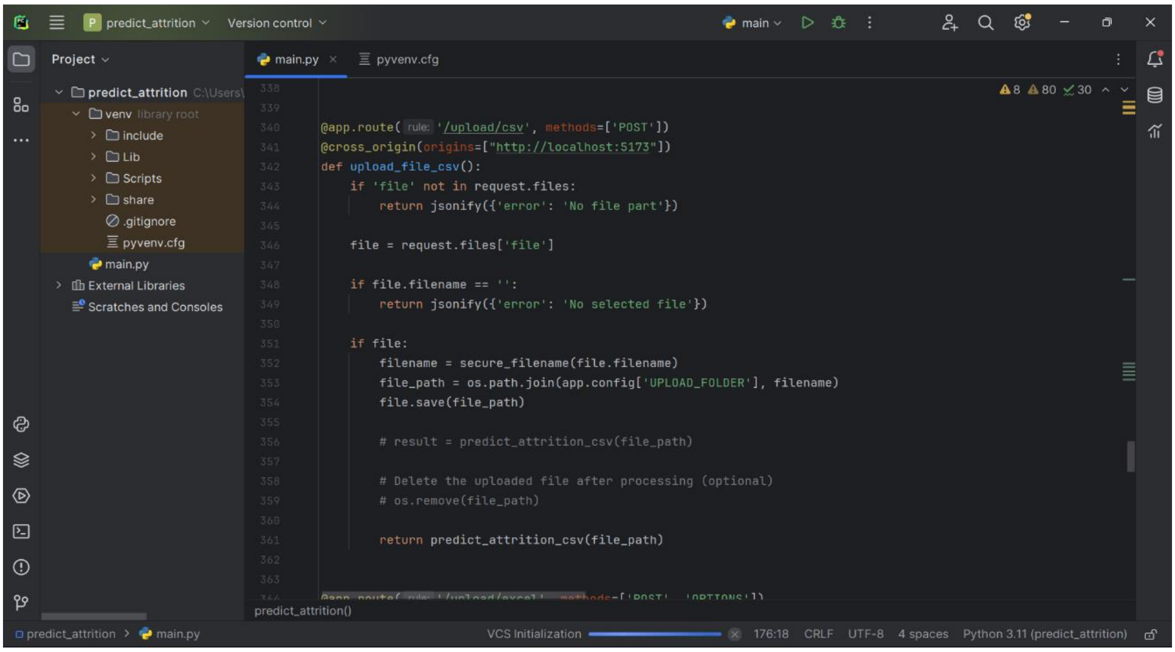
```python
def predict_attrition_csv(csv_file) -> flask.Response:
    encoding_map = {'BusinessTravel': {'Non-Travel': 0, 'Travel_Rarely': 1, 'Travel_Frequently': 2},
                    'Department': {'Sales': 0, 'Research & Development': 1, 'Human Resources': 2},
                    'EducationField': {'Other': 0, 'Life Sciences': 1, 'Medical': 2, 'Marketing': 3,
                                       'Technical Degree': 4, 'Human Resources': 5},
                    'Gender': {'Female': 0, 'Male': 1},
                    'JobRole': {'Sales Executive': 0, 'Research Scientist': 1, 'Laboratory Technician': 2,
                                'Manufacturing Director': 3, 'Healthcare Representative': 4, 'Manager': 5,
                                'Sales Representative': 6, 'Research Director': 7, 'Human Resources': 8},
                    'MaritalStatus': {'Single': 0, 'Married': 1, 'Divorced': 2}, 'OverTime': {'Yes': 1, 'No': 0}

    decoding_map = {'BusinessTravel': {0: 'Non-Travel', 1: 'Travel_Rarely', 2: 'Travel_Frequently'},
                    'Department': {0: 'Sales', 1: 'Research & Development', 2: 'Human Resources'},
                    'EducationField': {0: 'Other', 1: 'Life Sciences', 2: 'Medical', 3: 'Marketing',
                                       4: 'Technical Degree', 5: 'Human Resources'},
                    'Gender': {0: 'Female', 1: 'Male'},
                    'JobRole': {0: 'Sales Executive', 1: 'Research Scientist', 2: 'Laboratory Technician',
                                3: 'Manufacturing Director', 4: 'Healthcare Representative', 5: 'Manager',
                                6: 'Sales Representative', 7: 'Research Director', 8: 'Human Resources'},
                    'MaritalStatus': {0: 'Single', 1: 'Married', 2: 'Divorced'}, 'OverTime': {1: 'Yes', 0: 'No'}
    emp_data = pd.read_csv(csv_file)

    emp_data.replace(encoding_map, inplace=True)

    names = emp_data['Name']
    surnames = emp_data['Surname']
```



```python
    valuess = (valuess - minX) / (maxX - minX)
    preds = f_model.predict(valuess)

    counter = 0
    for i in preds:
        if i == 1:
            counter += 1

    statistics = (counter / preds.size) * 100
    global stat
    stat = statistics
    # preds=preds.reshape(12,1)
    emp_data['Attrition'] = preds
    emp_data['Name'] = names
    emp_data['Surname'] = surnames
    emp_data['%_Attrition'] = None
    emp_data.loc[0, '%_Attrition'] = statistics
    emp_data.replace(decoding_map, inplace=True)
    # emp_data, statistics,
    file_path = 'predFileC.csv'  # Specify the file path where you want to save the file
    emp_data.to_csv(file_path)
    return send_file(
        file_path,
        as_attachment=True,
        mimetype='text/csv',
        download_name='predFileC.csv'
    )
```

