

Section	Description	Points	Grade Breakdown and Requirements			Weightage
		60	What 80-100% looks like	What 60-80% looks like	What <60% looks like	100.00%
Data Registration	<ul style="list-style-type: none"> - Create a master folder and create a subfolder "data" - Register the data on the Hugging Face dataset space 	2	<ul style="list-style-type: none"> - Create a master folder and create a subfolder "data" [1] - Register the data on the Hugging Face dataset space [1] 			3.33%
Exploratory Data Analysis	<ul style="list-style-type: none"> - Data collection and background - Data overview - Univariate analysis - Bivariate analysis - Multivariate analysis - Insights/observations based on EDA 	3	<ol style="list-style-type: none"> 1) Observations on shape of data, data types of various attributes, statistical summary. [1] 2) Insight-based questions given in learner notebook answered. [1] 3) Univariate and bivariate analysis (variable distributions, interactions between variables) to understand the relationships in data beyond the set of questions already provided. [1] 	<ol style="list-style-type: none"> 1) Definition of problem 2) Observations on shape of data, data types of various attributes, statistical summary. 3) Insight-based questions given in learner notebook answered. 	<ol style="list-style-type: none"> 1) Definition of problem 2) Observations on shape of data and data types of various attributes 3) Some of the insight-based questions given in learner notebook answered. 	5.00%
Data preparation	<ul style="list-style-type: none"> - Load the dataset directly from the Hugging Face data space. - Perform data cleaning and remove any unnecessary columns. - Split the cleaned dataset into training and testing sets, and save them locally. - Upload the resulting train and test datasets back to the Hugging Face data space. 	4	<ul style="list-style-type: none"> - Load the dataset directly from the Hugging Face data space. [1] - Perform data cleaning and remove any unnecessary columns.[1] - Split the cleaned dataset into training and testing sets, and save them locally [1] - Upload the resulting train and test datasets back to the Hugging Face data space [1] 	<ul style="list-style-type: none"> - Load the dataset directly from the Hugging Face data space. - Split the cleaned dataset into training and testing sets, and save them locally - Upload the resulting train and test datasets back to the Hugging Face data space 	<ul style="list-style-type: none"> - Load the dataset directly from the Hugging Face data space. - Upload the resulting train and test datasets back to the Hugging Face data space 	6.67%
Model Building with Experimentation Tracking	<ul style="list-style-type: none"> - Load the train and test data from the Hugging Face data space - Define a model and parameters - Tune the model with the defined parameters - Log all the tuned parameters - Evaluate the model performance - Register the best model in the Hugging Face model hub <p>* The ML models to be built can be any of the following algorithms, such as Decision Tree, Bagging, Random Forest, AdaBoost, Gradient Boosting, and XGBoost</p>	6	<ul style="list-style-type: none"> - Load the train and test data from the Hugging Face data space [1] - Define a model and parameters [1] - Tune the model with the defined parameters [1] - Log all the tuned parameters [1] - Evaluate the model performance [1] - Register the best model into the Hugging Face model hub [1] 	<ul style="list-style-type: none"> - Load the train and test data from the Hugging Face data space - Define a model and parameters - Tune the model with the defined parameters - Log all the tuned parameters - Register the best model into the Hugging Face model hub 	<ul style="list-style-type: none"> - Load the train and test data from the Hugging Face data space - Define a model and parameters - Tune the model with the defined parameters - Register the best model into the Hugging Face model hub 	10.00%
Model Deployment	<ul style="list-style-type: none"> - Define a Dockerfile and list all configurations - Load the saved model from the Hugging Face model hub - Get the inputs and save them into a dataframe - Define a dependencies file for the deployment - Define a hosting script that can push all the deployment files into the Hugging Face space 	12	<ul style="list-style-type: none"> - Define a Dockerfile and list all configurations [3] - Load the saved model from the Hugging Face model hub [3] - Get the inputs and save them into a dataframe [1] - Define a dependencies file for the deployment [3] - Define a hosting script that can push all the deployment files into Hugging Face space [2] 	<ul style="list-style-type: none"> - Define a Dockerfile and list all configurations - Get the inputs and save them into a dataframe - Define a dependencies file for the deployment - Define a hosting script that can push all the deployment files into Hugging Face space 	<ul style="list-style-type: none"> - Define a Dockerfile and list all configurations - Define a dependencies file for the deployment - Define a hosting script that can push all the deployment files into Hugging Face space 	20.00%
Automated Github Actions Workflow	<ul style="list-style-type: none"> - Create a pipeline.yml file in the GitHub repo - Define a YAML file and list all steps to execute each step of Machine Learning - Push all files to GitHub - Automate the end-to-end workflow - Update the workflow to automatically push code updates to the main branch 	15	<ul style="list-style-type: none"> - Create a pipeline.yml file in the GitHub repo [4] - Define a YAML file and list all steps to execute each step of Machine Learning [3] - Push all files to GitHub [3] - Automate the end-to-end workflow [2] - Update the workflow to automatically push code updates to the main branch [3] 	<ul style="list-style-type: none"> - Create a pipeline.yml file in the GitHub repo - Define a YAML file and list all steps to execute each step of Machine Learning - Push all files to GitHub 	<ul style="list-style-type: none"> - Create a pipeline.yml file in the GitHub repo - Push all files to GitHub 	25.00%
Output Evaluation	<ul style="list-style-type: none"> - GitHub (link to repository, screenshot of folder structure and executed workflow) - Streamlit on Hugging Face (link to HF space, screenshot of Streamlit app) 	8	<ul style="list-style-type: none"> - GitHub <ul style="list-style-type: none"> - link to repository [2] - screenshot of folder structure and executed workflow [3] - Streamlit on Hugging Face <ul style="list-style-type: none"> - link to HF space [1] - screenshot of Streamlit app [2] 	<ul style="list-style-type: none"> - GitHub <ul style="list-style-type: none"> - link to repository - screenshot of folder structure and executed workflow - Streamlit on Hugging Face <ul style="list-style-type: none"> - link to HF space 	<ul style="list-style-type: none"> - GitHub <ul style="list-style-type: none"> - link to repository - Streamlit on Hugging Face <ul style="list-style-type: none"> - link to HF space 	13.33%

Actionable Insights and Recommendations	Key takeaways for the business	4	<ul style="list-style-type: none"> - 3-4 Actionable Insights are mentioned [2] - 2-3 Recommendations are mentioned [2] <p>[Recommendations can also include points on additional data sources for further analysis, model implementation in real world, potential business benefits from improving the model, etc.]</p>	<ul style="list-style-type: none"> - 1-2 Actionable Insights mentioned - 1 Recommendation mentioned 	Any Actionable Insights or Recommendations mentioned	6.67%
Business Report Quality	- Adhere to the business report checklist	6	<ul style="list-style-type: none"> Objective, guidance, and data description: 1 point Exclusion of code: 2 points Structure and readability: 1 point Rationale and logic: 1 point Visual clarity and referencing: 1 point 	<ul style="list-style-type: none"> Objective, guidance, and data description Structure and readability Rationale and logic 	<ul style="list-style-type: none"> Objective, guidance, and data description Rationale and logic 	10.00%