mlangles

CloudAngles

mlangles Predictive AI

DIABETES PREDICTION

Use Case

mlangles

CloudAngles

Customer Overview

A machine learning model that accurately predicts an individual's likelihood of developing diabetes based on their medical history and demographic information

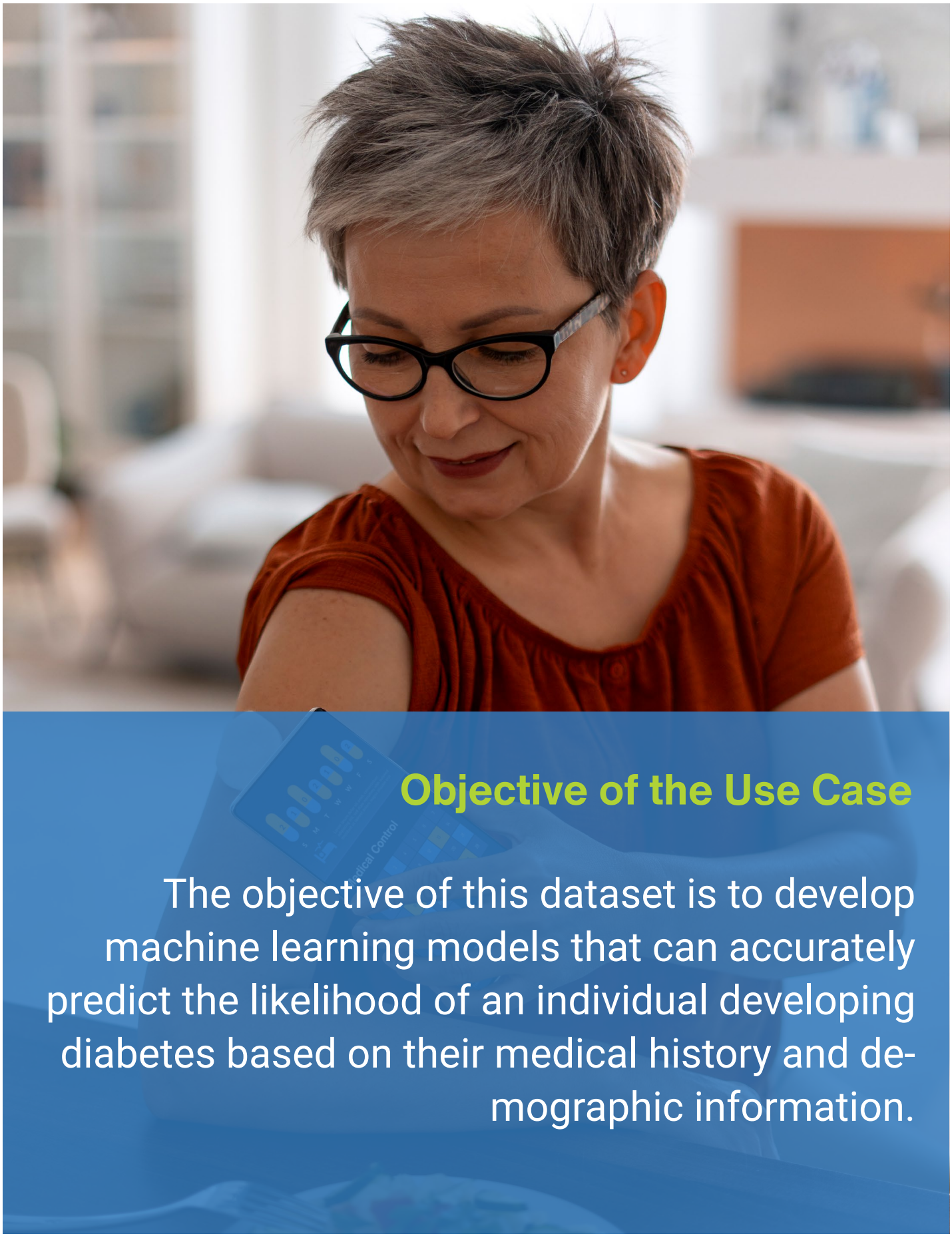
Challenges

- Early Detection & Personalization: Detecting diabetes early is challenging due to its often asymptomatic initial stages. Additionally, diabetes risk factors vary widely among individuals due to genetics, lifestyle, and environmental influences.
- Complexity of Data: Navigating extensive medical data, including genetic information, medical history, lifestyle factors, and environmental data, to detect diabetes.
- Data Integration: Integrating diverse healthcare data sources such as electronic health records (EHRs), medical imaging, wearable devices, and genetic databases.
- Prediction Accuracy: Traditional risk assessment methods for diabetes lack precision and often struggle to capture the ultra-fine interactions between risk factors.

About mlangles Predictive AI

mlangles is a comprehensive AI platform designed to manage the lifecycle of data and models, offering streamlined solutions for every stage of the process.

Through its Predictive AI component, mlangles provides a suite of tools to navigate efficiently through each phase of AI project development, encompassing data engineering, development, deployment, and monitoring. It facilitates continuous integration, continuous deployment, continuous training, continuous monitoring (CI-CD-CT-CM), enabling enterprises to effectively manage their AI initiatives.



Objective of the Use Case

The objective of this dataset is to develop machine learning models that can accurately predict the likelihood of an individual developing diabetes based on their medical history and demographic information.

Explanation of the use case

The Diabetes prediction dataset is a collection of medical and demographic data from patients, along with their diabetes status (positive or negative). The data includes features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level. This dataset can be used to build machine learning models to predict diabetes in patients based on their medical history and demographic information. This can be useful for healthcare professionals in identifying patients who may be at risk of developing diabetes and in developing personalized treatment plans. Additionally, the dataset can be used by researchers to explore the relationships between various medical and demographic factors and the likelihood of developing diabetes.

Features in the Dataset:

Gender: Gender refers to the biological sex of the individual, which can have an impact on their susceptibility to diabetes.

Age: Age is an important factor as diabetes is more commonly diagnosed in older adults. Age ranges from 0-80 in our dataset.

Hypertension: Hypertension is a medical condition in which the blood pressure in the arteries is persistently elevated.

Heart disease: Another medical condition associated with an increased risk of diabetes.

Smoking history: Smoking history is also considered a risk factor for diabetes and can exacerbate the complications associated with it.

BMI: BMI (Body Mass Index) is a measure of body fat based on weight and height.