Figures 15 and 16 shows the CSV prediction method. The input file, which is a CSV file, is encoded to apply the machine learning algorithm and decoded to return back the predicted file in understandable format. Employee attrition percentage is also taken and returned.

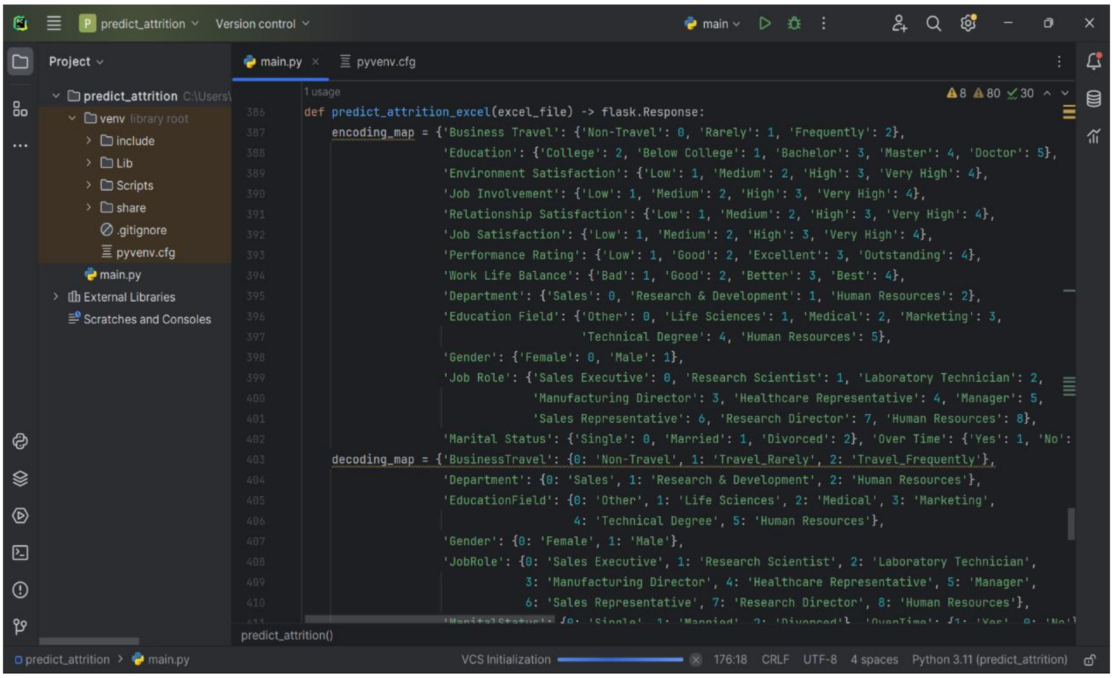
Figure 17: the API of the Predicting CSV File Method



