mlangles

CloudAngles

mlangles Predictive AI

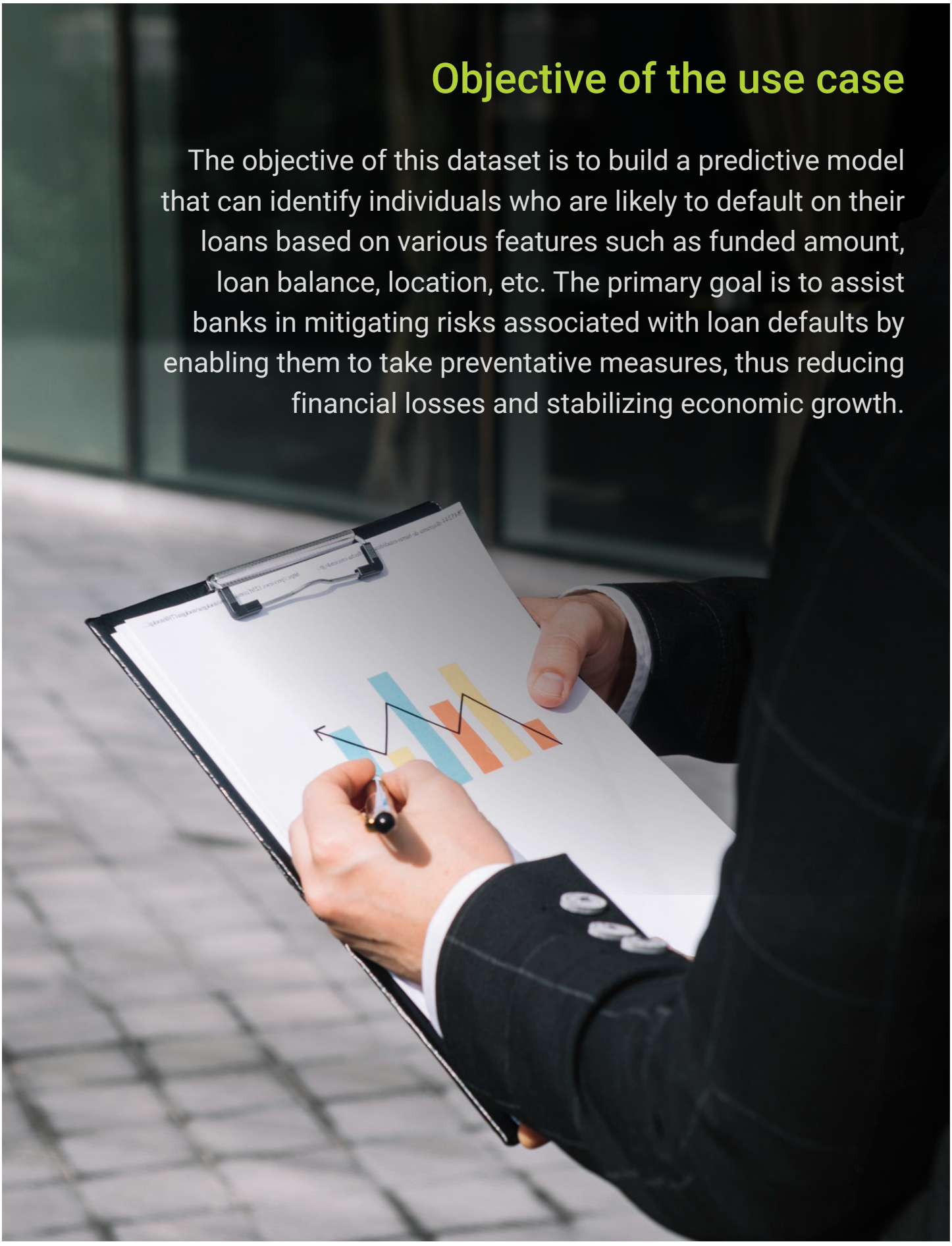
Bank Loan Defaulter Prediction

mlangles

CloudAngles

Objective of the use case

The objective of this dataset is to build a predictive model that can identify individuals who are likely to default on their loans based on various features such as funded amount, loan balance, location, etc. The primary goal is to assist banks in mitigating risks associated with loan defaults by enabling them to take preventative measures, thus reducing financial losses and stabilizing economic growth.



Explanation of the use case

The dataset for this hackathon comprises two separate sets: a training dataset and a testing dataset. Here is an overview of each, along with what you can expect from them:

Training Dataset

Size: The training dataset contains 67,463 rows, each representing a loan instance or a customer. There are 35 columns, which include various attributes that describe the loan, the customer, and other related data points.

Purpose: The training dataset is used to develop and fine-tune a machine learning model. It includes information on whether a customer has defaulted on a loan, which serves as the target variable for model training.

Attributes: The columns in this dataset may contain information such as:

- Loan details like funded amount, loan balance, interest rate, and loan term.
- Customer details like location, employment status, and income level.
- Other factors that might impact loan repayment, such as credit score or previous default history.

Target Variable: This is the column indicating whether the customer has defaulted on their loan. The target variable is what the model aims to predict.

Testing Dataset

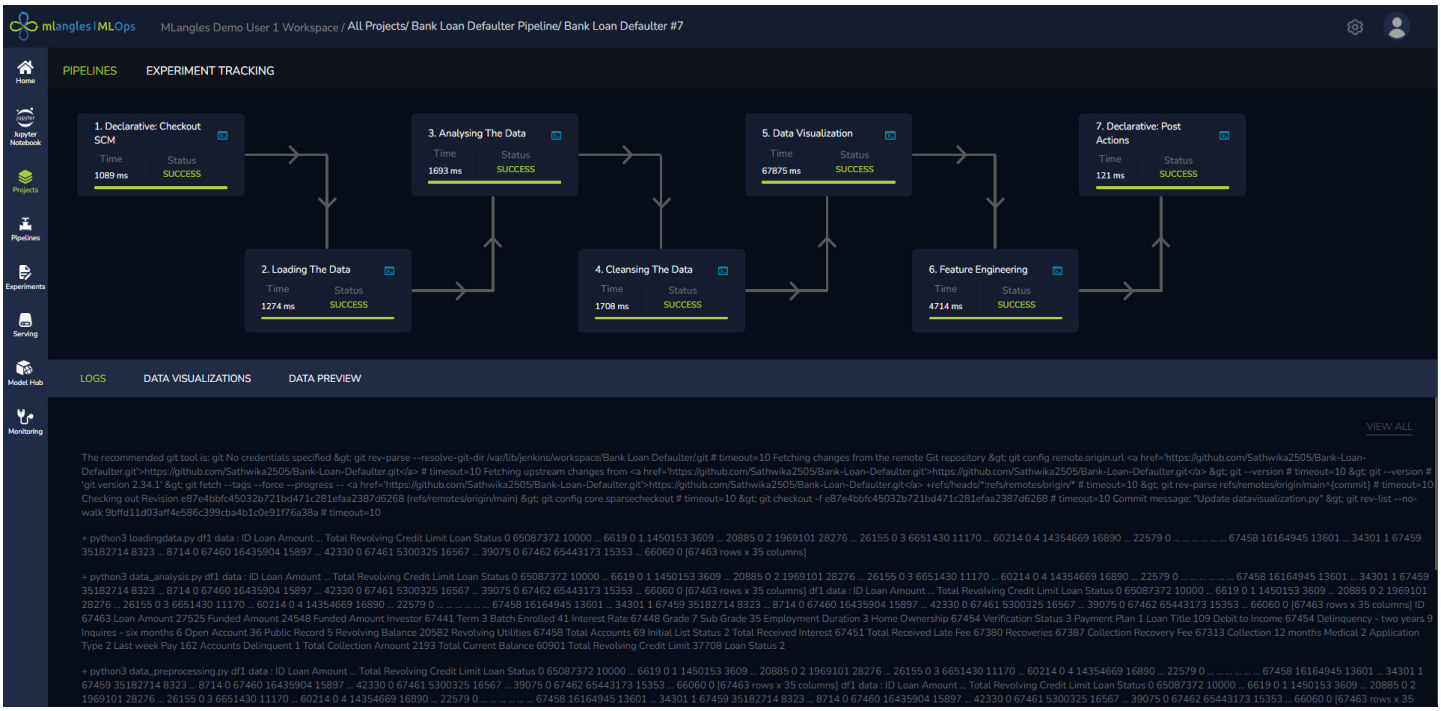
Size: The testing dataset has 28,913 rows with 34 columns. This dataset is smaller in size and lacks the target variable (whether a loan was defaulted).

