**System Requirements Specification Index**

**For Machine learning Algorithm No 7**

1.0

**Machine Learning usecase: Support Vector Machine and Random Forests**

**You are provided with fruit prediction system and employee promotion prediction system**

**Overview**

**1. Employee Promotion Dataset (employee\_promotion\_dataset.csv)**

This dataset contains HR-related information for 1,000 employees, used to predict whether an employee will be promoted based on various personal and professional attributes.

**Features:**

* age: Age of the employee in years
* years\_at\_company: Number of years the employee has been with the company
* last\_performance\_rating: Employee’s last performance rating (integer scale, typically 1 to 5)
* education\_level: Highest level of education attained (e.g., High School, Bachelors, Masters)
* department: Department where the employee works (e.g., Sales, Finance, Tech)
* training\_hours: Total number of hours the employee has spent in training programs
* gender: Gender of the employee (Male or Female)
* is\_manager: Indicates if the employee holds a managerial position (1 = Yes, 0 = No)

**Target:**

* promotion: Indicates if the employee was promoted (1 = Yes, 0 = No)

**2.**This dataset contains numerical measurements of various fruits used to classify them into distinct categories based on physical characteristics.

**Features:**

* mass: Weight of the fruit in grams
* width: Width of the fruit in centimeters
* height: Height of the fruit in centimeters
* color\_score: A numeric score representing the color of the fruit (higher score = more vibrant color)

**Target:**

* fruit\_name: The name of the fruit (e.g., Apple, Orange, Lemon, Mango)

**1.)Your are provide with the function load\_and\_preprocess\_data()**

The dataset provided to you is **fruit\_data.csv**

* Load dataset in a DataFrame for further processing.
* Separate the feature set X by dropping the fruit\_name column (which is the label).
* Extract the target variable y, which consists of the fruit\_name column values.
* Return both the feature set X and the target variable y for use in model training and evaluation.

**2** split\_data() function, which:

* Use train\_test\_split() to divide data into training and testing sets
* Has a default test\_size=0.2
* Print the size of the resulting sets
* Return the 4 splits: X\_train, X\_test, y\_train, y\_test

3 train\_and\_save\_model() function:

* Use the RandomForestClassifier to train data
* Save the trained model to fruit\_rf\_model.pkl (by default) using joblib
* You are required to the same model name for saving the model
* Return the trained model

4) load\_model() :

* Load the saved model from the specified path (default: "fruit\_rf\_model.pkl") using joblib.load()
* Return the loaded model object

5) predict\_from\_json() function:

* Load the JSON input from fruit\_item.json
* Extracts the "fruit" object
* Convert it into a single-row DataFrame
* Pass it to the trained model for prediction
* Return the prediction

**Task 2: SVM on Employee promotion Dataset**

You will implement the following functions in `**Employee promotion Dataset**`:

Loads a dataset from "employee\_promotion\_dataset.csv"

* You need to apply one-hot encoding via pd.get\_dummies()
* Splits into:
  + X: features (excluding "promotion")
  + y: target ("promotion")
* Return the full preprocessed DataFrame, X, and y

perform\_eda(df) function:

* Filter the DataFrame for rows where "age" > 35
* Count how many such rows exist
* return that count

split\_data() function :

* Split X and y using train\_test\_split()
* Use test\_size=0.2 by default
* You need to apply a fixed random\_state=42 for reproducibility
* Return: X\_train, X\_test, y\_train, y\_test

train\_and\_save\_model() function:

* You should apply the StandardScaler to scale X\_train
* Train SVM model with a linear kernel
* Save the trained model to a .pkl file (default: "promotion\_svm\_model.pkl")
* You are required to the same model name for saving the model
* Return the trained model and the fitted scaler

**predict\_from\_json() function:**

* Load the input data from a JSON file (default: "new\_employee.json")
* Convert the JSON to a DataFrame and applies pd.get\_dummies() to handle categorical variables
* Align the input DataFrame with the expected feature\_columns used during training
* Scale the input using the provided StandardScaler
* Use the trained model to predict promotion (returns 1 for "Promoted", 0 for "Not Promoted")
* returns the prediction

**Running the Tests**

To run the tests, use the following command:

Python3 -m unittest

**Submission Guidelines**

1. Complete all the required functions in `EmployeePromotion.py` and `fruit.py`

2. Ensure all tests pass

3. Submit your code files

**Execution Steps to Follow:**

* + All actions like build, compile, running application, running test cases will be through Command Terminal.
  + To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
  + This editor Auto Saves the code
  + If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use **CTRL+Shift+B** -command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
  + These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
  + To setup environment:

You can run the application without importing any packages

* + To launch application:
  + **Python3 EmployeePromotion.py` and `python3 fruit.py**
  + To run Test cases:

**python3 -m unittest**

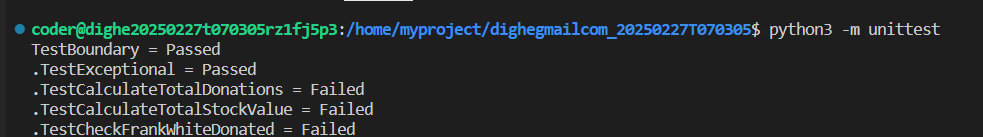
* + Before Final Submission also, you need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

**Screen shot to run the program**

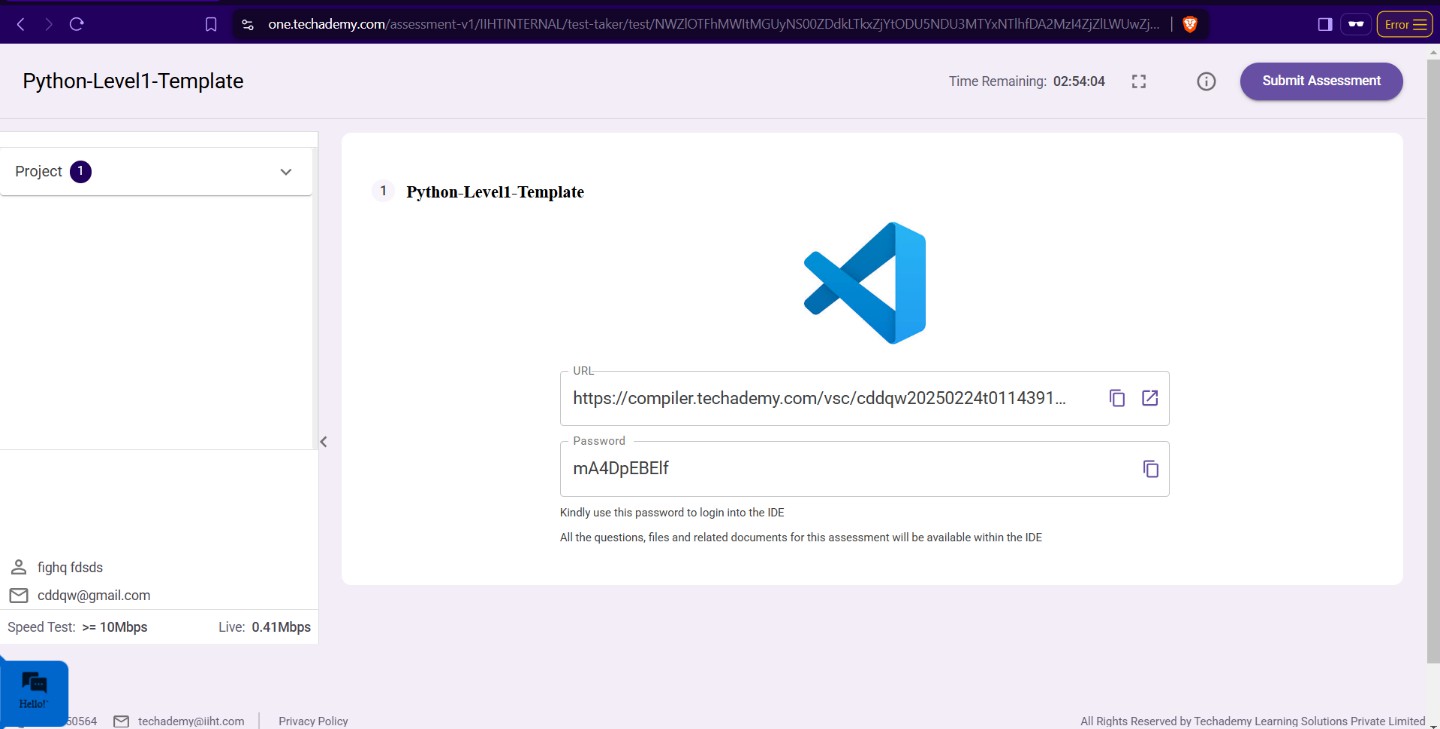
**To run the application**



* + **Python3 EmployeePromotion.py`**
  + **`python3 fruit.py**



**To run the testcase python3 -m unittest**



* + **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.**