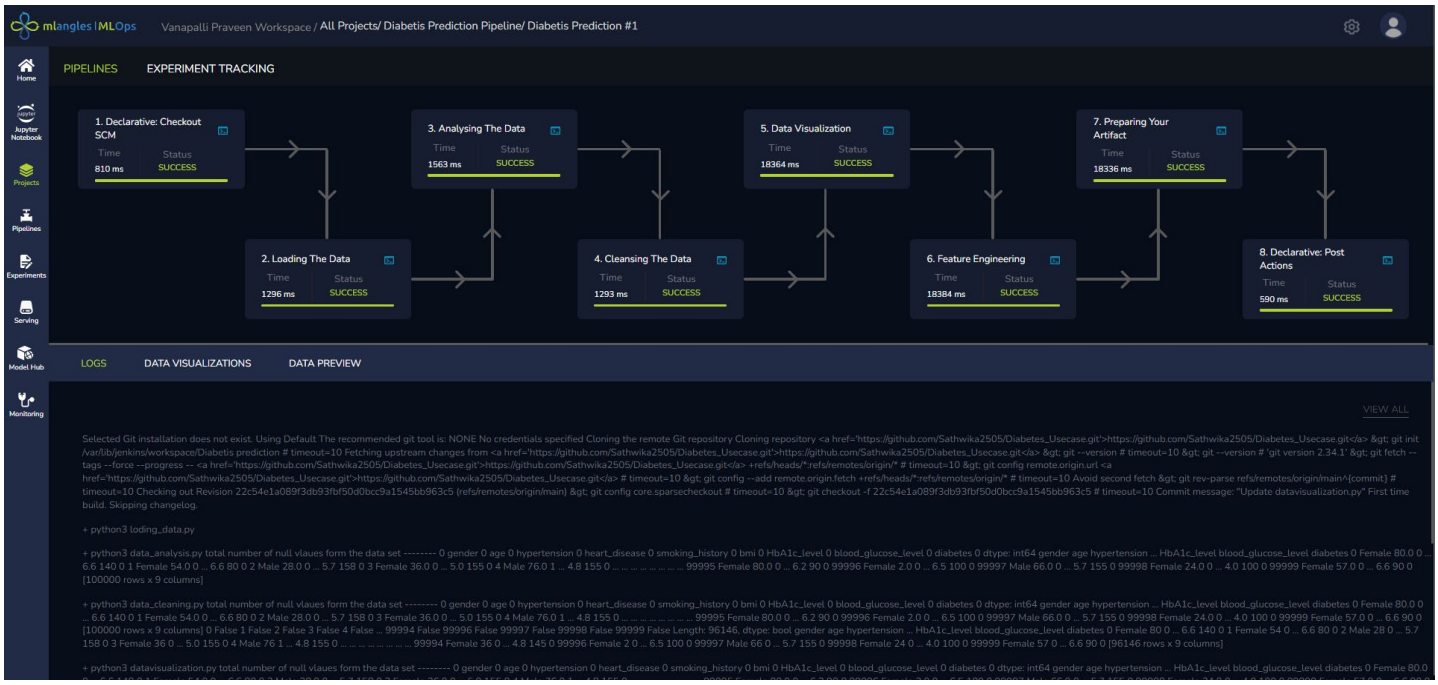
Hemoglobin A1c: HbA1c (Hemoglobin A1c) level is a measure of a person's average blood sugar level over the past 2-3 months.

Blood glucose level: Blood glucose level refers to the amount in the bloodstream.

Diabetes (Target Variable): Diabetes is the target variable being predicted, with values of 1 indicating the presence of diabetes and 0 indicating the absence of it.

Working of the use case

Step 1: Data Engineering and Pipeline Creation



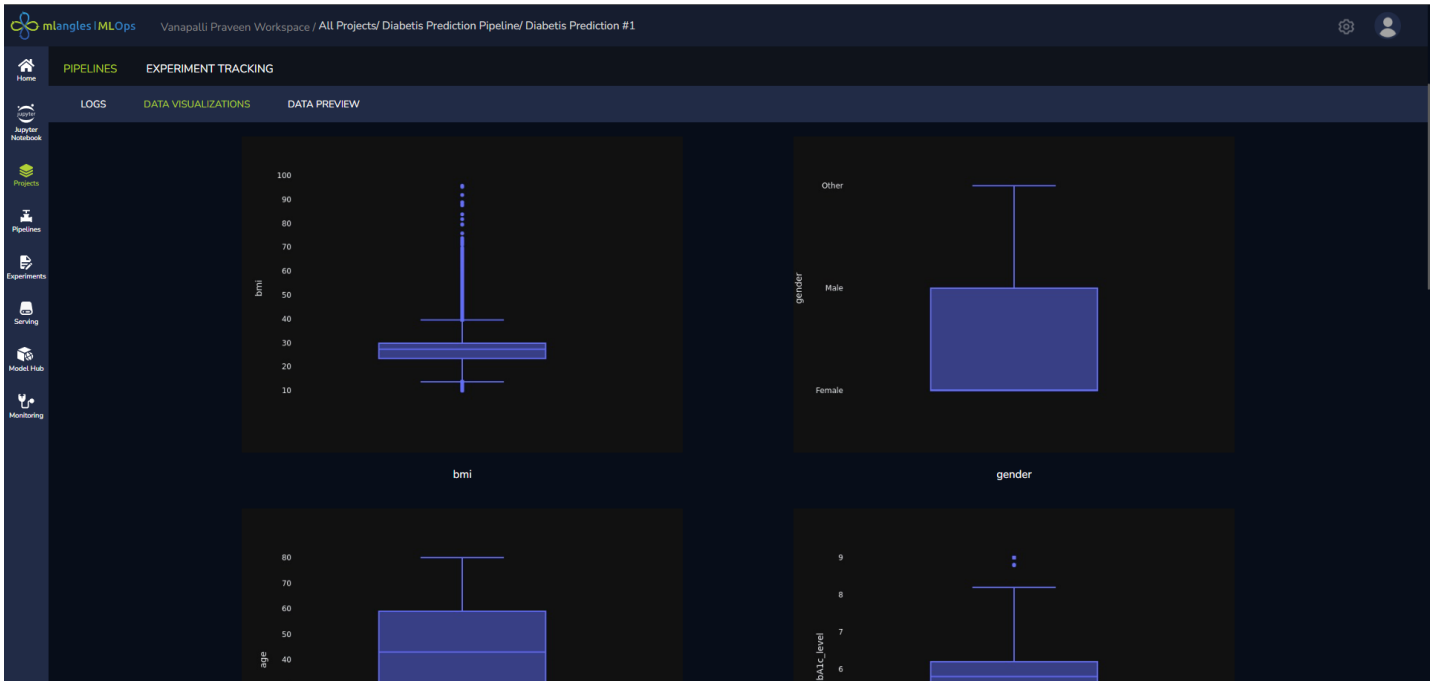
Install Dependencies: Essential packages and libraries have been installed.

Data Extraction: Data extraction involves gathering raw data from the CSV file.

Data Analysis: Once the raw data is collected, the next step is to analyse it to gain insights and understand its characteristics. Data analysis involves examining the structure, patterns, and relationships within the data. This step helps in identifying trends, outliers, or any anomalies present in the dataset. Techniques such as descriptive statistics, correlation analysis, and data profiling are commonly used during this stage.

Data Preprocessing: After analyzing the data, it's common to encounter inconsistencies, missing values, or errors that need to be addressed. Data cleaning involves preprocessing the data to ensure its quality and reliability. This may include tasks such as inputting missing values, removing duplicates, standardizing formats, and handling outliers. The goal is to prepare the data for further analysis and modeling.