Purpose: The testing dataset is used to evaluate the performance of the trained machine learning model. The goal is to ensure the model generalizes well to new data, avoiding overfitting or underfitting.

Attributes: The columns in the testing dataset are similar to those in the training dataset, except they don't include the target variable (loan default). This allows for an unbiased assessment of the model's accuracy.

Working of the use case

Step 1: Data Engineerining and Pipeline Creation



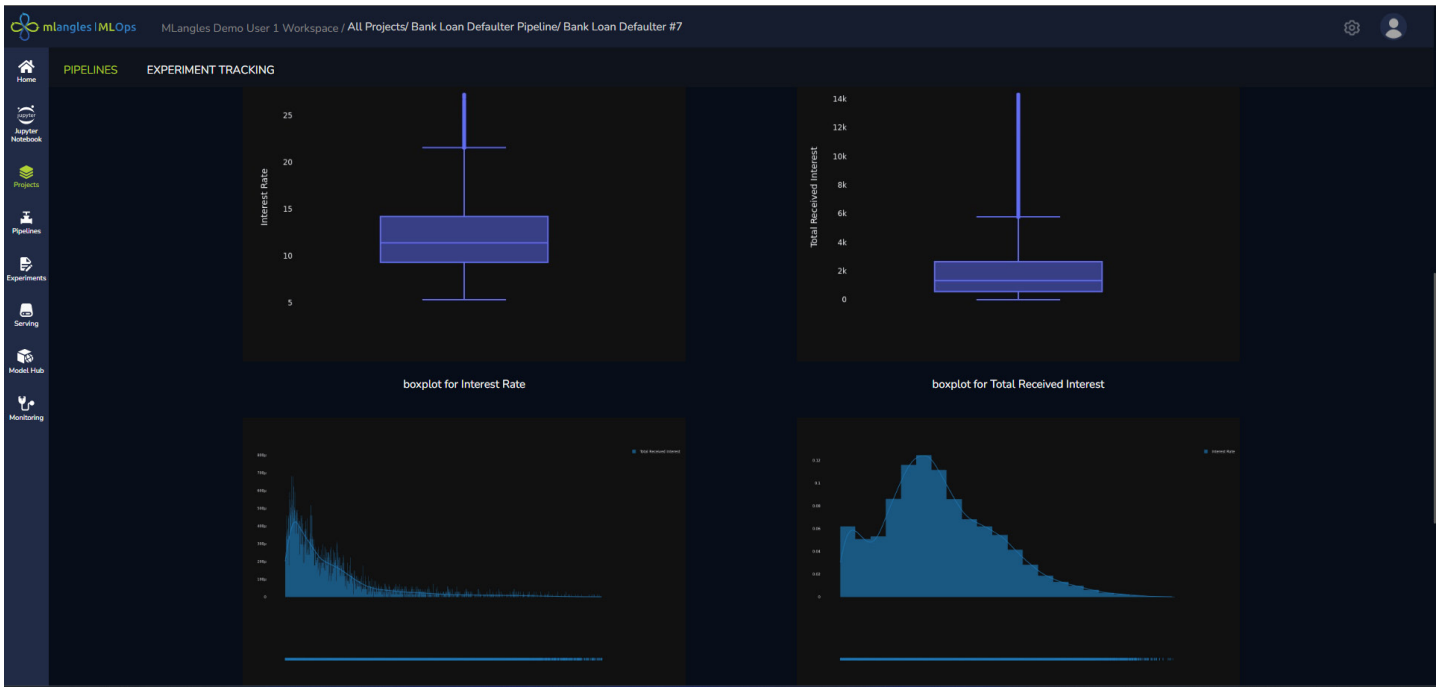
Install Dependencies: Essential packages and libraries have been installed.

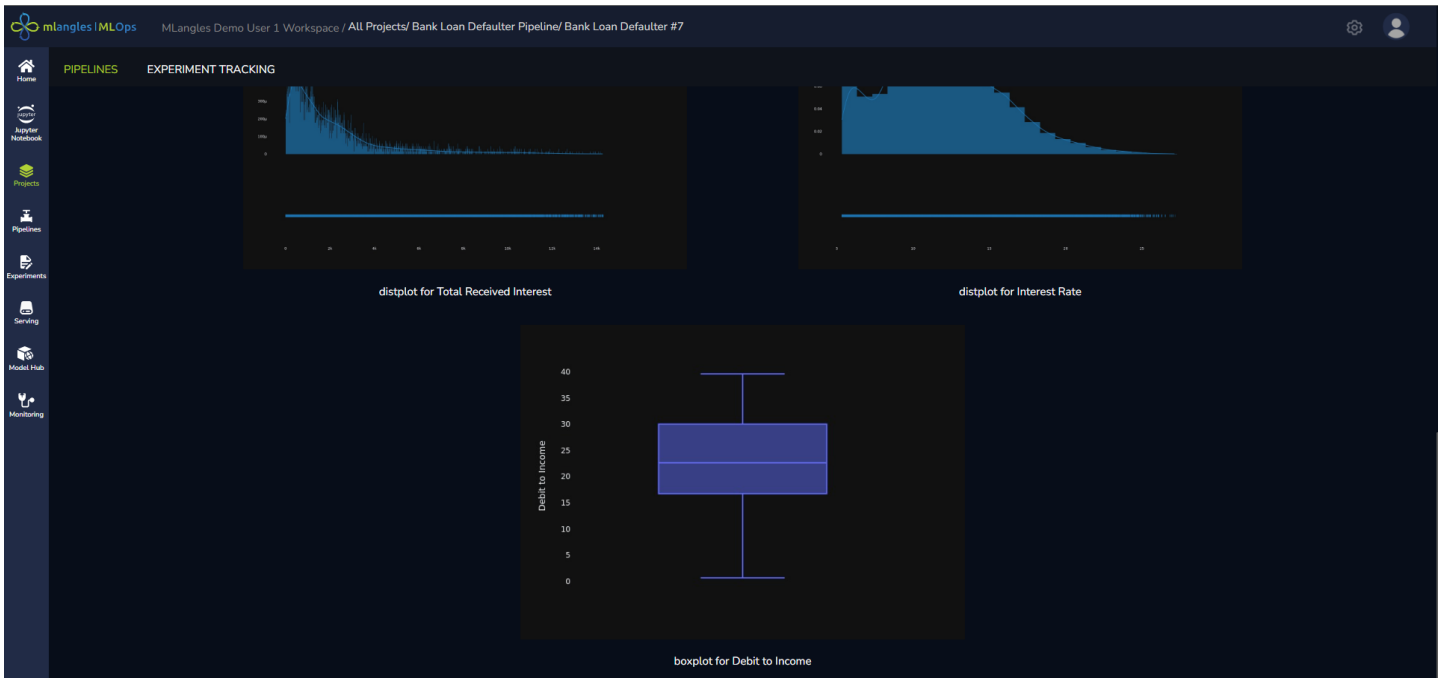
Data Extraction: Data extraction involves gathering raw data from the CSV file.