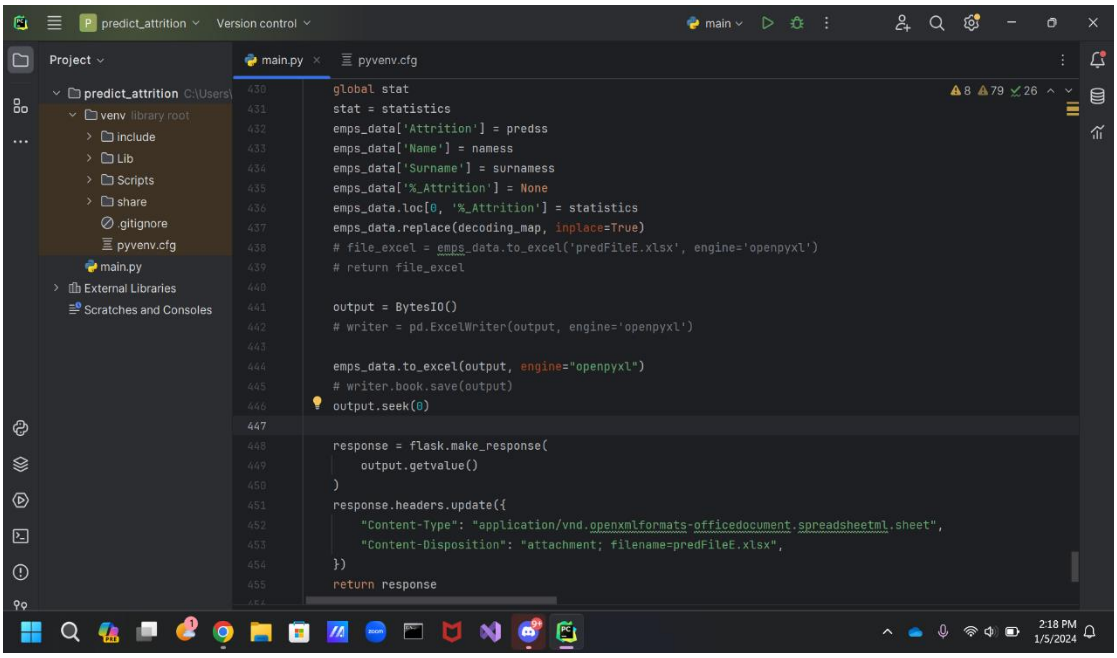
```python

@app.route('/upload/csv', methods=['POST'])
@cross_origin(origins=["http://localhost:5173"])
def upload_file_csv():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})

    file = request.files['file']

    if file.filename == '':
        return jsonify({'error': 'No selected file'})

    if file:
        filename = secure_filename(file.filename)
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(file_path)

        # result = predict_attrition_csv(file_path)

        # Delete the uploaded file after processing (optional)
        # os.remove(file_path)

        return predict_attrition_csv(file_path)


@app.route('/upload/excel', methods=['POST', 'OPTIONS'])
```

Figure 17 shows the API of the predicting CSV file method which is mentioned in figures 15 and 16. Front-end calls with this API.