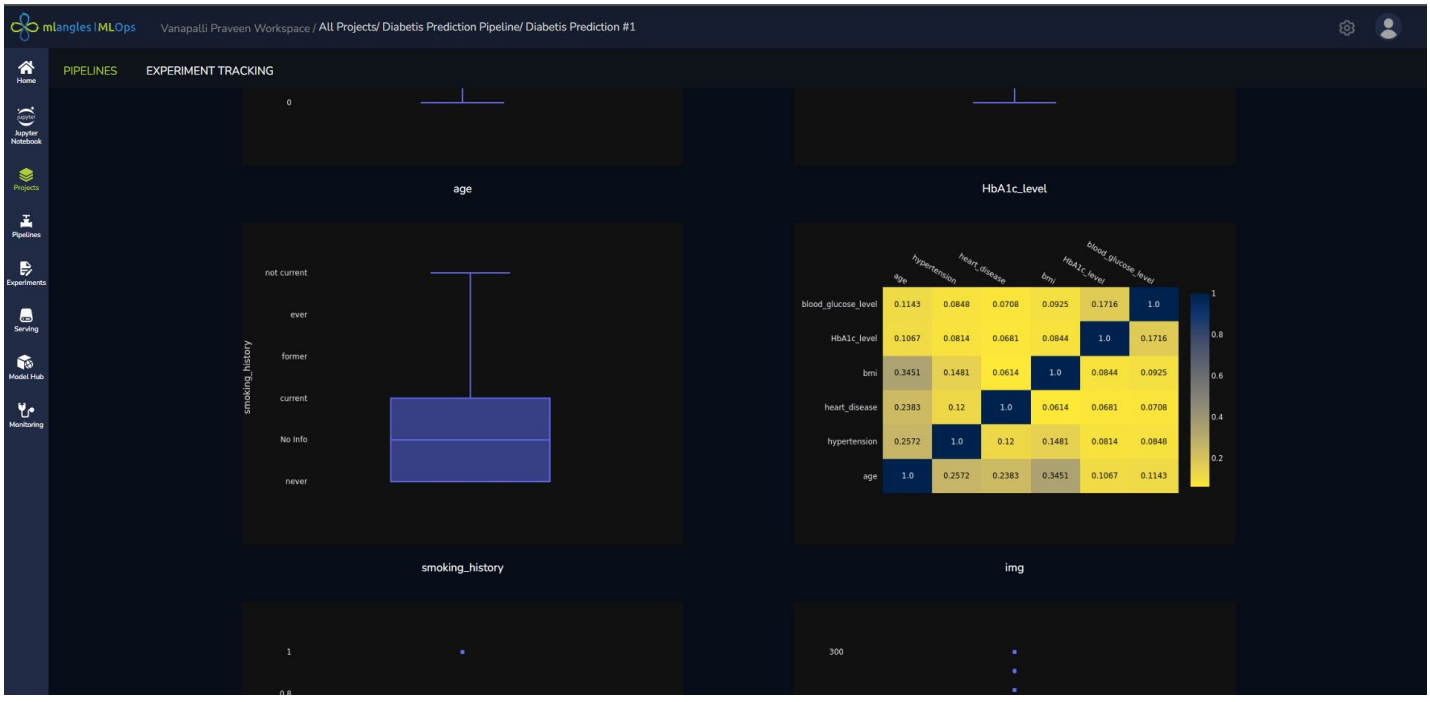
Data Visualization: Data visualization is a powerful tool for exploring and communicating insights from the data. Visualizations such as box plots, histograms, and heat maps are used to represent different aspects of the data distribution, relationships, and trends. Box plots are useful for visualizing the distribution of numerical data and detecting outliers. Histograms provide a graphical representation of the frequency distribution of continuous variables. Heat maps are effective for visualizing the correlation between variables in a tabular dataset.



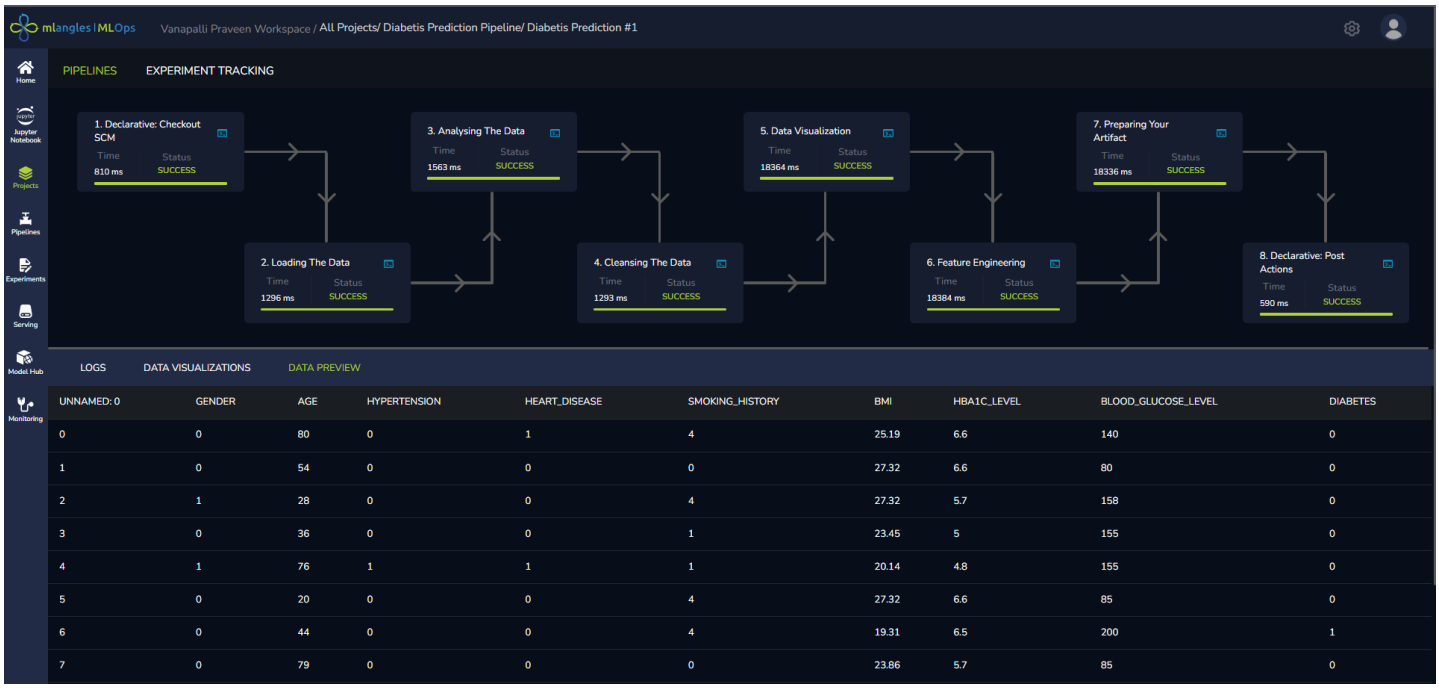
Box Plots: They are commonly used in exploratory data analysis to compare distributions across different groups or categories, identify outliers, and assess the variability within a dataset.