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.



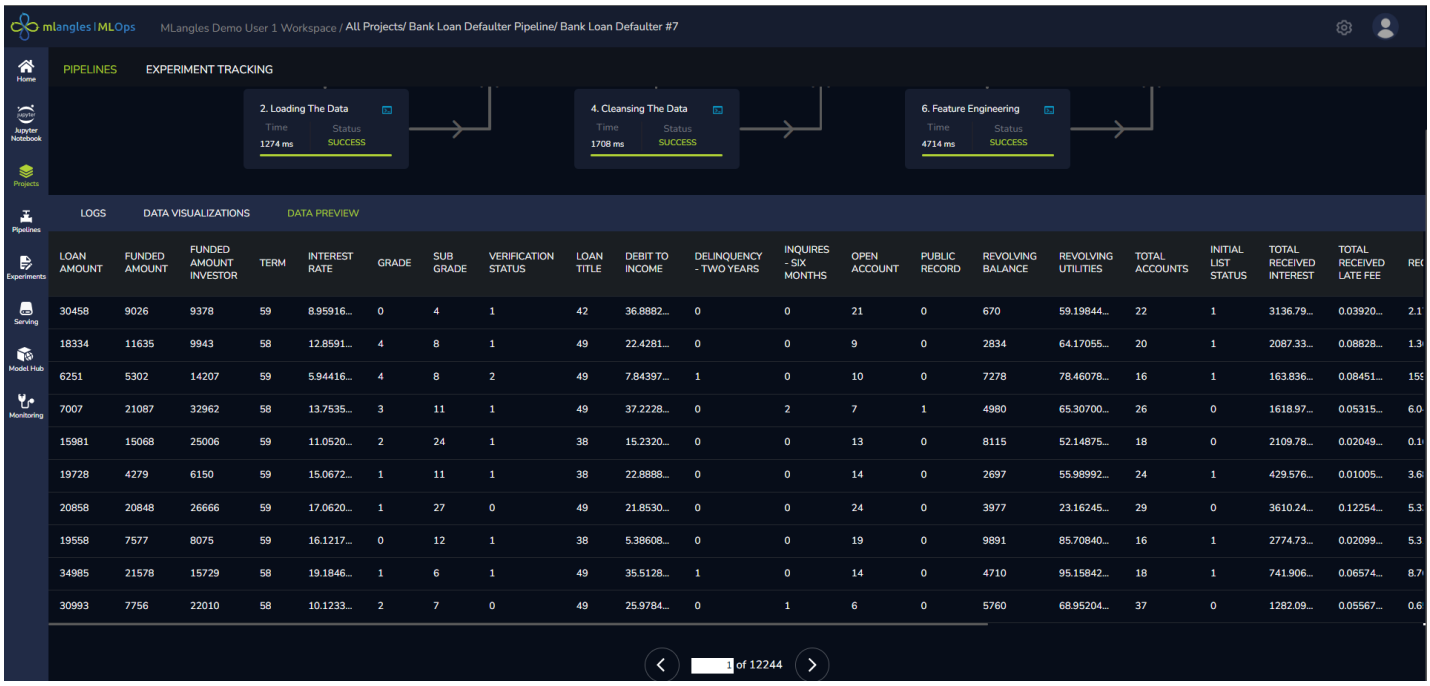


Box Plots: They are commonly used in exploratory data analysis to compare distributions across different groups or categories, identify outliers, and assess the variability within a dataset.

Dist Plots:

- Identifying data patterns and anomalies.
- Understanding the distribution before modeling or statistical analysis.
- Comparing distributions between different groups or categories.
- Assessing assumptions for statistical tests (like normality).

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.