Figure 18-19: Predicting Excel File Method



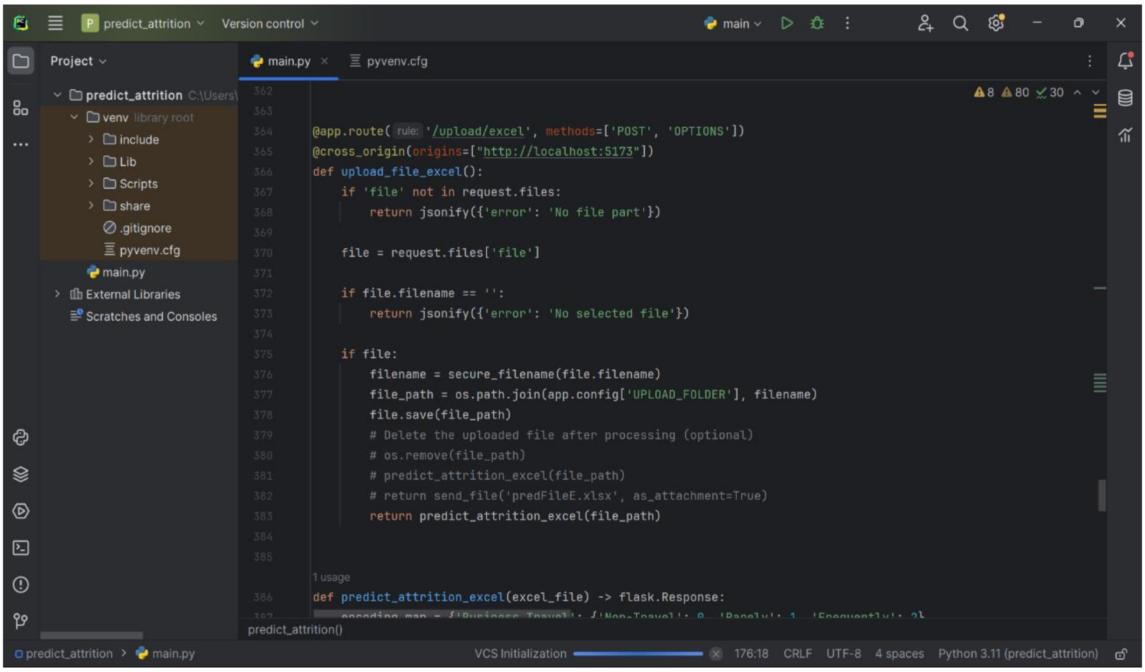
```python
def predict_attrition_excel(excel_file) -> flask.Response:
    encoding_map = {'Business Travel': {'Non-Travel': 0, 'Rarely': 1, 'Frequently': 2},
                    'Education': {'College': 2, 'Below College': 1, 'Bachelor': 3, 'Master': 4, 'Doctor': 5},
                    'Environment Satisfaction': {'Low': 1, 'Medium': 2, 'High': 3, 'Very High': 4},
                    'Job Involvement': {'Low': 1, 'Medium': 2, 'High': 3, 'Very High': 4},
                    'Relationship Satisfaction': {'Low': 1, 'Medium': 2, 'High': 3, 'Very High': 4},
                    'Job Satisfaction': {'Low': 1, 'Medium': 2, 'High': 3, 'Very High': 4},
                    'Performance Rating': {'Low': 1, 'Good': 2, 'Excellent': 3, 'Outstanding': 4},
                    'Work Life Balance': {'Bad': 1, 'Good': 2, 'Better': 3, 'Best': 4},
                    'Department': {'Sales': 0, 'Research & Development': 1, 'Human Resources': 2},
                    'Education Field': {'Other': 0, 'Life Sciences': 1, 'Medical': 2, 'Marketing': 3,
                                        'Technical Degree': 4, 'Human Resources': 5},
                    'Gender': {'Female': 0, 'Male': 1},
                    'Job Role': {'Sales Executive': 0, 'Research Scientist': 1, 'Laboratory Technician': 2,
                                 'Manufacturing Director': 3, 'Healthcare Representative': 4, 'Manager': 5,
                                 'Sales Representative': 6, 'Research Director': 7, 'Human Resources': 8},
                    'Marital Status': {'Single': 0, 'Married': 1, 'Divorced': 2}, 'Over Time': {'Yes': 1, 'No':
    decoding_map = {'BusinessTravel': {0: 'Non-Travel', 1: 'Travel_Rarely', 2: 'Travel_Frequently'},
                    'Department': {0: 'Sales', 1: 'Research & Development', 2: 'Human Resources'},
                    'EducationField': {0: 'Other', 1: 'Life Sciences', 2: 'Medical', 3: 'Marketing',
                                       4: 'Technical Degree', 5: 'Human Resources'},
                    'Gender': {0: 'Female', 1: 'Male'},
                    'JobRole': {0: 'Sales Executive', 1: 'Research Scientist', 2: 'Laboratory Technician',
                                3: 'Manufacturing Director', 4: 'Healthcare Representative', 5: 'Manager',
                                6: 'Sales Representative', 7: 'Research Director', 8: 'Human Resources'},
                    'MaritalStatus': {0: 'Single', 1: 'Married', 2: 'Divorced'}, 'OverTime': {1: 'Yes', 0: 'No'}
```



```python
    global stat
    stat = statistics
    emps_data['Attrition'] = predss
    emps_data['Name'] = namess
    emps_data['Surname'] = surnamess
    emps_data['%_Attrition'] = None
    emps_data.loc[0, '%_Attrition'] = statistics
    emps_data.replace(decoding_map, inplace=True)
    # file_excel = emps_data.to_excel('predFileE.xlsx', engine='openpyxl')
    # return file_excel

    output = BytesIO()
    # writer = pd.ExcelWriter(output, engine='openpyxl')

    emps_data.to_excel(output, engine="openpyxl")
    # writer.book.save(output)
    output.seek(0)

    response = flask.make_response(
        output.getvalue()
    )
    response.headers.update({
        "Content-Type": "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet",
        "Content-Disposition": "attachment; filename=predFileE.xlsx",
    })
    return response
```