Correlation Matrix: It is a table that displays the pairwise correlation coefficients between variables in a dataset. Correlation matrices are widely used in data analysis, especially in fields like finance, economics, and social sciences, to identify patterns and relationships between variables. They help researchers understand the underlying structure of the data and can guide feature selection.

Data Preprocessing: After analyzing the data, it's common to encounter inconsistencies, missing values, or errors that need to be addressed. Data cleaning involves preprocessing the data to ensure its quality and reliability. This may include tasks such as inputting missing values, removing duplicates, standardizing formats, and handling outliers. The goal is to prepare the data for further analysis and modeling.



Feature Engineering: This step aims to extract relevant information from the data and represent it in a format that is suitable for modelling. Feature engineering techniques include encoding categorical variables, scaling numerical features, creating interaction terms, and extracting domain-specific features. The goal is to enhance the predictive power of the model by providing it with informative and discriminative features.



UNNAMED: 0	GENDER	AGE	HYPERTENSION	HEART_DISEASE	SMOKING_HISTORY	BMI	HBA1C_LEVEL	BLOOD_GLUCOSE_LEVEL	DIABETES
0	0	80	0	1	4	25.19	6.6	140	0
1	0	54	0	0	0	27.32	6.6	80	0
2	1	28	0	0	4	27.32	5.7	158	0
3	0	36	0	0	1	23.45	5	155	0
4	1	76	1	1	1	20.14	4.8	155	0
5	0	20	0	0	4	27.32	6.6	85	0
6	0	44	0	0	4	19.31	6.5	200	1
7	0	79	0	0	0	23.86	5.7	85	0

Data Versioning:

- Various processed data versions can be generated through different transformations applied to the same raw dataset, such as deleting columns or applying various transformations on specific columns.
- Throughout the data pipeline, diverse transformations can be executed at each iteration. Consequently, the resulting data at the pipeline's end is systematically versioned.
- Given that each version of the final data is distinct, models trained on these different versions will exhibit varying behaviors.

Step 2: Experiment Tracking- Modelling with Hyper-Parameter Optimization

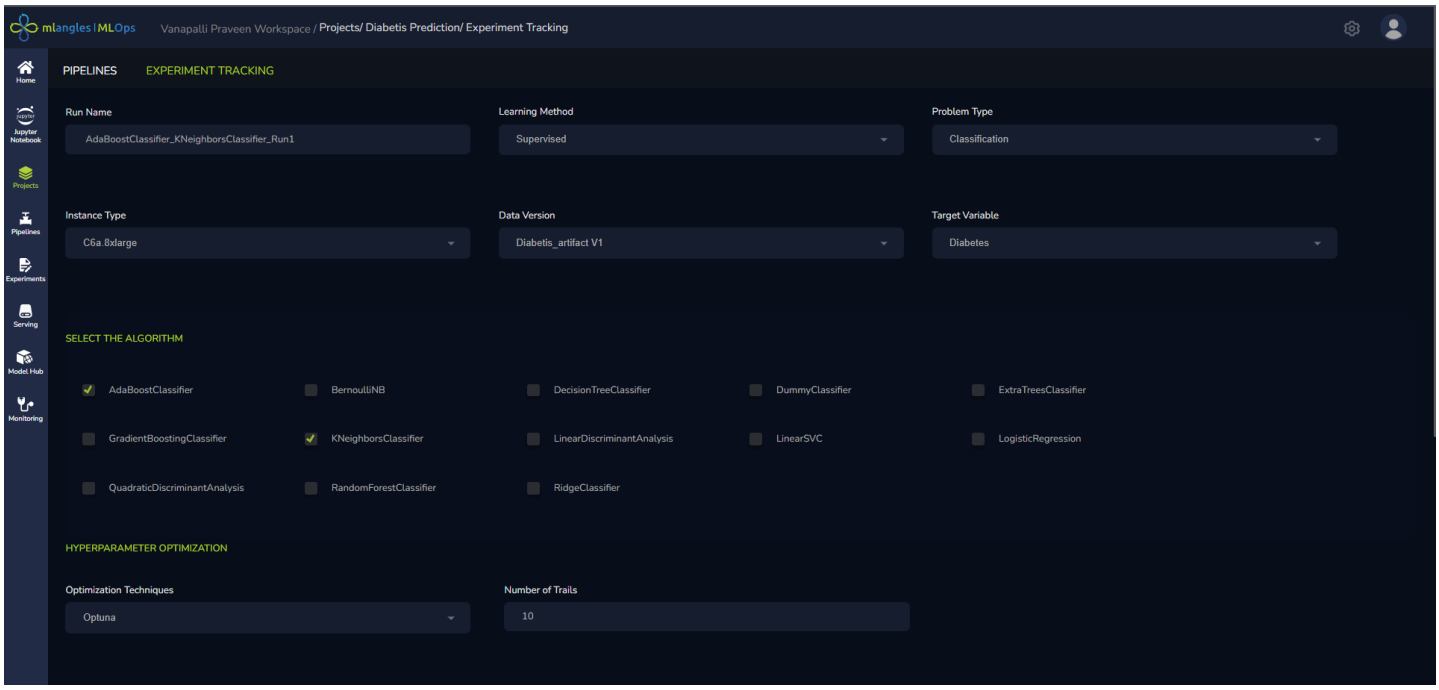
After the data has been prepared and cleaned, the subsequent step involves training a model using this refined dataset. Since the problem at hand is a classification task, there are several models suitable for this purpose. Common options include the AdaBoost Classifier and K-Neighbors Classifier.

AdaBoost Classifier: Trains a series of weak learners sequentially, adjusting the weights ofmisclassified instances at each iteration, and combines their predictions to form a strongensemble model with improved performance.

KNN classifier: The fundamental idea behind the KNN classifier is to classify a data pointbased on the majority class of its nearest neighbors in the feature space. It assumes thatsimilar data points will have similar class labels.