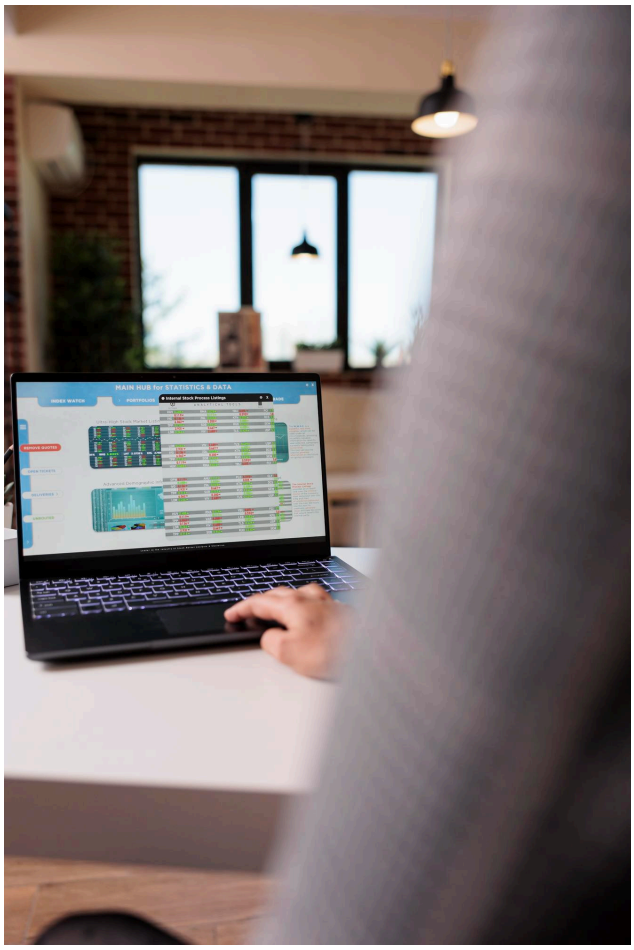


LOAN AMOUNT	FUNDED AMOUNT	FUNDED AMOUNT INVESTOR	TERM	INTEREST RATE	GRADE	SUB GRADE	VERIFICATION STATUS	LOAN TITLE	DEBIT TO INCOME	DELINQUENCY - TWO YEARS	INQUIRES - SIX MONTHS	OPEN ACCOUNT	PUBLIC RECORD	REVOLVING BALANCE	REVOLVING UTILITIES	TOTAL ACCOUNTS	INITIAL LIST STATUS	TOTAL RECEIVED INTEREST	TOTAL RECEIVED LATE FEE	REC
30458	9026	9378	59	8.95916...	0	4	1	42	36.8882...	0	0	21	0	670	59.19844...	22	1	3136.79...	0.03920...	2.1
18334	11635	9943	58	12.8591...	4	8	1	49	22.4281...	0	0	9	0	2834	64.17055...	20	1	2087.33...	0.08828...	1.3
6251	5302	14207	59	5.94416...	4	8	2	49	7.84397...	1	0	10	0	7278	78.46078...	16	1	163.836...	0.08451...	15
7007	21087	32962	58	13.7535...	3	11	1	49	37.2228...	0	2	7	1	4980	65.30700...	26	0	1618.97...	0.05315...	6.0
15981	15068	25006	59	11.0520...	2	24	1	38	15.2320...	0	0	13	0	8115	52.14875...	18	0	2109.78...	0.02049...	0.1
19728	4279	6150	59	15.0672...	1	11	1	38	22.8888...	0	0	14	0	2697	55.98992...	24	1	429.576...	0.01005...	3.6
20858	20848	26666	59	17.0620...	1	27	0	49	21.8530...	0	0	24	0	3977	23.16245...	29	0	3610.24...	0.12254...	5.3
19558	7577	8075	59	16.1217...	0	12	1	38	5.38608...	0	0	19	0	9891	85.70840...	16	1	2774.73...	0.02099...	5.3
34985	21578	15729	58	19.1846...	1	6	1	49	35.5128...	1	0	14	0	4710	95.15842...	18	1	741.906...	0.06574...	8.7
30993	7756	22010	58	10.1233...	2	7	0	49	25.9784...	0	1	6	0	5760	68.95204...	37	0	1282.09...	0.05567...	0.6

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.

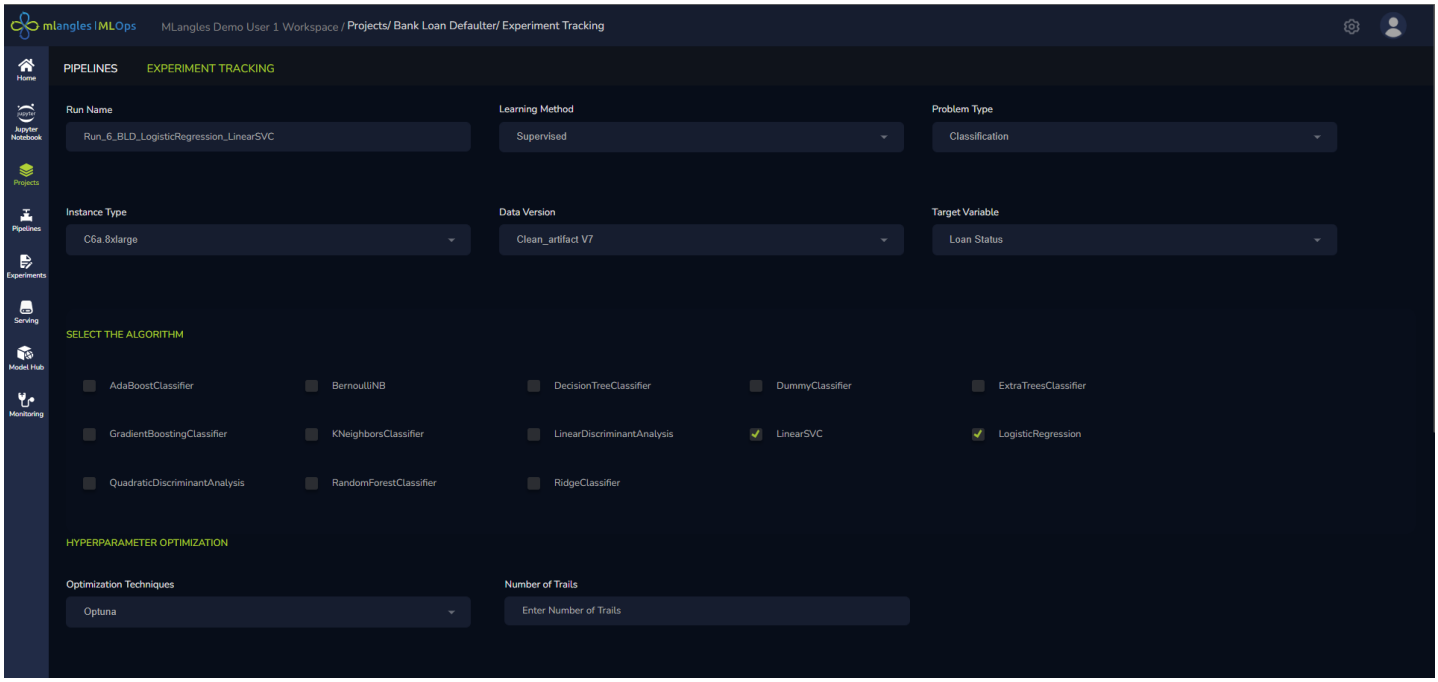
STEP 2: Experriment Tracking-Modelling with Hyperparameter Optimization



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.

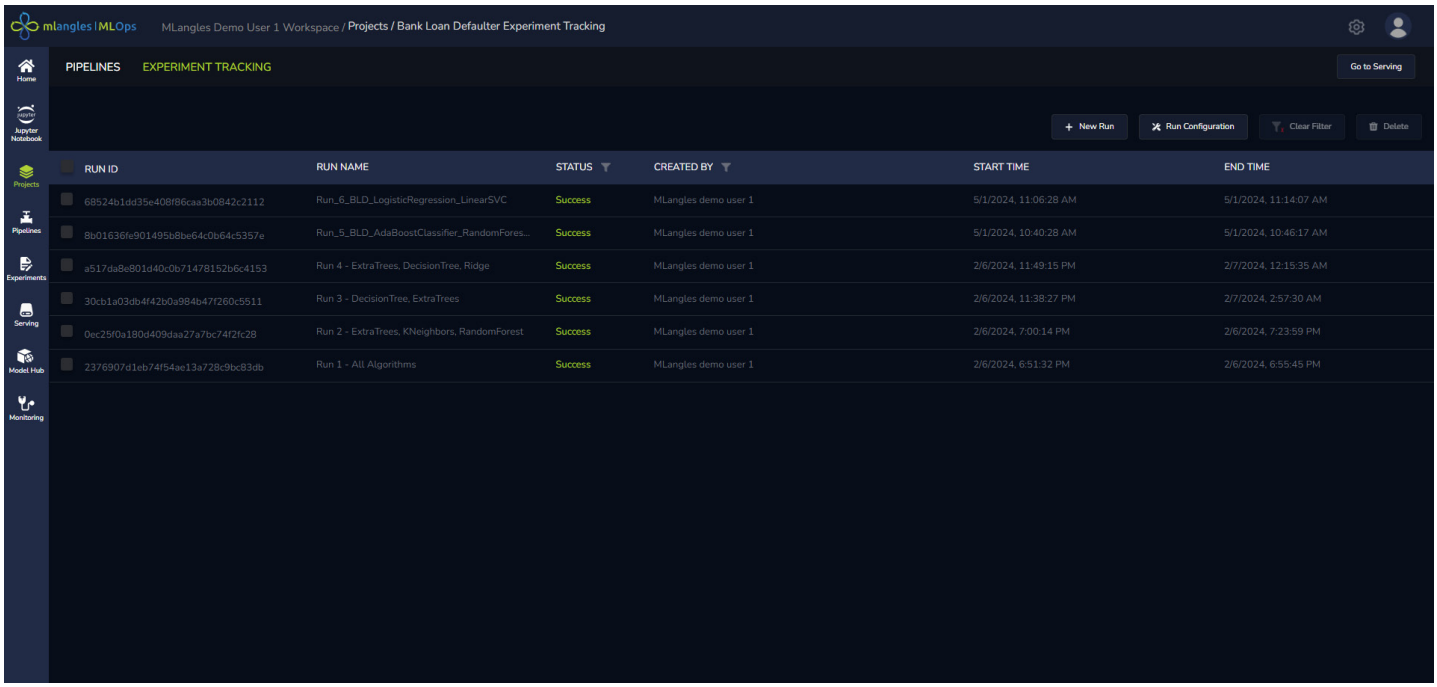
LinearSVC (Support Vector Classification): The goal is to maximize the margin between classes, which is the distance between the hyperplane and the closest data points from each class (support vectors). It uses different techniques, such as regularization, to avoid overfitting and to generalize better.