Figures 18 and 19 shows the Excel prediction method. The input file, which is an Excel file, is encoded to apply the machine learning algorithm and decoded to return back the predicted file in understandable format. Employee attrition percentage is also taken and returned.

Figure 20: the API of the Predicting Excel File Method



```python
@app.route('/upload/excel', methods=['POST', 'OPTIONS'])
@cross_origin(origins=["http://localhost:5173"])
def upload_file_excel():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})

    file = request.files['file']

    if file.filename == '':
        return jsonify({'error': 'No selected file'})

    if file:
        filename = secure_filename(file.filename)
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(file_path)
        # Delete the uploaded file after processing (optional)
        # os.remove(file_path)
        # predict_attrition_excel(file_path)
        # return send_file('predFileE.xlsx', as_attachment=True)
        return predict_attrition_excel(file_path)


def predict_attrition_excel(excel_file) -> flask.Response:
```

Figure 20 shows the API of the predicting Excel file method which is mentioned in figures 18 and 19. Front-end calls with this API.

Flow Chart Diagram

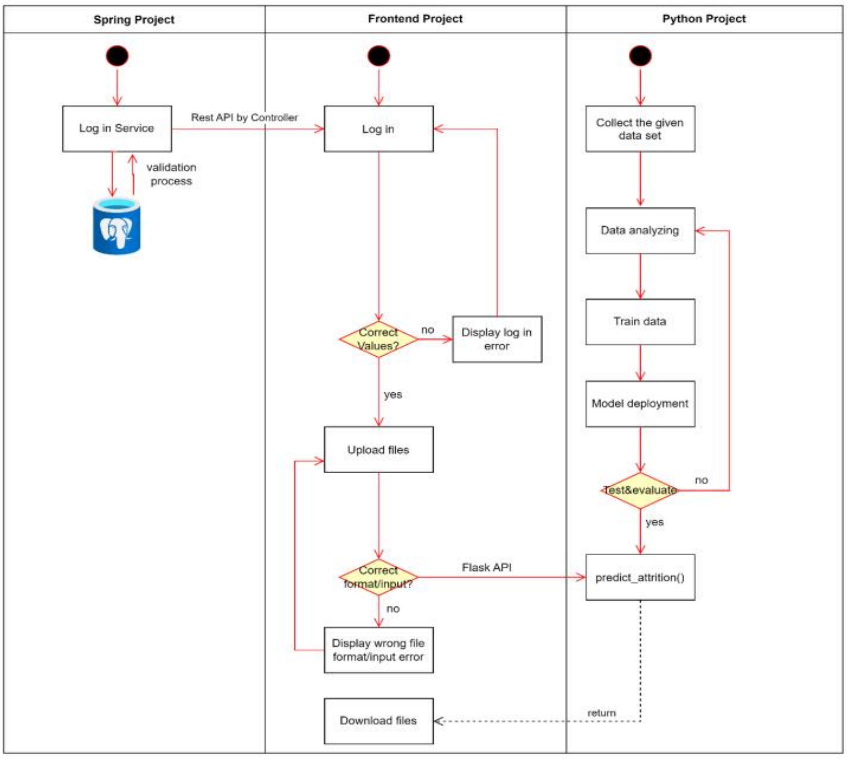


Figure 21

The diagram above (Figure 21) illustrates the sequence of steps, activities and processes in the application.