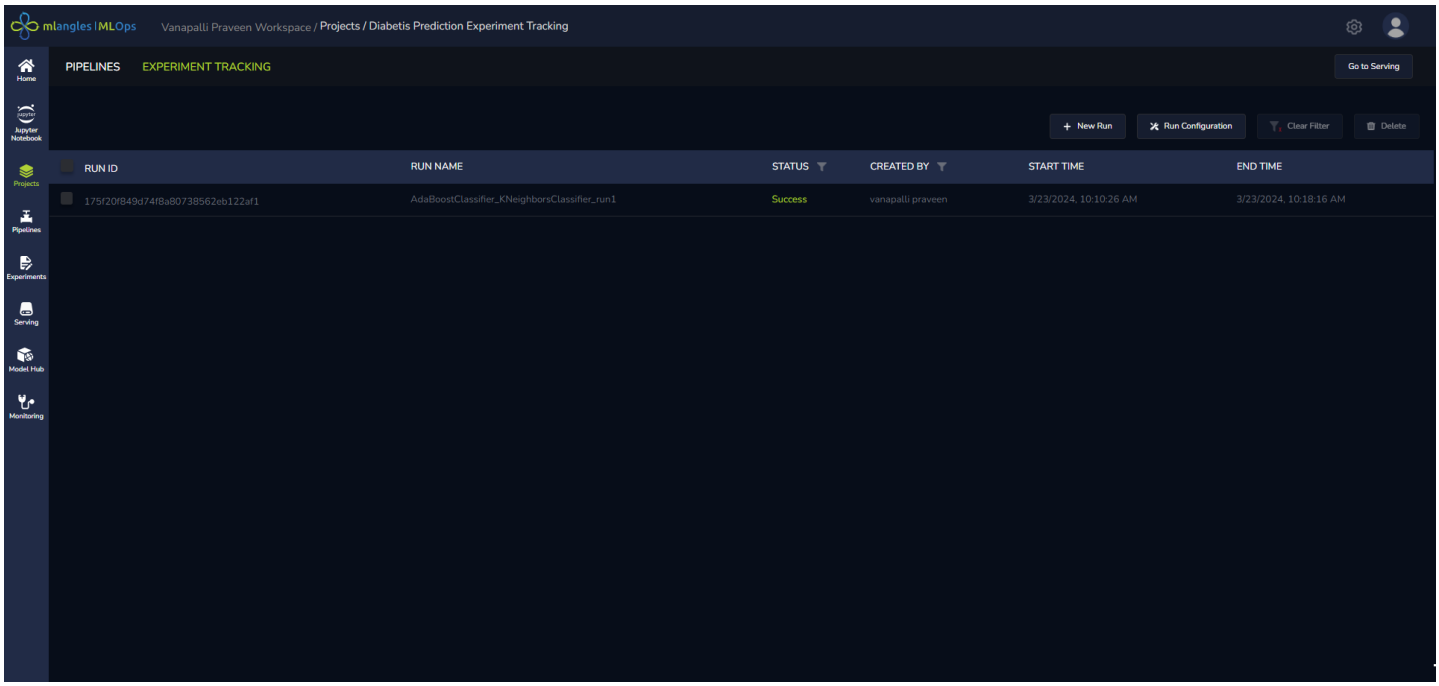
Gradient Boosting Classifier: Sequentially builds a series of weak learners (usually decisiontrees) by correcting the errors of the previous models, producing a strong predictive modelwith high accuracy.

Decision Tree Classifier: Decision trees partition the data into subsets based on the valuesof features and make decisions at each node. They are straightforward yet effective forclassification tasks.

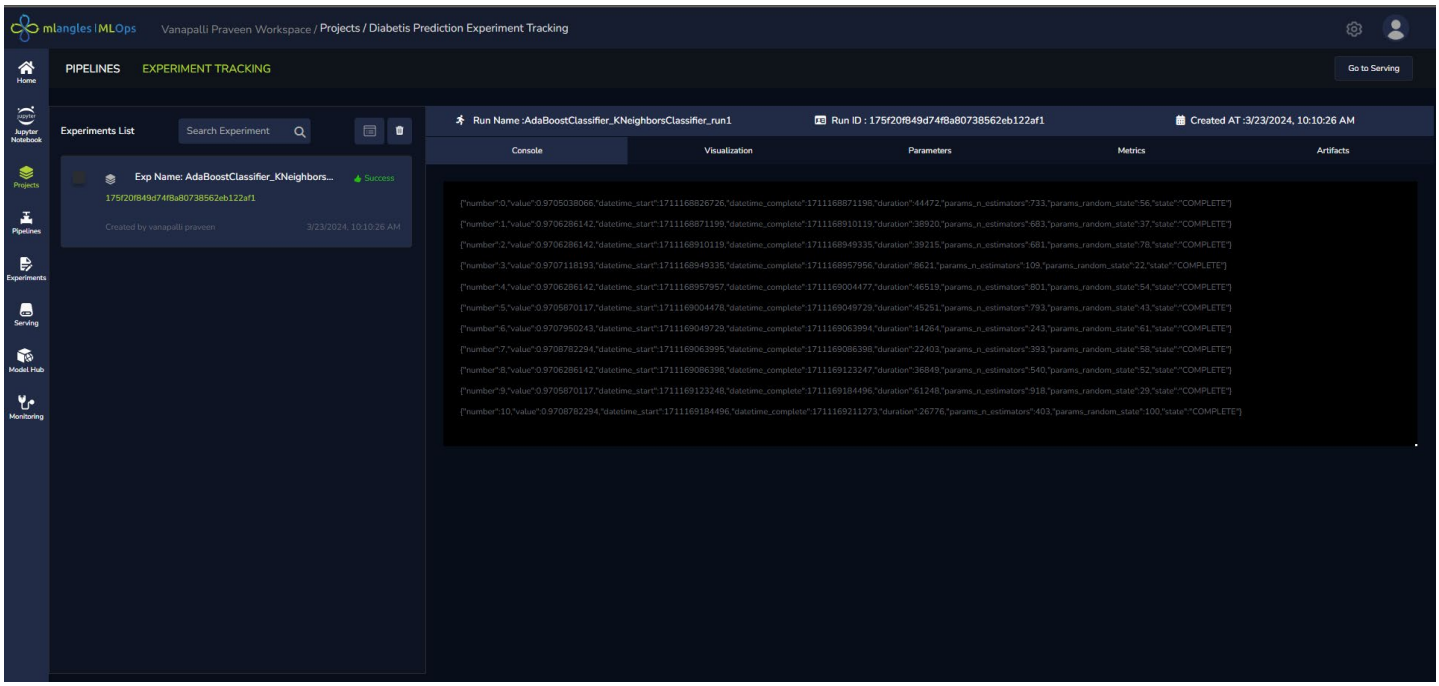


Additionally, to enhance model performance, a hyperparameter optimization technique called Optuna is employed. Optuna automates the process of tuning hyperparameters, such as learning rate or tree depth, to find the optimal configuration that maximizes model performance. This approach ensures that the model is fine-tuned to achieve the best possible results on the given dataset, improving its accuracy and predictive power.

After generating the run, a range of information can be extracted, such as visualizations of hyperparameters for the optimal algorithm, parameters employed during training, and metrics along with artifacts. These observations offer crucial insights into the model's performance and characteristics, facilitating comprehension of its efficacy and areas where enhancements can be made.



After a run is created there is a console where we can find the hyper parameters for each trial that has been taken.



Hyperparameter Visualization

Visual representations of hyperparameters illustrate the influence of various parameter settings on model performance, aiding in the identification of optimal configurations.