Logistic Regression: It works by modeling the log-odds of a class as a linear combination of the input features. The logistic function maps these log-odds to probabilities between 0 and 1, allowing for probabilistic interpretation.

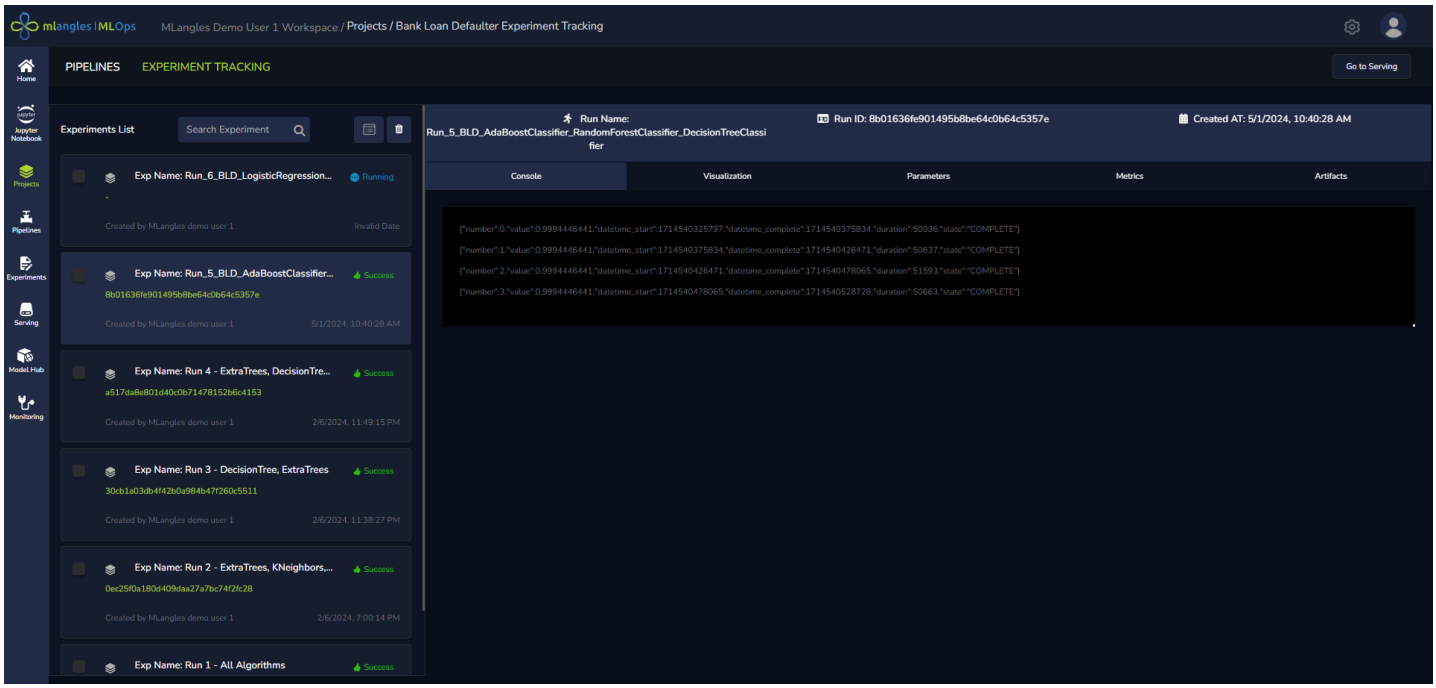


Additionally, to enhance model performance, a hyperparameter optimization technique called Optuna is employed. Optuna automates the process of tuning hyperparameters, such as learning rate or tree depth, to find the optimal configuration that maximizes model performance. This approach ensures that the 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.

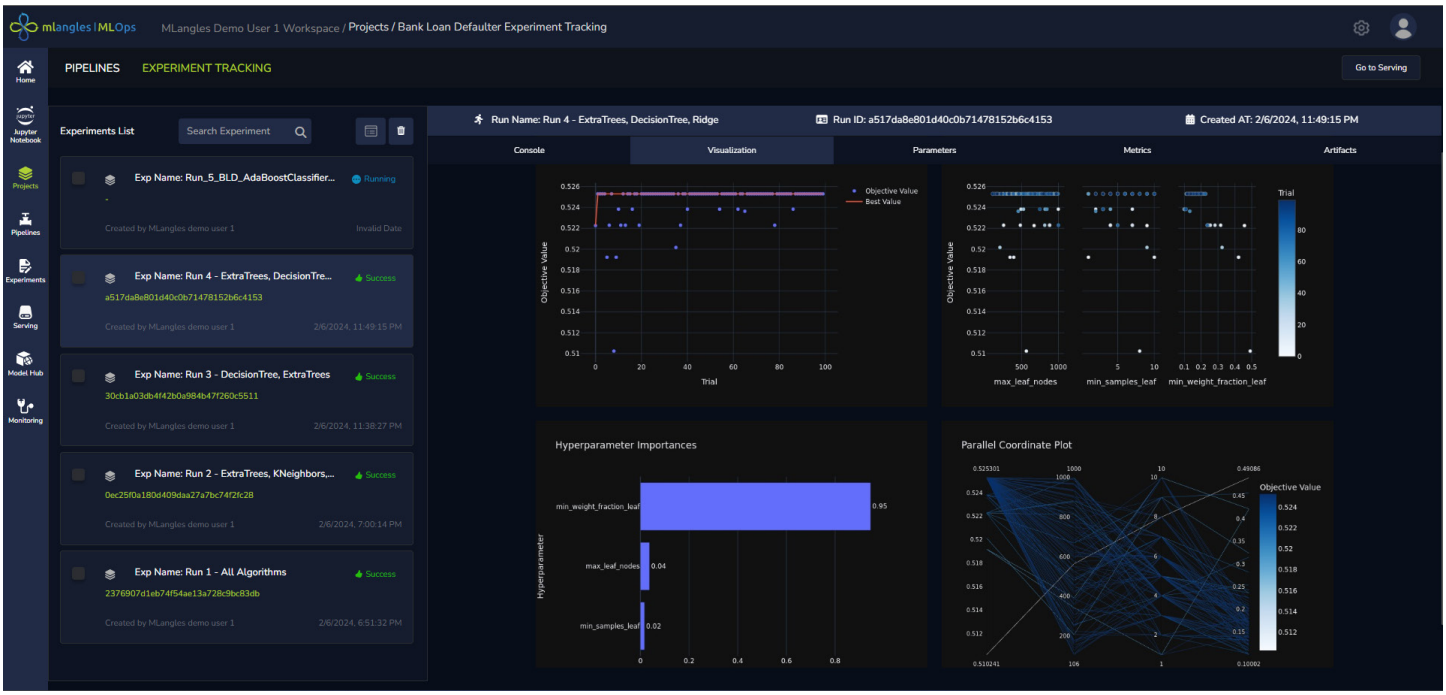


HYPERPARAMETER VISUALIZATION:

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

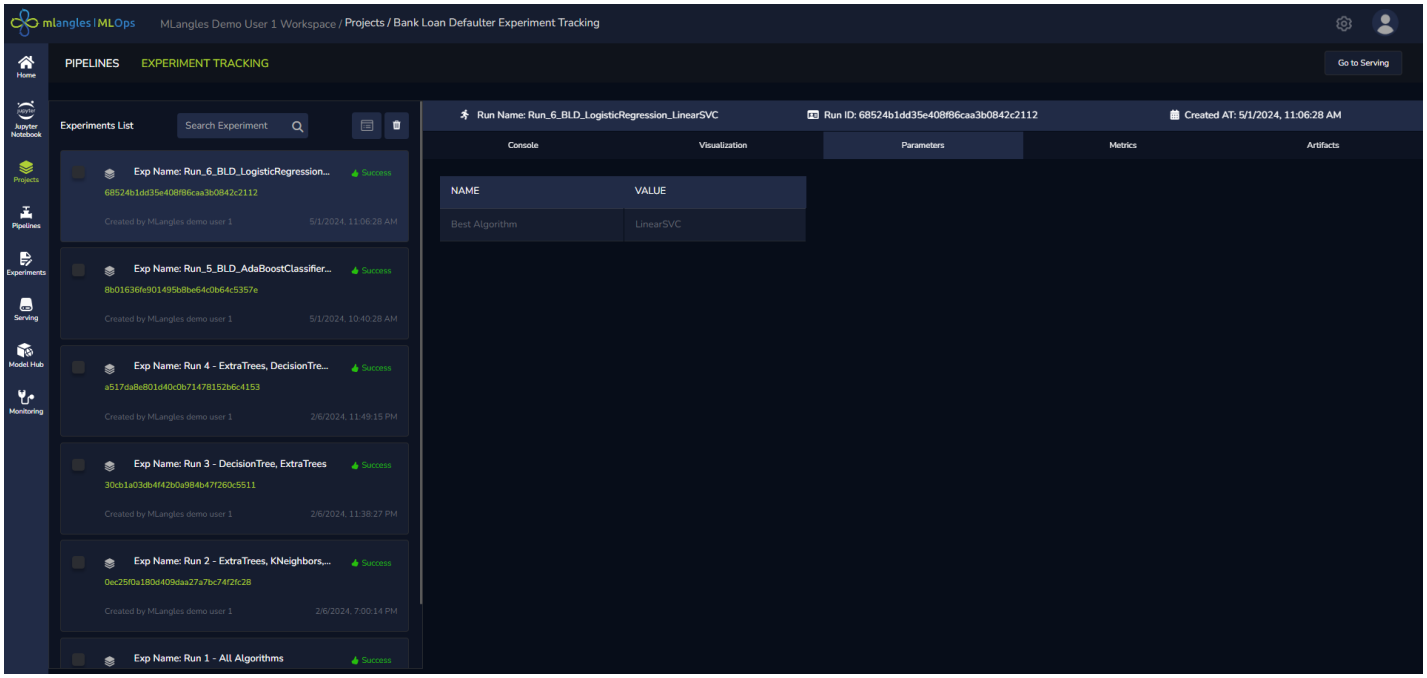
The different types of plots that represent the values of the parameters for each trail are as follows:

- Optimization History Plot: It works by modeling the log-odds of a class as a linear combination of the input features. The logistic function maps these log-odds to probabilities between 0 and 1, allowing for probabilistic interpretation.
- 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



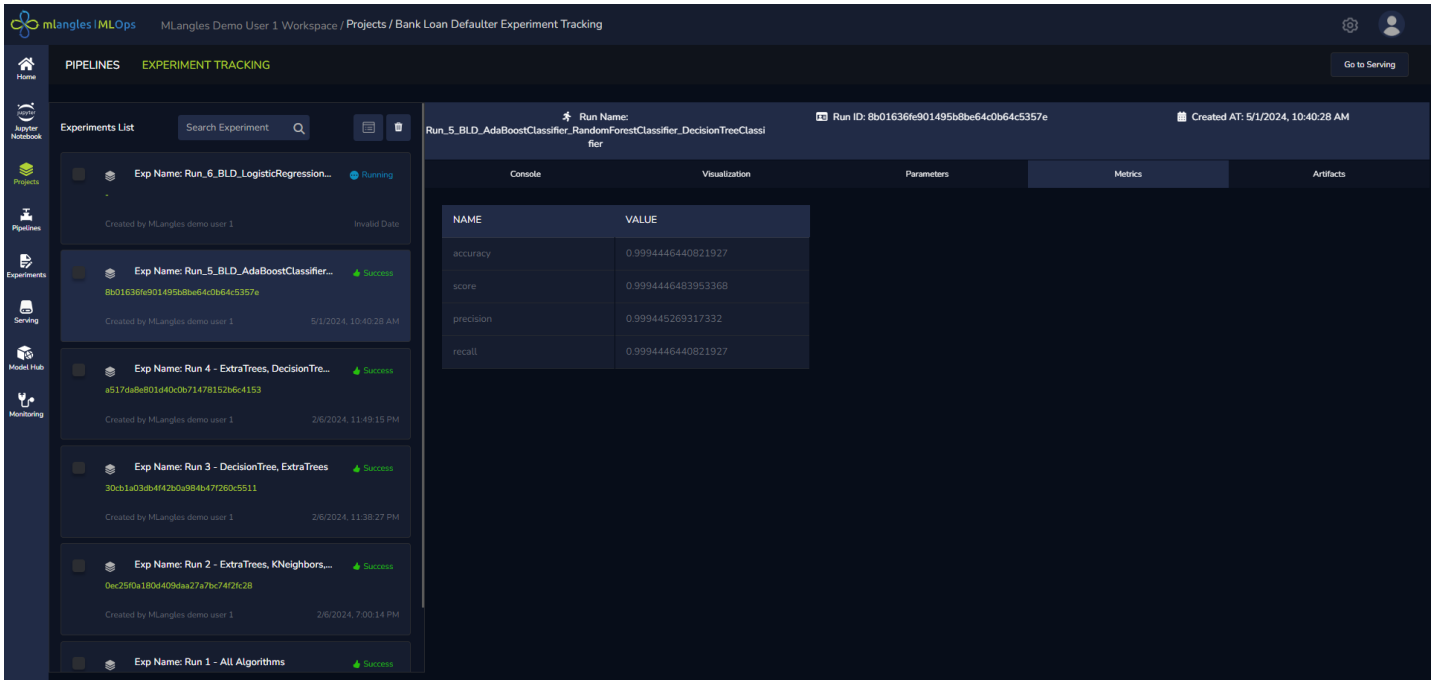
PARAMETERS

Parameters has the best model among all the models that were selected.



