Part 1: Data Simulation and Preprocessing

Task 1: Data Simulation (InternshipSuccessDataSimulator class simulator process)

The InternshipSuccessDataSimulator class simulator process simulates a dataset of internship success data, with a specified number of interns and a specified proportion of interns who are successful. The data includes features such as intern ID, gender, age, company name, intern role, internship duration, academic GPA, number of internships completed, number of certifications, intern technical skills, programming level, technical skills required, soft skills, technical interview score, education, and department.

How the simulator works:

1. The simulator starts by generating a list of intern IDs.
2. For each intern ID, the simulator generates a random gender, age, company name, intern role, internship duration, academic GPA, number of internships completed, number of certifications, intern technical skills, programming level, technical skills required, soft skills, technical interview score, education, and department.
3. The simulator then sets the internship success status for each intern based on the specified success proportion. Interns with a random value below the success proportion are marked as successful.
4. The simulator then returns the simulated dataset as a Pandas DataFrame.

Task 2: Data Cleaning and Preprocessing

Handling Missing Values: The dataset does not contain any missing value

Consistency in Data Types: Ensuring consistency in data types was a fundamental aspect of the data cleaning process. All features were thoroughly inspected to guarantee uniform data types across the dataset. This included converting variables to their appropriate data types, such as transforming categorical variables into the categorical data type and ensuring numerical variables were in the correct numeric format.

Encoding Categorical Variables: Categorical variables were appropriately encoded to facilitate their integration into the predictive models. The categorical features were transformed using one-hot encoding, which converted categorical variables into a binary format. This process ensured that each categorical variable was transformed into a series of binary variables, maintaining the integrity of the categorical information within the dataset while enabling seamless integration into the machine learning algorithms.

Task 3: Model Selection

Dataset Characteristics

The simulated dataset created for predicting internship success comprises various features related to applicants' academic performance, relevant skills, and additional factors potentially influencing the outcome. These features can be a mix of both numerical and categorical data. The dataset, which represents a binary classification task, includes a target variable indicating the success (1) or failure (0) of internship applications.

Model 1: Logistic Regression

Logistic regression has been selected due to its interpretability and simplicity. The model's transparency in illustrating the relationship between individual features and the likelihood of internship success is vital for understanding the factors influencing the outcome. It serves as an efficient baseline model, well-suited for scenarios where the relationship between predictors and the target outcome might exhibit linear patterns.

Model 2: Random Forest Classifier

Random forest classifier has been incorporated to address potential complexities within the dataset. This model excels in capturing nonlinear and intricate relationships between features and the predicted outcome. By leveraging ensemble learning, it combines multiple decision trees to mitigate overfitting and enhance generalization. This makes it a suitable candidate for scenarios where the relationships between predictors and the outcome are complex and nonlinear.

Task 5: Hyperparameter Optimization

Task 6: Model Deployment

The trained machine learning model, after its development and testing phase, was successfully deployed as an API using the Render platform through integration with a GitHub repository. The model was serialized and saved to maintain its structure and parameters, ensuring its seamless integration into the API service. Leveraging Streamlit, a Python web framework, an endpoint was created to accept input data and generate predictions based on the trained model. The GitHub repository served as the source for deployment, hosting the codebase inclusive of the model and the API service. Through the linkage between the GitHub repository and the Render platform, the deployment process was streamlined, allowing automatic fetching of the code and subsequent deployment of the API. This deployment provides an easily accessible endpoint, enabling users to submit input data, receive predictions in real time, and interact with the trained machine learning model over the web, ensuring its practical use and accessibility for predictive tasks.

Application link: https://internship-success.onrender.com/
Github link: https://github.com/Sodiq179/Machine_Learning_Test_CLIMDES

Task 7: Explainability

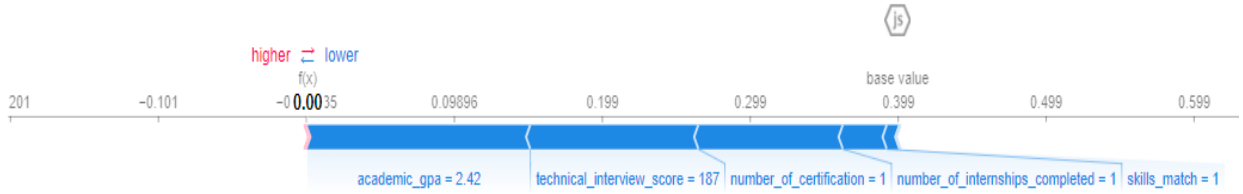

Figure 1: *Internship application failed*

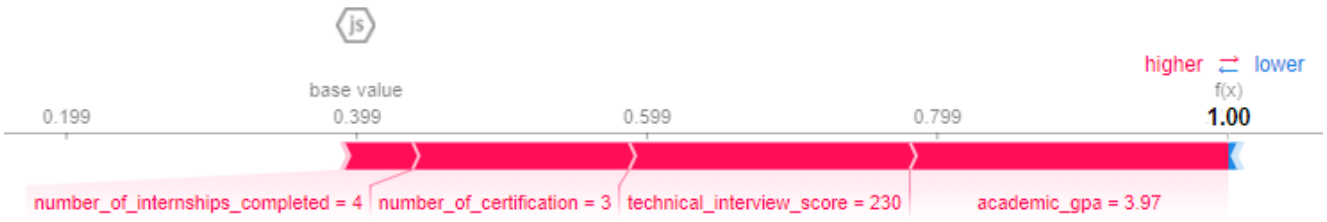

Figure 2: *Internship application successful*

Interpretation:

In the visualizations above, feature values highlighted in pink contribute to an increase in the prediction, while those in blue lead to a decrease. The bar's size indicates the magnitude of each feature's effect on the prediction. Cumulatively, the sum of all feature SHAP values clarifies why the model's prediction deviates from the baseline.

Figure 1 shows an intern with the following details:

```
{'age': 24,
 'internship_duration': 6},
 'academic_gpa': 2.42,
 'number_of_internships_completed': 1,
 'number_of_certification': 1,
 'programming_level': "Intermediate",
 'technical_interview_score': 187…}
```

In this specific instance, the model predicted a value of 0, indicating an unsuccessful internship application. The most influential factor affecting this prediction is the intern's academic GPA, which significantly diminishes the likelihood of a successful application. Additionally, the intern's technical score of 187 also notably reduces the chances of securing the internship.

Figure 2: shows an intern with the following details:

```
{'age': 25,
 'internship_duration': 6},
 'academic_gpa': 3.97,
 'number_of_internships_completed': 4,
 'number_of_certification': 3,
 'programming_level': 2,
```

```
'technical_interview_score': 230…}
```

The model predicted a value of 1, indicating a successful internship application. The most influential factors contributing to this positive prediction are the intern's high academic GPA (3.97) and impressive technical interview score (230), significantly enhancing their prospects of securing the internship. Furthermore, having completed four previous internships and possessing three certifications also notably increased the likelihood of obtaining the internship.