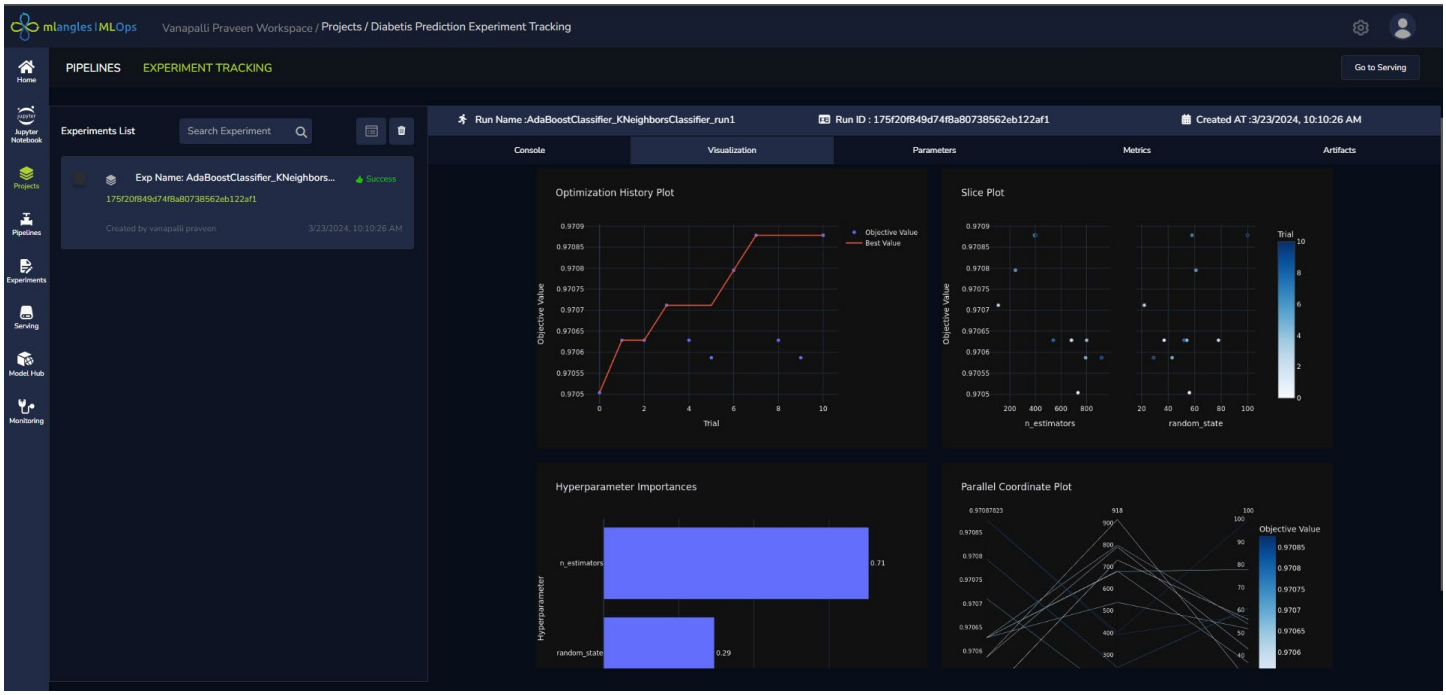
The different types of plots that represent the values of the parameters for each trail are as follows:

Optimization History Plot: The Optimization History Plot illustrates the evolution of the objective function (e.g., accuracy or loss) throughout the hyperparameter search iterations, providing insights into convergence patterns and the efficacy of the optimization algorithm.

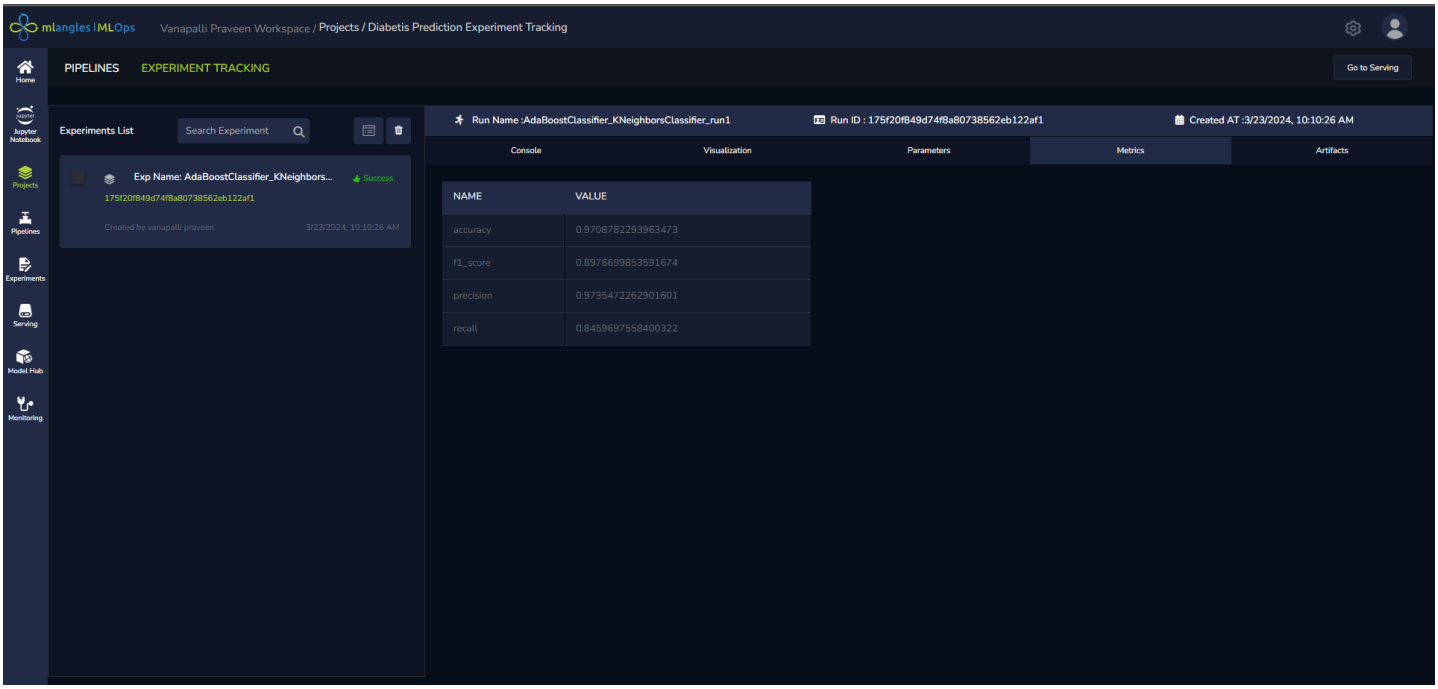
Slice Plot: A Slice Plot depicts the correlation between two hyperparameters while keeping the values of other hyperparameters constant. This visualization enables the exploration of interactions among hyperparameters and their impact on model performance, aiding in the discovery of optimal parameter combinations.

Hyperparameter Importances Plot: The Hyperparameter Importances Plot ranks the significance of hyperparameters according to their impact on model performance. This visualization assists in identifying the most influential hyperparameters, informing subsequent optimization endeavors or strategies for feature selection.

Parallel Coordinate Plot: The Parallel Coordinate Plot represents high-dimensional hyperparameter spaces by depicting each hyperparameter as a vertical axis and each point in the plot as a hyperparameter configuration. Lines connecting points illustrate.

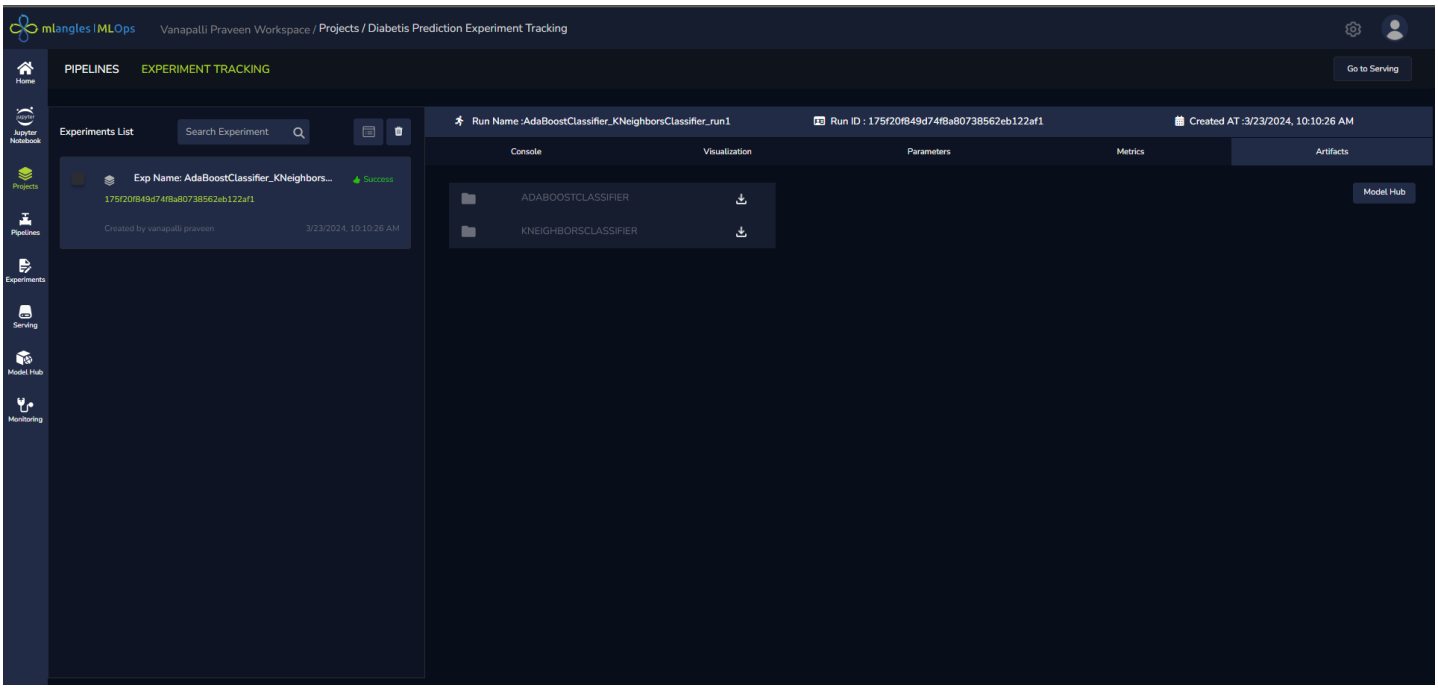


Metrics such as accuracy offer insights into the model's performance. Accuracy measures overall correctness, precision assesses the accuracy of positive predictions, recall emphasizes capturing all positive instances, and the F1 score balances precision and recall. These metrics collectively aid in evaluating the effectiveness and robustness of classification models.



Downloading Artifacts:

Artifacts is the model that is trained, and we can download the model for further inference.

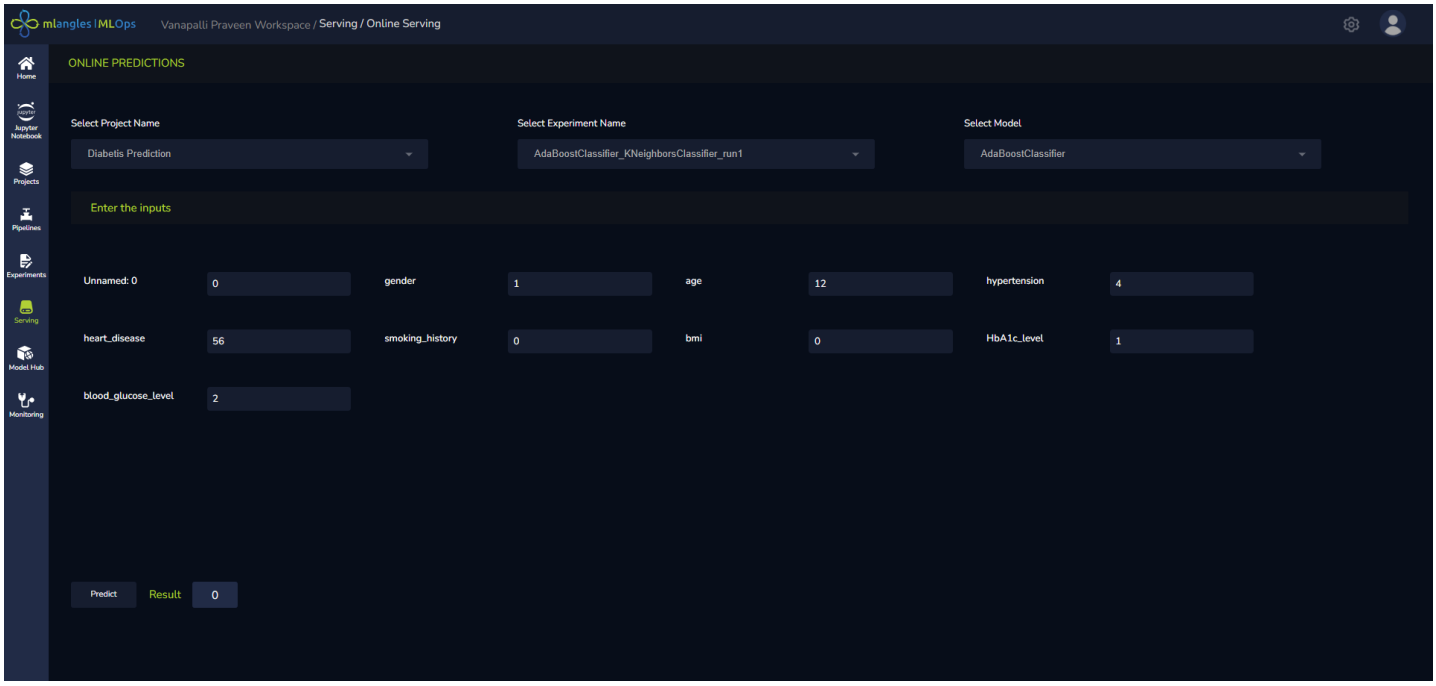


Model Versioning:

- Models are sensitive to a plethora of hyperparameters and parameters, including learning rate, loss function, and optimizers.
- Consequently, a model selected for training, with both the model and final data versions remaining constant but changes in parameters, may yield differing performance metrics.
- These diverse model versions can be uploaded to the model hub, facilitating the management of multiple iterations and variations.

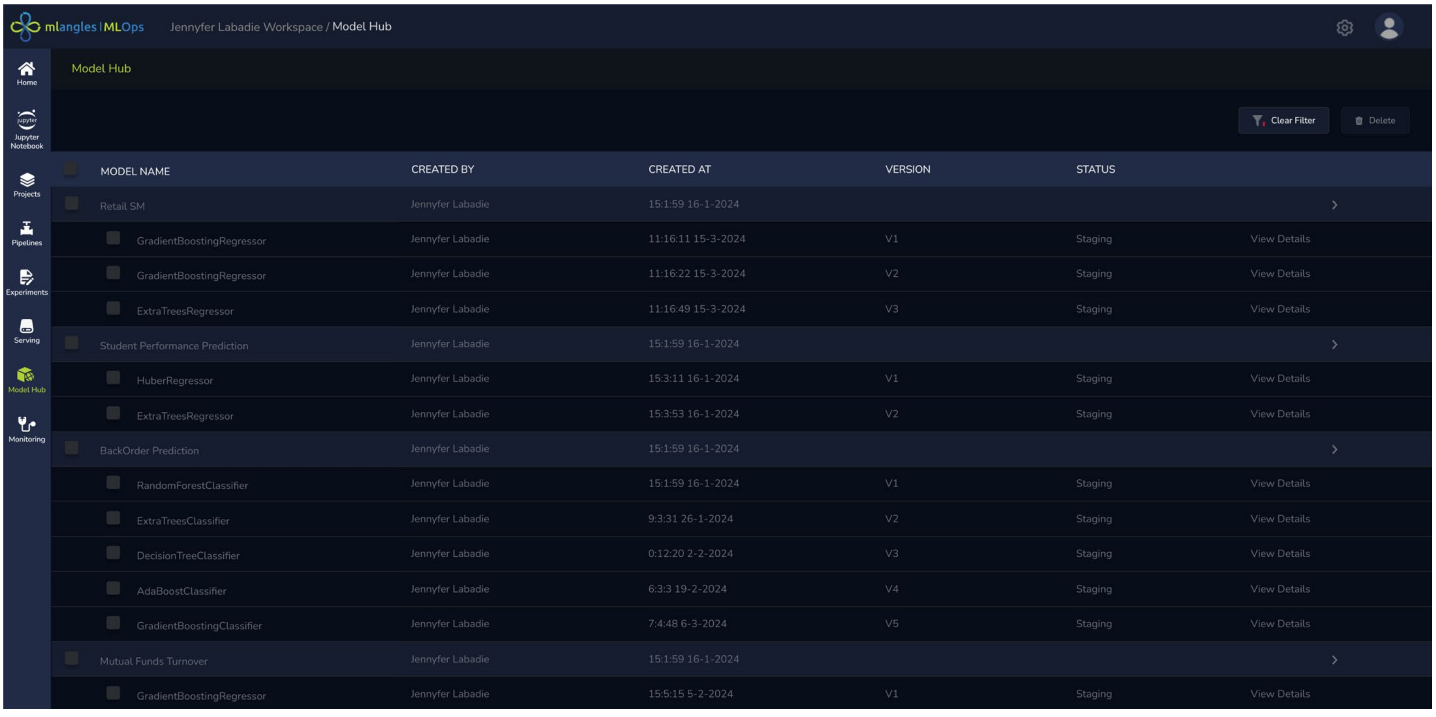
Step 3: Serving

In a serving context, we provide values for each column of data and use this information to predict whether a person has diabetes or not. The model learns from historical data during training and uses this knowledge to make predictions on new data.



Model Hub:

- Trained models are uploaded to the model hub, whereupon deployment, a REST API endpoint is automatically generated.
- Data is transmitted to this endpoint as a request, triggering the model to execute a prediction and return the output as the response to the request.



Step 4: Monitoring

Monitoring for data drift involves regularly assessing the distribution of patient data over time and updating machine learning models accordingly. This ensures that the models remain accurate and reliable in predictions, guiding treatment decisions, and improving patient outcomes. Effective monitoring and adaptation strategies are essential for maintaining the relevance and effectiveness of machine learning applications in the management of predicting the disease.

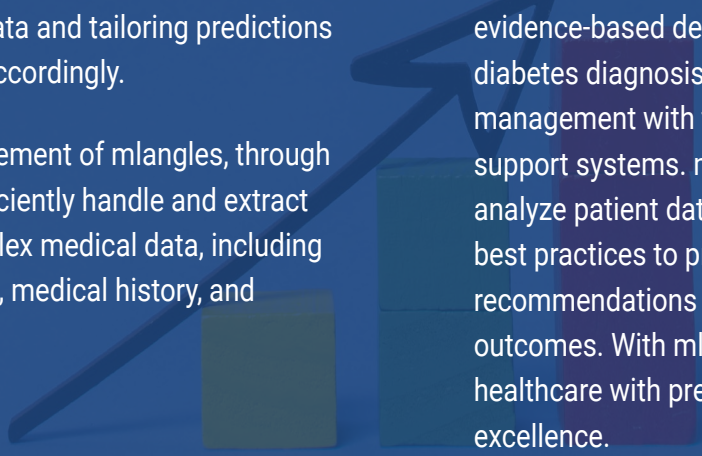
This explains that 11 out of 11 features are affecting the data drift. This change could result from alterations in the data distribution, data collection methods, or external influences.

Conclusion

Incorporating advanced classification algorithms into medical data analysis demonstrates their effectiveness in accurately predicting diseases based on diverse health parameters. By leveraging machine learning techniques, healthcare practitioners can enhance diagnostic accuracy and treatment planning, ultimately leading to better patient outcomes. These findings underscore the importance of integrating advanced technology into healthcare practices, paving the way for more efficient and personalized patient care.

Business Impact of mlangles Predictive AI

- The 2021 report from the International Diabetes Federation indicates that approximately 10.5% of adults currently live with diabetes, with projections suggesting a 46% increase by 2045. mlangles' Predictive AI models can analyze various health indicators to identify individuals at high risk of developing diabetes before symptoms appear prominently, allowing for early interventions and prevention strategies.
- With the help of mlangles Predictive AI , develop personalized risk profiles by analyzing individual patient data and tailoring predictions and interventions accordingly.
- The Predictive AI element of mlangles, through algorithms, can efficiently handle and extract insights from complex medical data, including genetic information, medical history, and lifestyle factors.
- mlangles' AI systems integrate all types of healthcare data sources to provide a comprehensive view of a patient's health status, enabling more accurate predictions of diabetes risk.
- With mlangles' AI components - machine learning and deep learning, leverage large datasets to identify complex patterns and improve the accuracy of diabetes risk prediction models.
- Assist healthcare providers in making evidence-based decisions related to diabetes diagnosis, treatment, and management with the help of mlangles AI support systems. mlangles models can analyze patient data, medical literature, and best practices to provide personalized recommendations and improve patient outcomes. With mlangles, revolutionize healthcare with precision, efficiency, and excellence.



To setup Demo

Info.mlangles@cloudangles.com

Visit: www.mlangles.ai