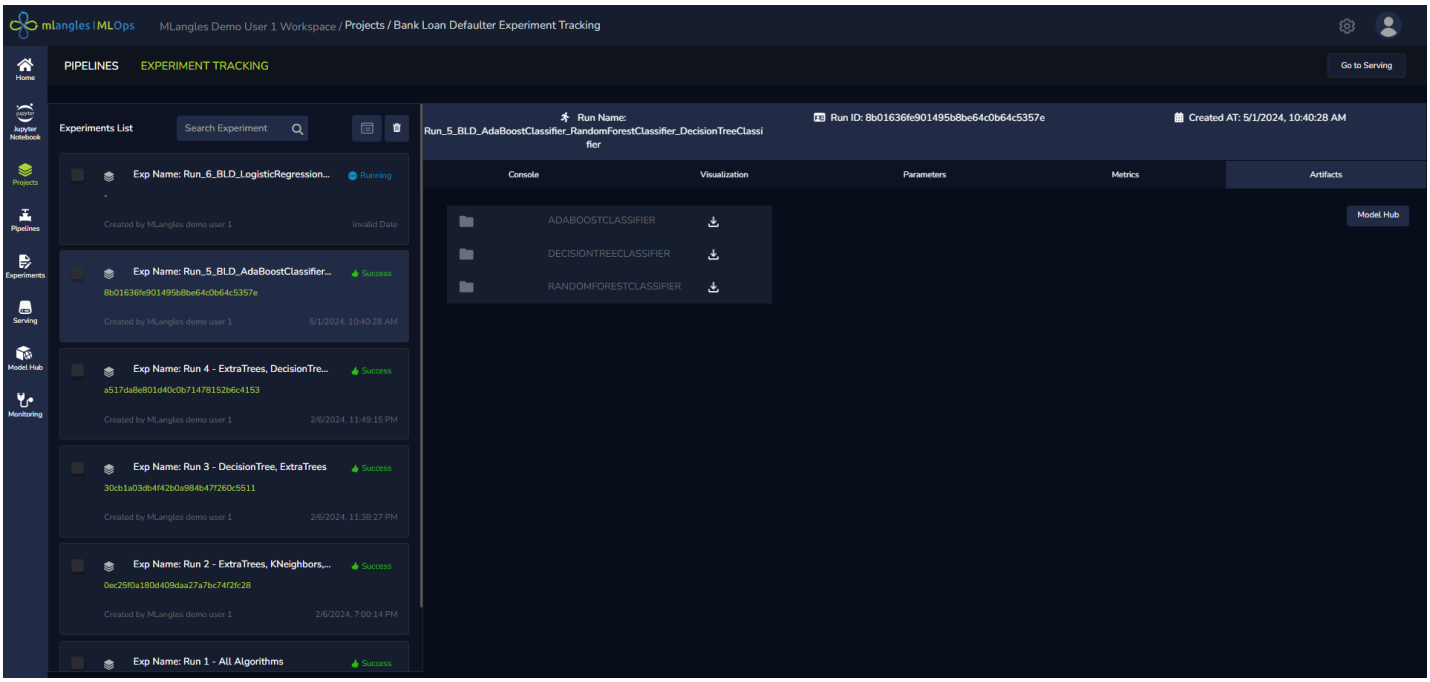
METRICS

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

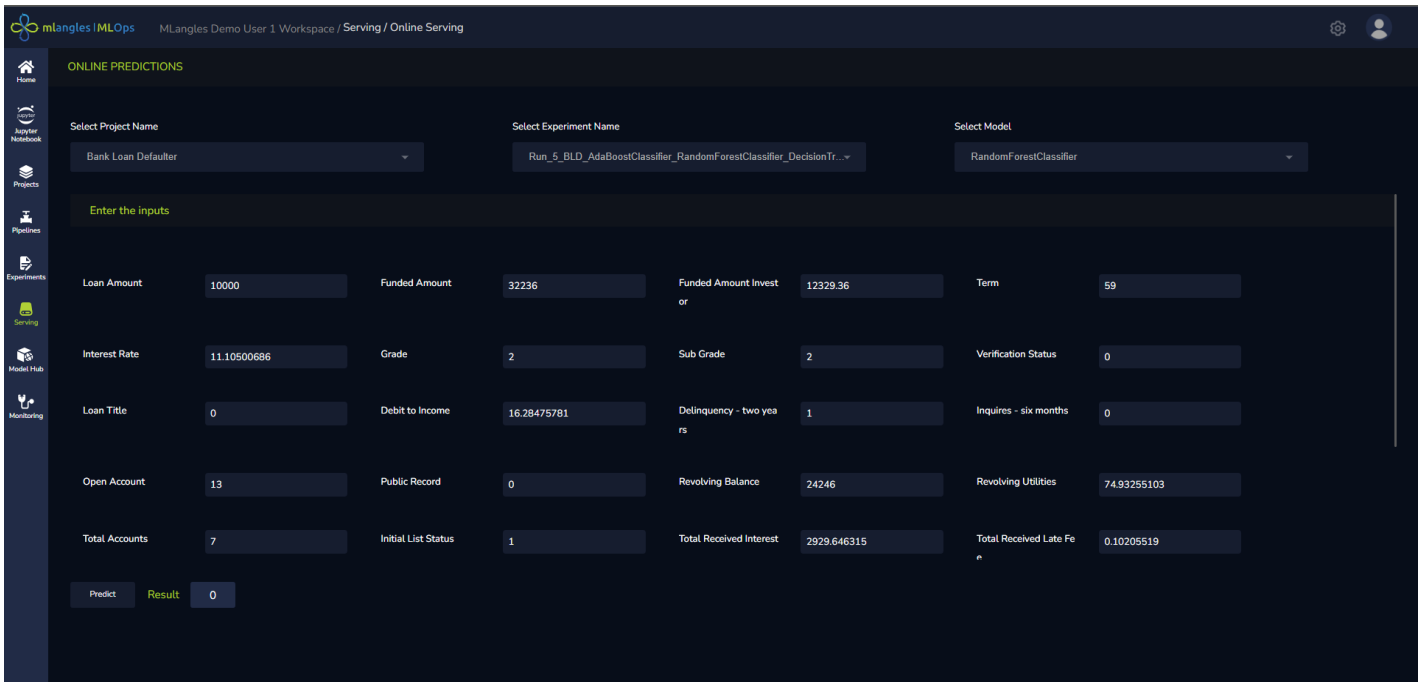
Artifacts is the model that is trained, and we can download the model for further inference.

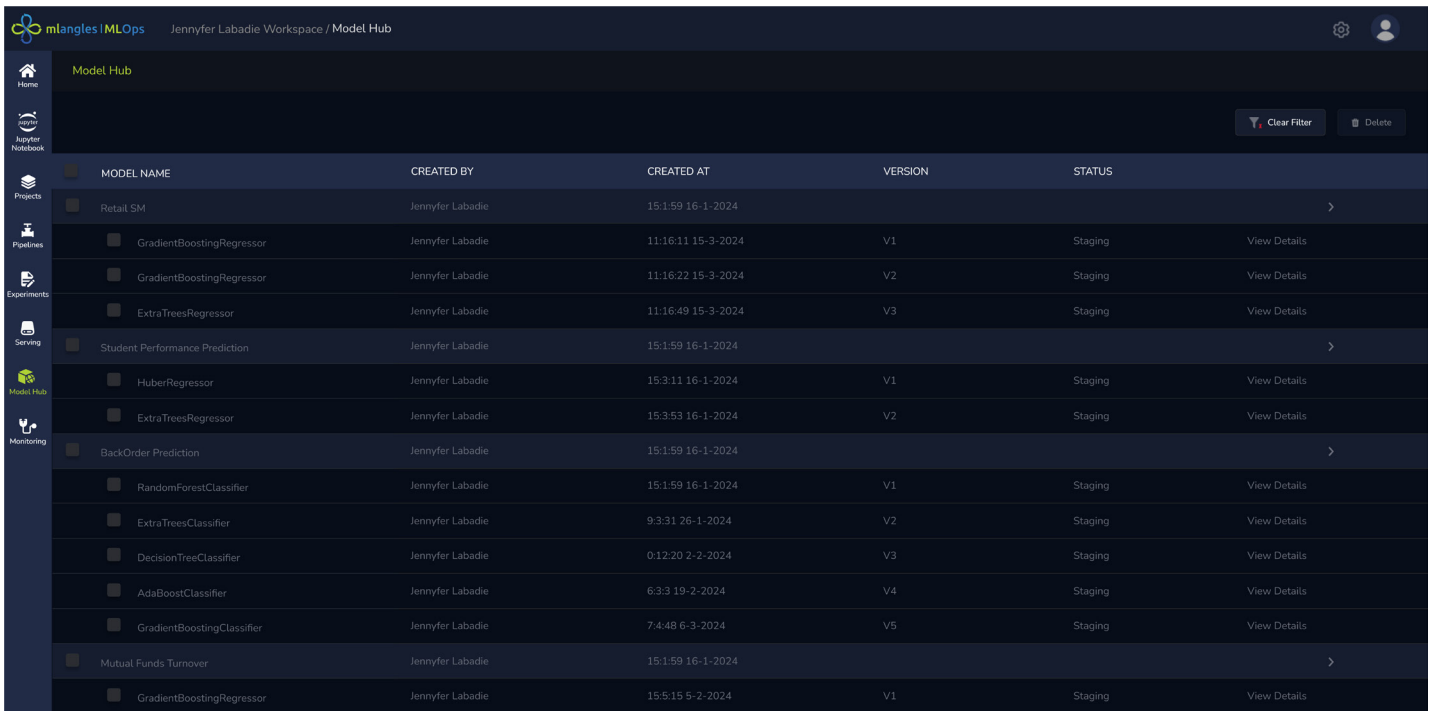


Model Hub: The best model can be pushed to a model hub where it is deployed and exposed as a REST API endpoint. This endpoint allows applications to send new data to the model for making predictions. By integrating the endpoint into applications, users can easily leverage the model's predictive capabilities in their workflows, enabling real-time decision-making based on the model's insights.

Step 3: Serving

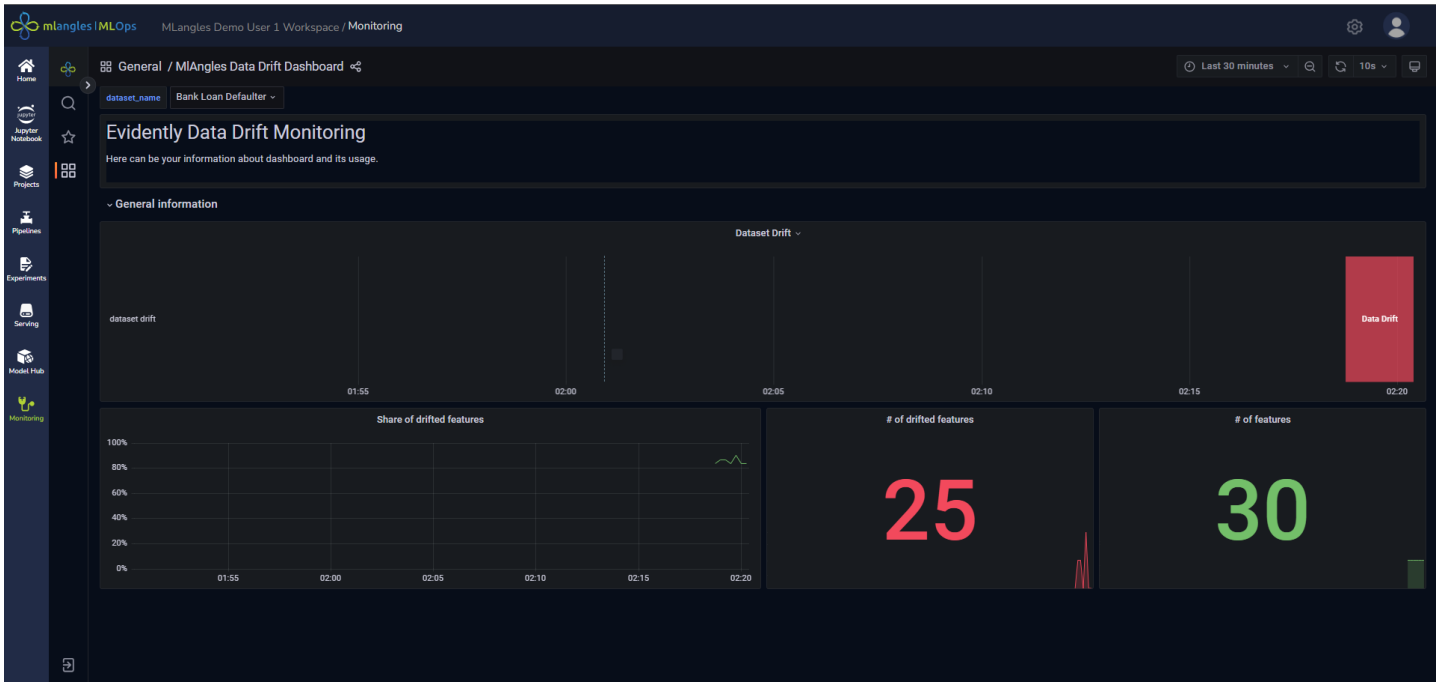
In a serving context, we provide values for each column of data and use this information to predict whether a person has diabetes or not. The model learns from historical data during training and uses this knowledge to make predictions on new data.





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.



Conclusion

In conclusion, this dataset presents an opportunity to create a machine learning model that can identify potential loan defaulters based on a variety of features. By effectively analyzing the dataset and addressing modeling challenges, participants can contribute to developing risk mitigation strategies for banks and financial institutions, ultimately supporting broader economic stability.

To setup Demo

Info.mlangles@cloudangles.com

Visit: www.mlangles.ai