Use Case Diagram

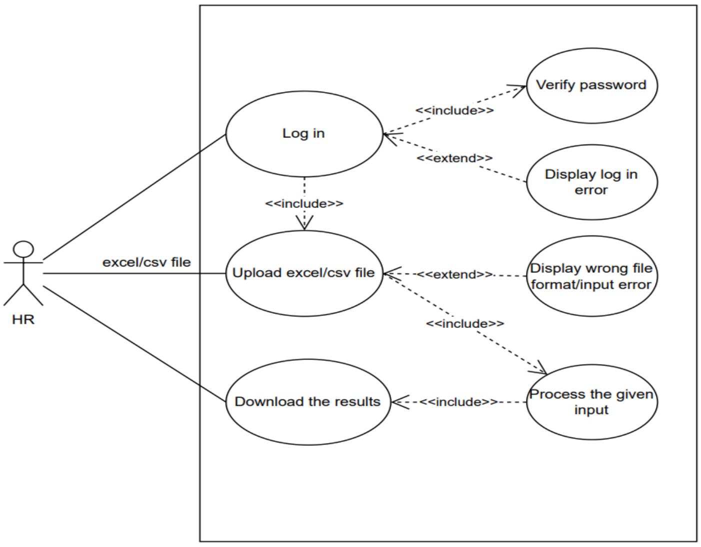


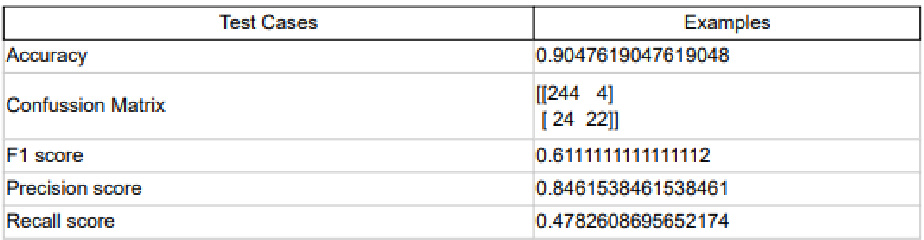
Figure 22

The diagram above (Figure 22) illustrates the interactions between the users and the application.

Encountered Challenges

The biggest problem during the development cycle was the "File corrupted" error response while opening the output Excel file. The process for downloading CSV format worked successfully while the Excel file didn't, the system gave the mentioned error. The reason emerged in the front-end project. The "axios" method has been replaced with the "fetch" method and the error cleared away (solely because of the structure of the methods). It turned out that it was an encoding problem. Because the CSV file type is text, there is no problem with encoding it. But the "axios" method accepts the Excel file as text too when it isn't. The "axios" method encodes as UTF-8 when it should be ANSI. But the "fetch" method takes it as binary so there is no problem with it.

Test and Results



Test Cases	Examples
Accuracy	0.9047619047619048
Confussion Matrix	[[244 4] [24 22]]
F1 score	0.6111111111111112
Precision score	0.8461538461538461
Recall score	0.4782608695652174

Figure 23

The table above shows the test results of the machine learning model. The accuracy is %90. In the confusion matrix, there is a total of 294 samples. 266 samples are successfully predicted and 28 samples are predicted false. F1 score is %61 which is considered to be intermediate reliability. The precision score is %85 and the recall score is %48. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall score is the ratio of correctly predicted positive observations to actual positive observations. The F1 score is the harmonic mean of precision and recall.

Future Works

Two new pages and functionalities will be added to the application. The first one will keep records of previous predictions and display a graph that shows statistics on employee attrition levels monthly. With this new functionality, human resources will be able to see past attritions and improvements or deteriorations. The second one will keep the predicted datasets on the application and display it on the page. Records of these datasets will also be kept.

References

- **https://www.linkedin.com/pulse/employee-attrition-prediction-project-analysis-research-noor-saeed/**
- **https://www.analyticsvidhya.com/blog/2021/11/employee-attrition-prediction-a-comprehensive-guide/**
- Dataset: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset