



mlangles
Predictive AI

Mutual Funds Turnover Estimation

Use Case

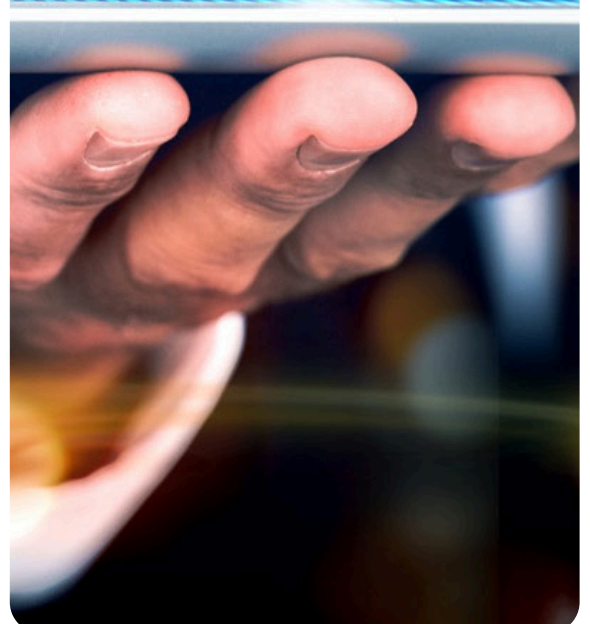




About mlangles Predictive AI

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.



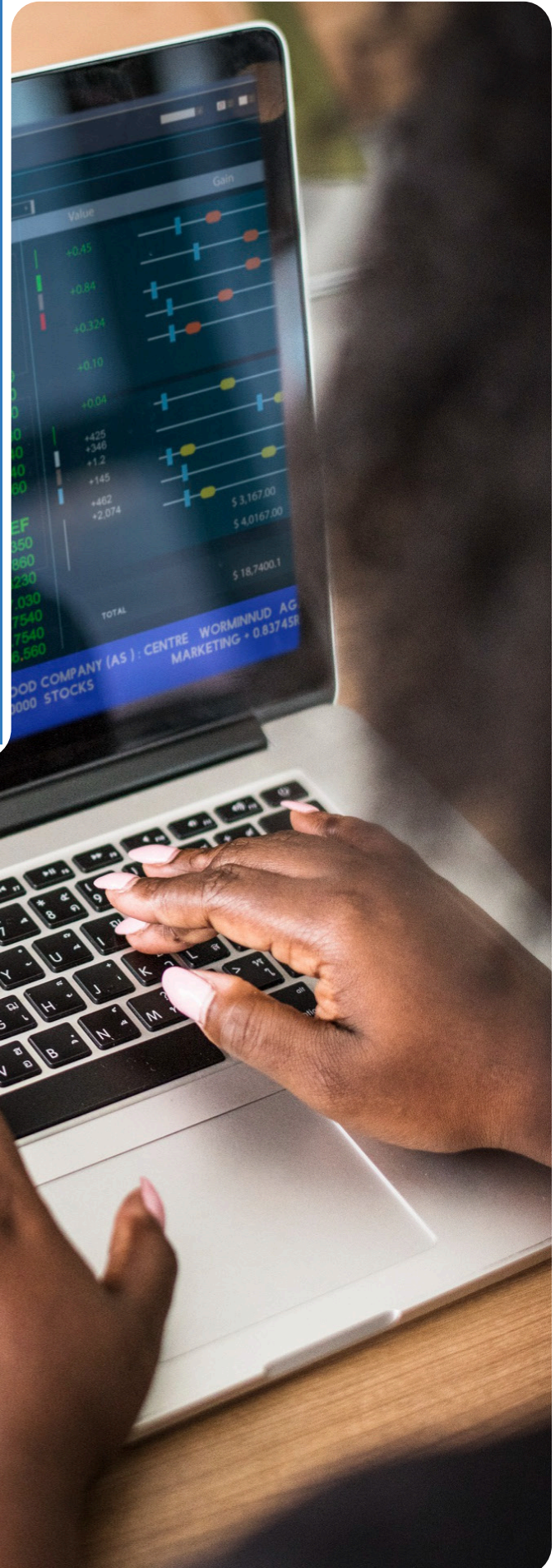


Objective of the Use Case

The objective is to construct a machine learning model capable of accurately predicting the annual turnover of a mutual fund. We aim to empower portfolio managers by providing them with actionable insights for making well-informed investment decisions.

Working of the Use Case

- ➔ The AI problem will be tackled through a phased approach, starting with the **data engineering phase** utilizing the pipeline module.
- ➔ The **modelling process will follow**, with the experiment tracking module aiding in the selection of suitable hyperparameters.
- ➔ Subsequently, the **model will be trained and executed** on the provided dataset.
- ➔ **Predictions generated** by the model will be presented using the serving module.
- ➔ **Continuous monitoring** will be implemented to maintain the accuracy and effectiveness of the model over time.



Overview of Dataset and Use Case



Mutual funds and ETFs (Exchange-Traded Funds) are investment vehicles that pool money from multiple investors to buy a diversified portfolio of stocks, bonds, or other securities. Several features affect the performance of a mutual fund such as underlying investments, asset allocation, expense ratio, market conditions etc. [The objective of our use case is focused on estimating the annual turnover of a mutual fund based on the historical data](#) which refers to the percentage of a mutual fund's assets that are bought or sold within a given year.

The dataset was sourced from Kaggle which in turn had been collected using Yahoo Finance API's. The dataset features can be broken down into -

- ▶ General fund aspects which include total_net_assets, fund family, inception date, etc.
- ▶ Portfolio indicators such as cash, stocks, bonds, sectors, etc.
- ▶ Historical yearly and quarterly returns (e.g. year to_date, 1-year, 3-years, etc.)
- ▶ Financial ratios such as price/ earning, Treynor and Sharpe ratios, alpha, and beta etc.
- ▶ ESG scores

Working of Use Case

Step 1: Pipeline Creation

Extract Data: This step consisted of reading data from source (here s3)

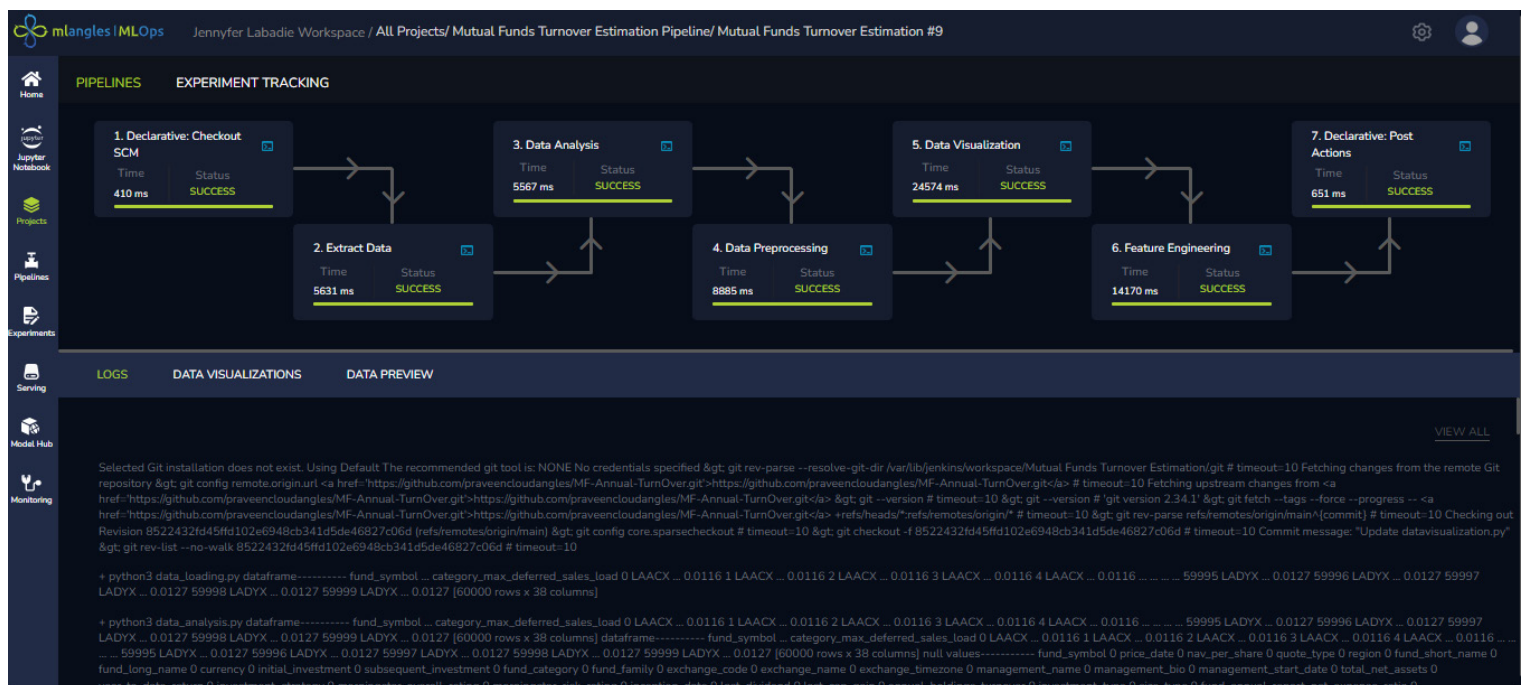
Data Analysis: We perform a closer inspection of the dataset to get an overview of the data as well as get an understanding of the statistical properties of the features within the dataset.

Data Preprocessing: This step consists of preparation of data for model development and feature engineering. We preprocess the data by label encoding categorical data features, fill in missing values and dropping redundant columns from the dataset such as the abbreviated short forms of the mutual fund stocks.

Data Visualization: We display and analyze a number of graphs including boxplot, bar plots and heatmap.

We use these graphs to shortlist features that need to be dropped, understand the distribution of our feature set and identify columns that have outlier datapoints.

Feature Engineering: All feature transformations and generation are performed in this step. A few of the steps that we performed in it are normalization of numerical features, dropping of highly correlated features, treatment of outliers by dropping those data points etc.



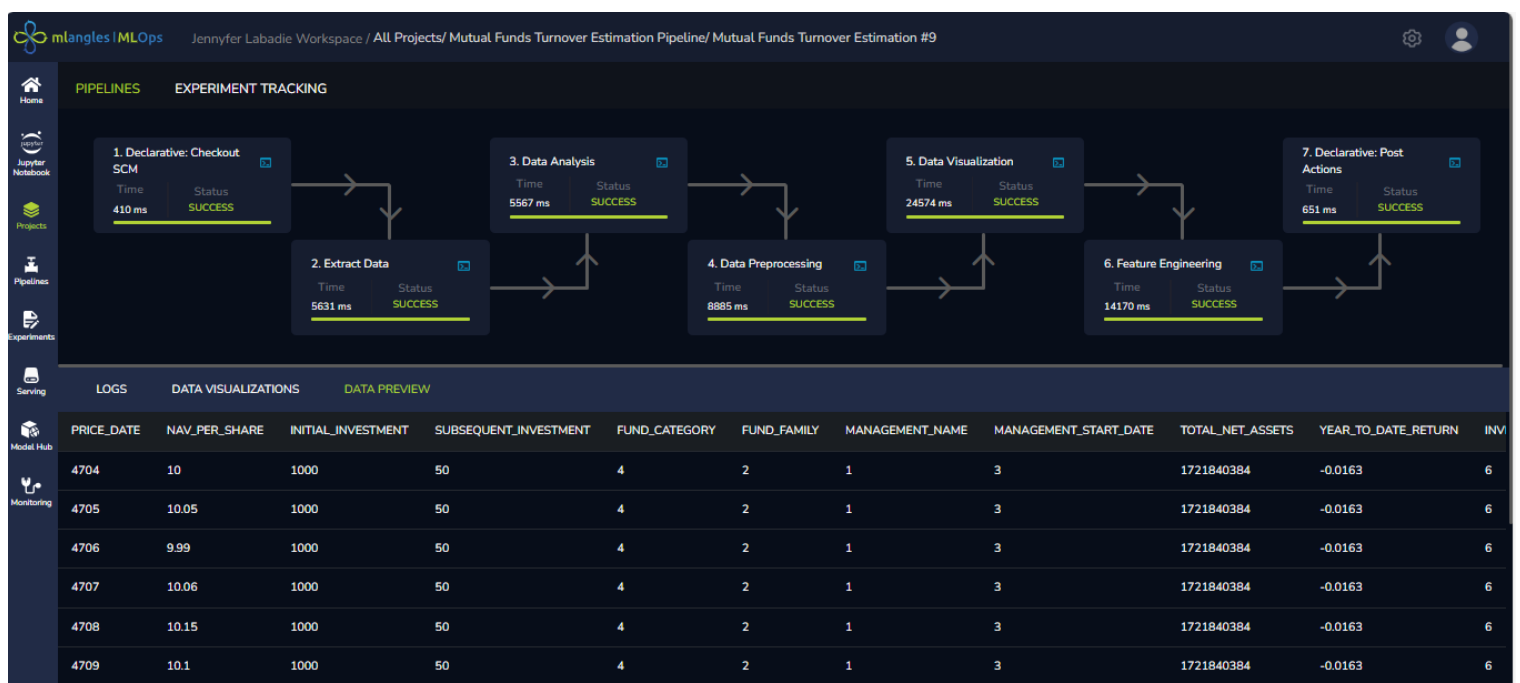
An overview of the data pipeline and steps along with the logs

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.



Histogram plot of nav per share which makes it clear that most data points have nav per share value of 10. Thus, the data would be a more accurate representation of mutual funds which have similar nav per share values. Along with that we also have a heatmap of numerical features which display the correlation amongst features. Since most features have little to no correlation between them we utilize all of these features to build our machine learning model.



The screenshot shows the mlangles IMLops interface for the 'Mutual Funds Turnover Estimation Pipeline'. The pipeline is composed of seven main steps:

- 1. Declarative: Checkout SCM:** Time 410 ms, Status SUCCESS.
- 2. Extract Data:** Time 5631 ms, Status SUCCESS.
- 3. Data Analysis:** Time 5567 ms, Status SUCCESS.
- 4. Data Preprocessing:** Time 8885 ms, Status SUCCESS.
- 5. Data Visualization:** Time 24574 ms, Status SUCCESS.
- 6. Feature Engineering:** Time 14170 ms, Status SUCCESS.
- 7. Declarative: Post Actions:** Time 651 ms, Status SUCCESS.

Below the pipeline, there is a table showing the preview of the data after the pipeline has been run to successful completion.

PRICE_DATE	NAV_PER_SHARE	INITIAL_INVESTMENT	SUBSEQUENT_INVESTMENT	FUND_CATEGORY	FUND_FAMILY	MANAGEMENT_NAME	MANAGEMENT_START_DATE	TOTAL_NET_ASSETS	YEAR_TO_DATE_RETURN	INV
4704	10	1000	50	4	2	1	3	1721840384	-0.0163	6
4705	10.05	1000	50	4	2	1	3	1721840384	-0.0163	6
4706	9.99	1000	50	4	2	1	3	1721840384	-0.0163	6
4707	10.06	1000	50	4	2	1	3	1721840384	-0.0163	6
4708	10.15	1000	50	4	2	1	3	1721840384	-0.0163	6
4709	10.1	1000	50	4	2	1	3	1721840384	-0.0163	6

Preview of the data after the data pipeline has been run to successful completion

Step 2: Experiment Tracking

Multiple experiments were run in which different machine learning models were trained and then tested. Each model's accuracy was then compared. This process was further repeated with different hyperparameters. Key hyperparameters affecting model performance were identified.

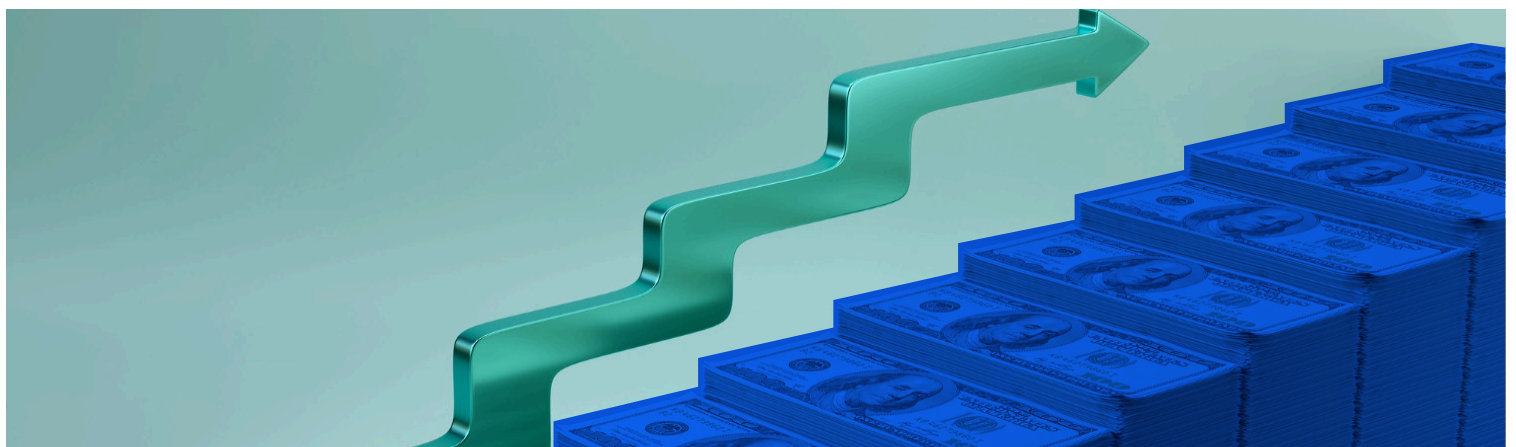
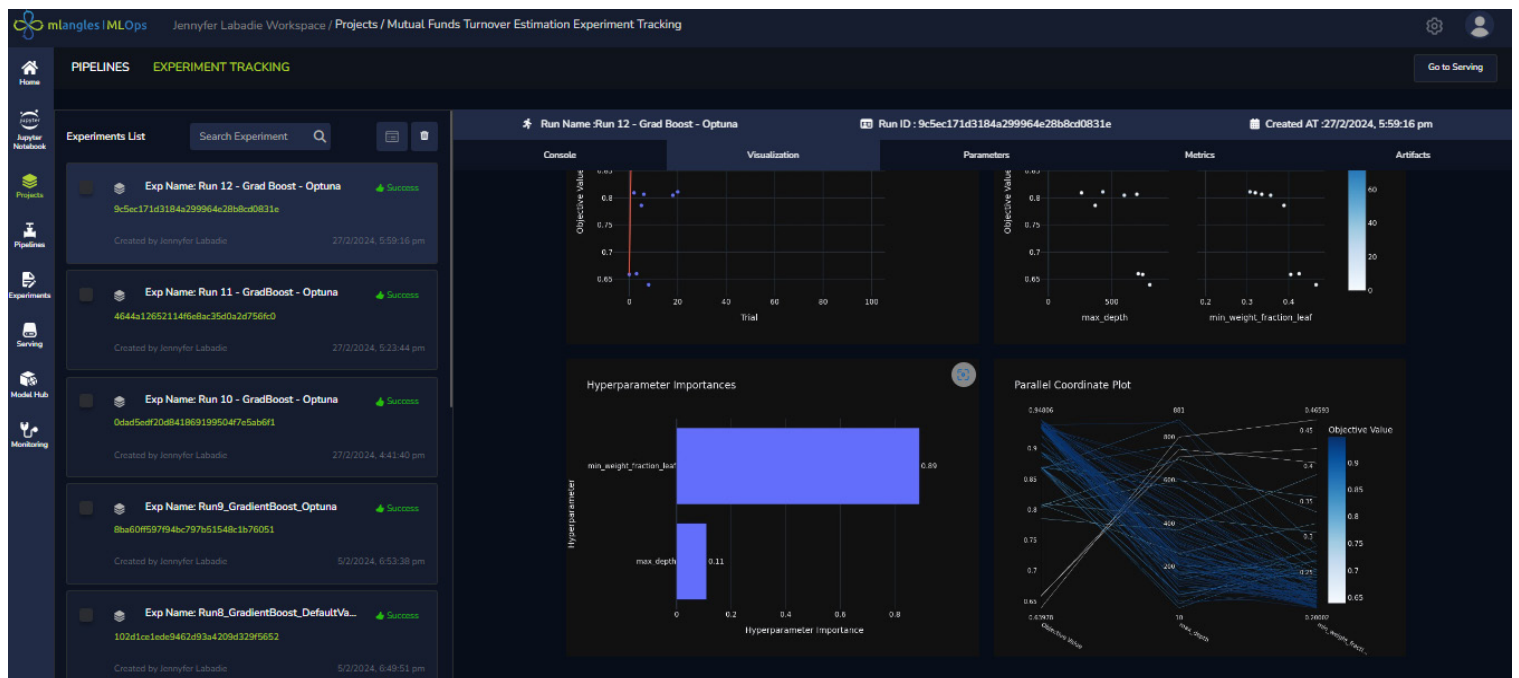
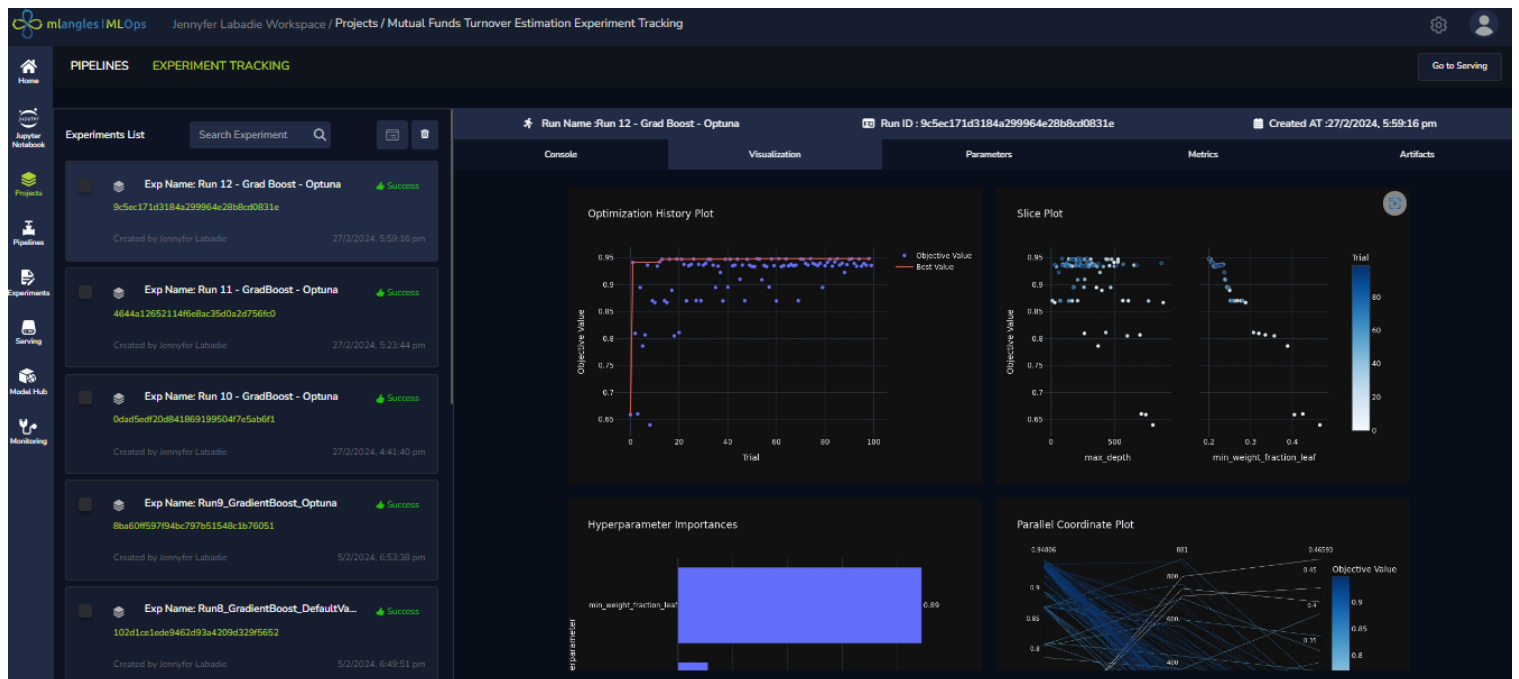
Emphasis was placed on ensemble and tree-based models (adboost, gradient boosting , decision tree etc.). Bayesian ridge with default parameters was used as the threshold accuracy (85 percent accuracy).

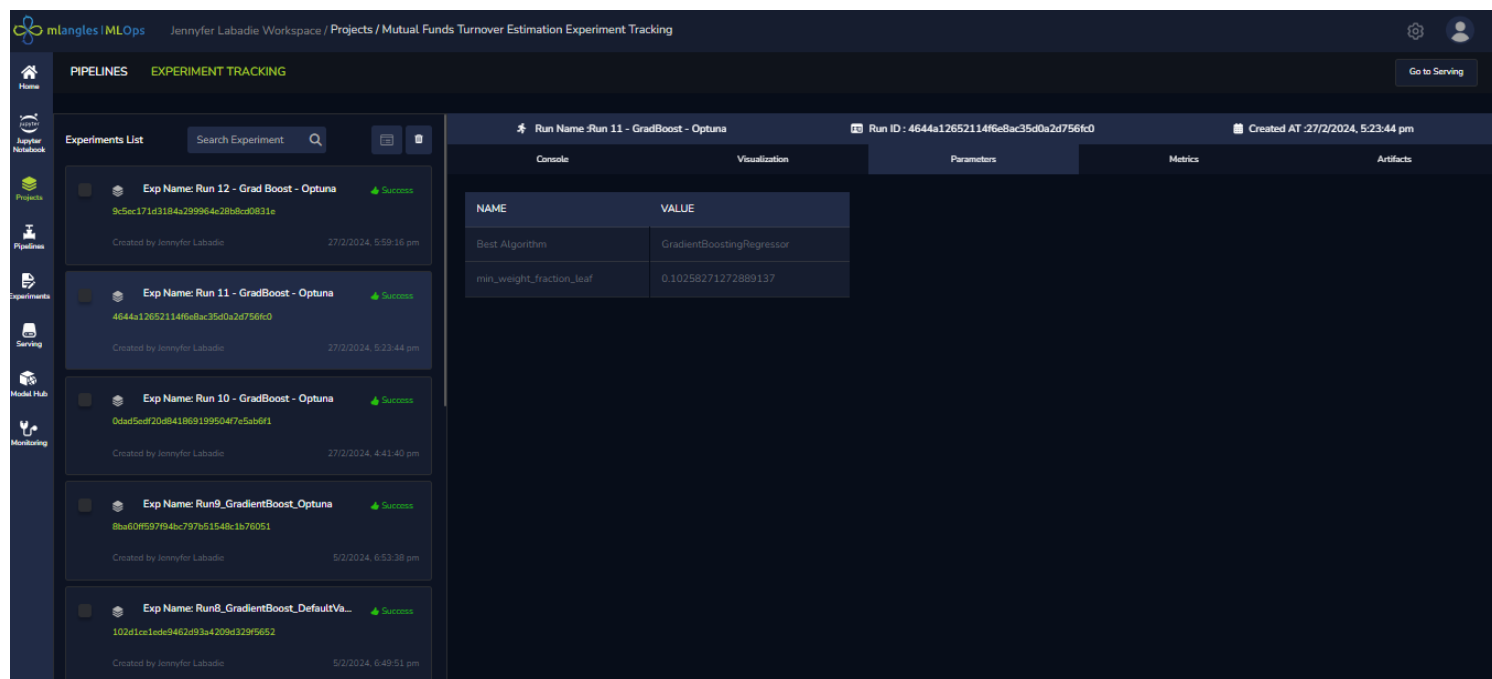
It was observed that Gradient Boosting Algorithms based models performed the best which included high accuracy on both training and validation data subsets.

To illustrate, for a gradient boosted model we tested the min_weight_fraction_leaf from 0.1 to 0.5. The minimum weighted fraction of the total of weights (of all the input samples) required to be at a leaf node. As is evident from the graph, we prefer to keep this param between 0.1 to 0.3 for maximum accuracy.



mlangles IMLops Jennyfer Labadie Workspace / Projects / Mutual Funds Turnover Estimation Experiment Tracking						
PIPELINES EXPERIMENT TRACKING		+ New Run Run Configuration Clear Filter Delete				
RUN ID	RUN NAME	STATUS	CREATED BY	START TIME	END TIME	
9c5ec171d3184a299964e28b8cd0831e	Run 12 - Grad Boost - Optuna	Success	Jennyfer Labadie	27/2/2024, 5:59:16 pm	27/2/2024, 6:06:56 pm	
4644a12652114f6e8ac35da2d756e0	Run 11 - GradBoost - Optuna	Success	Jennyfer Labadie	27/2/2024, 5:23:44 pm	27/2/2024, 5:28:28 pm	
0dad5ed720d8418691995047e5ab6f1	Run 10 - GradBoost - Optuna	Success	Jennyfer Labadie	27/2/2024, 4:41:40 pm	27/2/2024, 5:12:36 pm	
8ba60f597f94bc797b515481b76051	Run9_GradientBoost_Optuna	Success	Jennyfer Labadie	5/2/2024, 6:53:38 pm	5/2/2024, 6:55:01 pm	
102d1ce1ede9462d93a4209d329f5652	Run8_GradientBoost_DefaultValues	Success	Jennyfer Labadie	5/2/2024, 6:49:51 pm	5/2/2024, 6:50:07 pm	
c34054d022794e5196220c8ce4d5ff4c	Run7_DecisionTree_Optuna	Success	Jennyfer Labadie	5/2/2024, 6:34:53 pm	5/2/2024, 6:38:43 pm	
1313faa081614ad7bfab3fe942da520c	Run4_adaboost_optuna	Success	Jennyfer Labadie	5/2/2024, 6:23:40 pm	5/2/2024, 6:25:00 pm	
59bd0f8f0de947ada0363cd2471d48f7	Run3_DecisionTree_DefaultParams	Success	Jennyfer Labadie	5/2/2024, 6:15:43 pm	5/2/2024, 6:15:52 pm	
45921934fb644615be070b7f0b5a2b8	Run2_BayesianRidge_DefaultParams	Success	Jennyfer Labadie	5/2/2024, 6:13:40 pm	5/2/2024, 6:13:48 pm	
432481933774347b4c6b8189ad8887	Run1_adaboost_defaultParams	Success	Jennyfer Labadie	5/2/2024, 6:13:11 pm	5/2/2024, 6:13:21 pm	



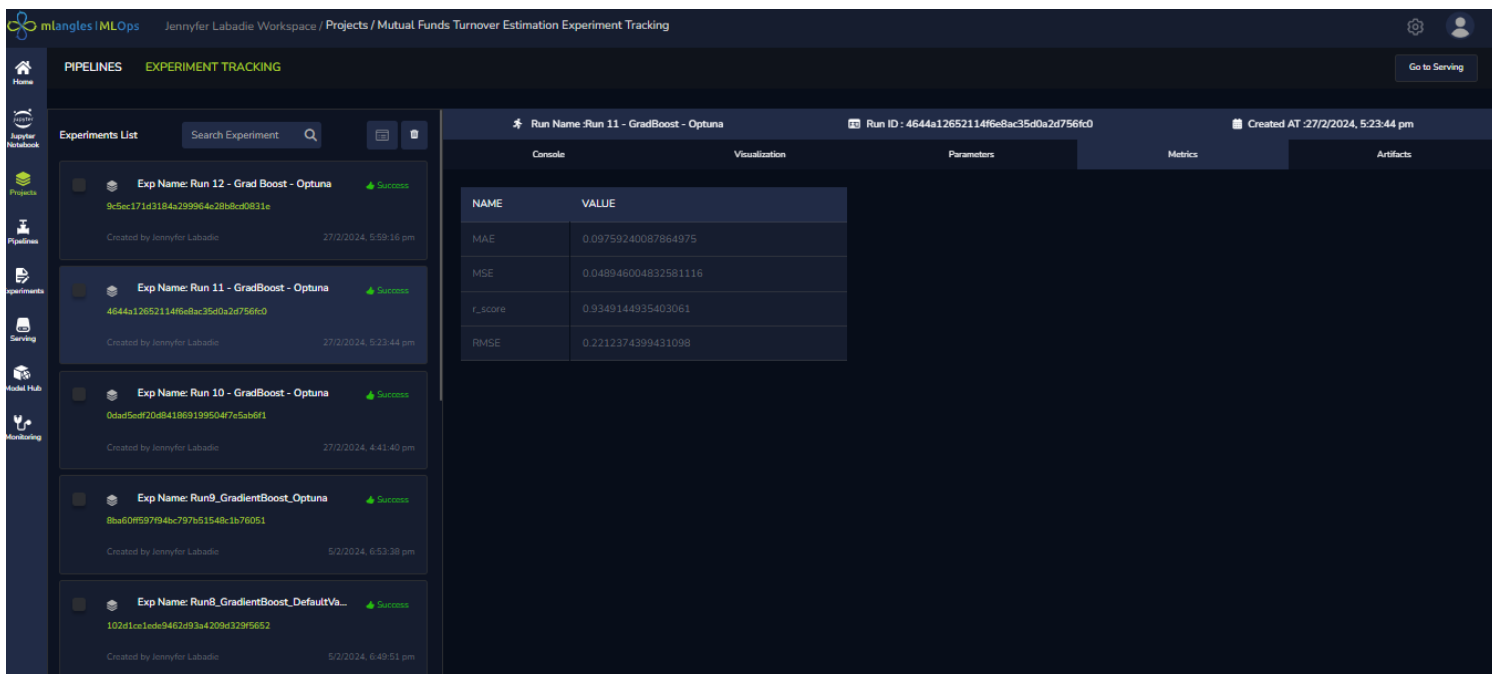


Experiments List

Exp Name	Status	Created by	Created At
Exp Name: Run 12 - Grad Boost - Optuna	Success	Jennyfer Labadie	27/2/2024, 5:59:16 pm
Exp Name: Run 11 - GradBoost - Optuna	Success	Jennyfer Labadie	27/2/2024, 5:23:44 pm
Exp Name: Run 10 - GradBoost - Optuna	Success	Jennyfer Labadie	27/2/2024, 4:41:40 pm
Exp Name: Run9_GradientBoost_Optuna	Success	Jennyfer Labadie	5/2/2024, 6:53:38 pm
Exp Name: Run8_GradientBoost_DefaultVa...	Success	Jennyfer Labadie	5/2/2024, 6:49:51 pm

Run Name: Run 11 - GradBoost - Optuna
Run ID: 4644a12652114f6e8ac35d0a2d756fc0
Created AT: 27/2/2024, 5:23:44 pm

NAME	VALUE
Best Algorithm	GradientBoostingRegressor
min_weight_fraction_leaf	0.10258271272889137



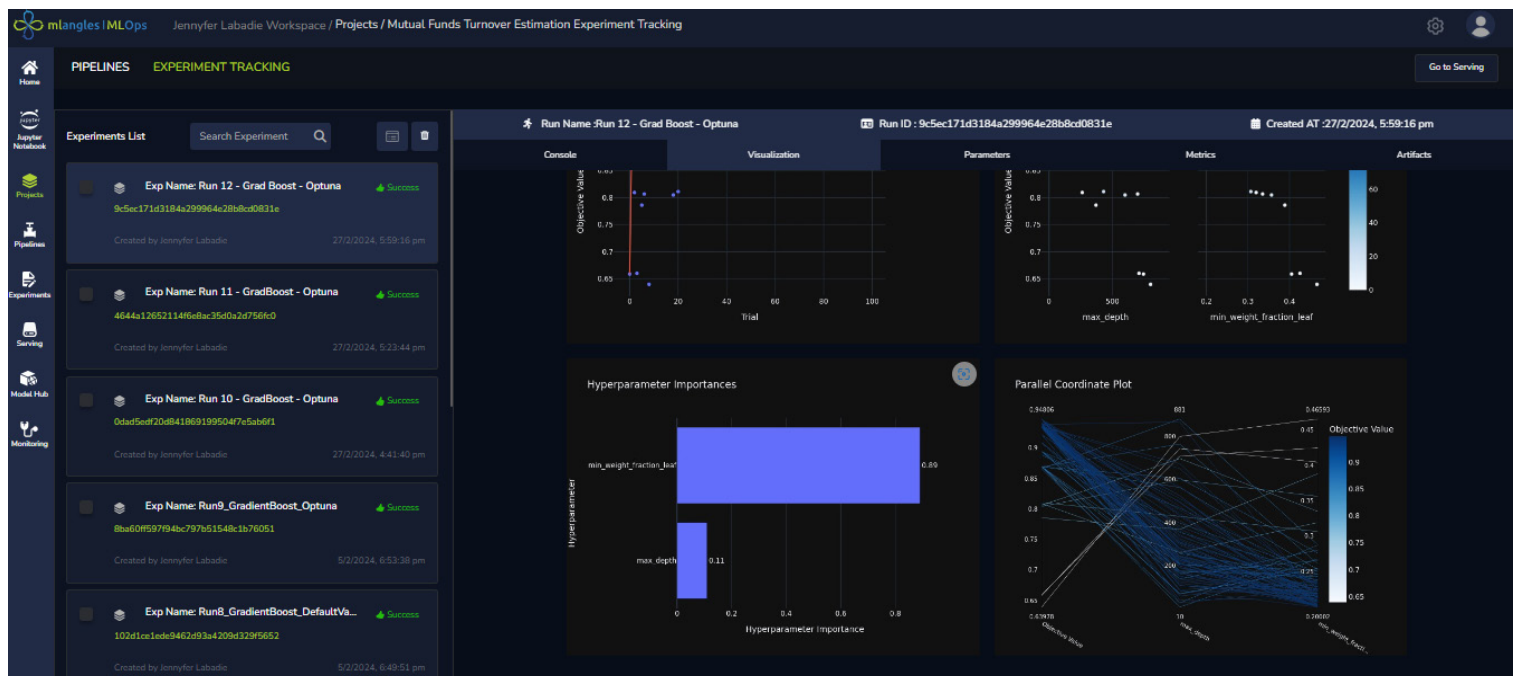
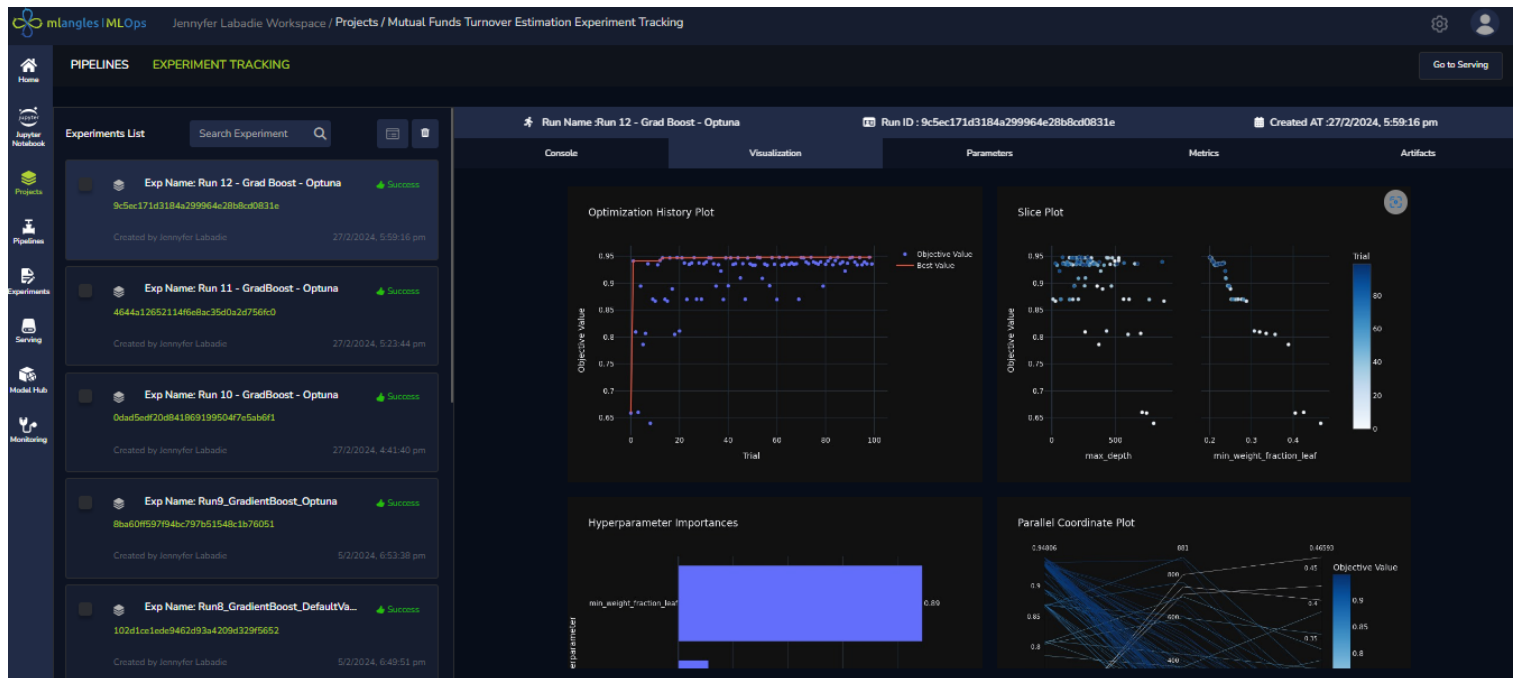
Experiments List

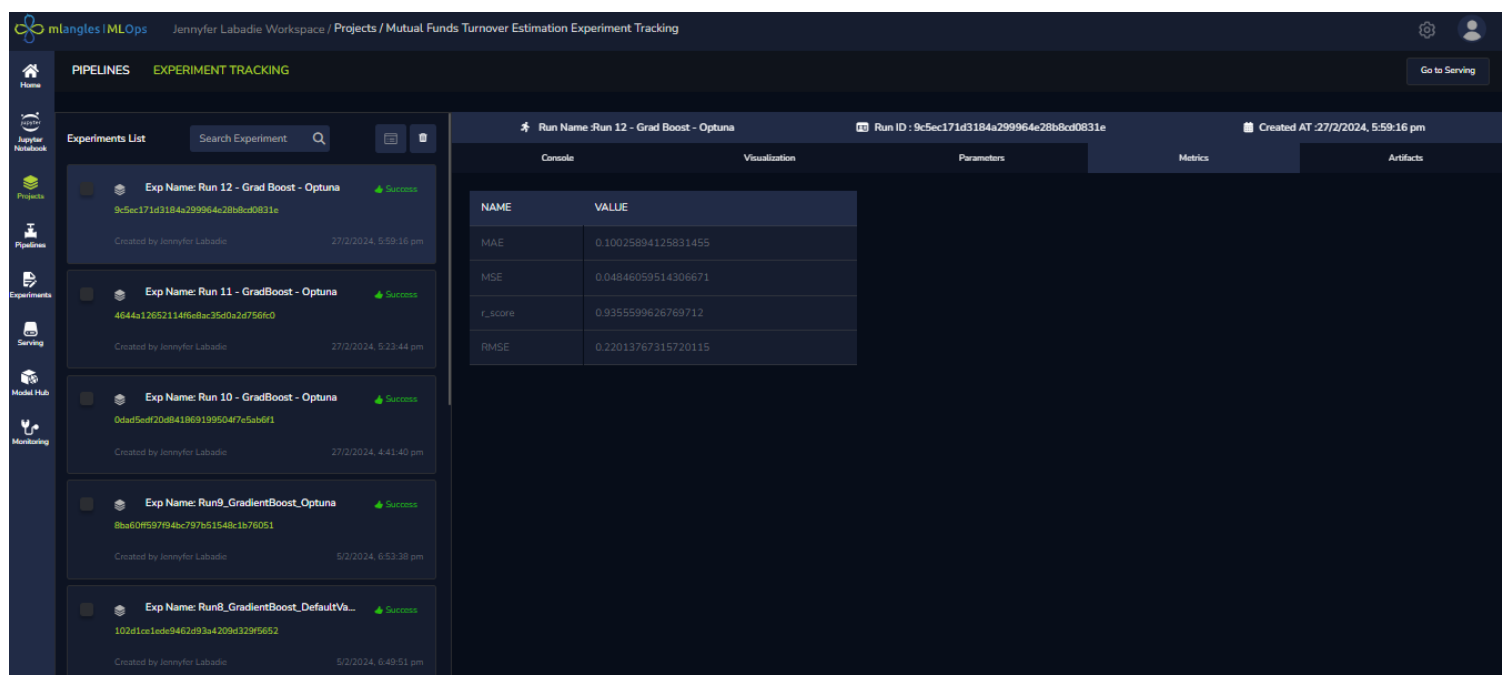
Exp Name	Status	Created by	Created At
Exp Name: Run 12 - Grad Boost - Optuna	Success	Jennyfer Labadie	27/2/2024, 5:59:16 pm
Exp Name: Run 11 - GradBoost - Optuna	Success	Jennyfer Labadie	27/2/2024, 5:23:44 pm
Exp Name: Run 10 - GradBoost - Optuna	Success	Jennyfer Labadie	27/2/2024, 4:41:40 pm
Exp Name: Run9_GradientBoost_Optuna	Success	Jennyfer Labadie	5/2/2024, 6:53:38 pm
Exp Name: Run8_GradientBoost_DefaultVa...	Success	Jennyfer Labadie	5/2/2024, 6:49:51 pm

Run Name: Run 11 - GradBoost - Optuna
Run ID: 4644a12652114f6e8ac35d0a2d756fc0
Created AT: 27/2/2024, 5:23:44 pm

NAME	VALUE
MAE	0.09759240087864975
MSE	0.048946004832581116
r_score	0.9349144935403061
RMSE	0.2212374399431098

We also tested the combination of two different hyper-params, min_weight_fraction_leaf (0.1 -0.5) and max_depth (10 - 900) . The highest accuracy is achieved when min_weight_fraction_leaf optimally lies between 0.1-0.25 while at the same time max_depth lies between (10 - 400). The larger range for max_depth points to the fact that it has less effect on improving accuracy and this is observed in the hyper-param importance graph where min_weight_fraction_leaf has much higher weightage.





The screenshot displays the 'EXPERIMENT TRACKING' section of the mlangles IMLOps interface. On the left, a list of experiments is shown, including 'Run 12 - Grad Boost - Optuna', 'Run 11 - GradBoost - Optuna', 'Run 10 - GradBoost - Optuna', 'Run9_GradientBoost_Optuna', and 'Run8_GradientBoost_DefaultVa...'. The right panel provides details for 'Run 12 - Grad Boost - Optuna', showing a 'Console' tab with a table of metrics.

NAME	VALUE
MAE	0.10025894125831455
MSE	0.04846059514306671
r_score	0.9355599626769712
RMSE	0.22013767315720115

The Gradient Boosting model with max_depth 213 and min_weight_fraction_leaf 0.20 had an R-Square value of 0.9 with root mean square error of 0.220. This was selected as one of our shortlisted models.

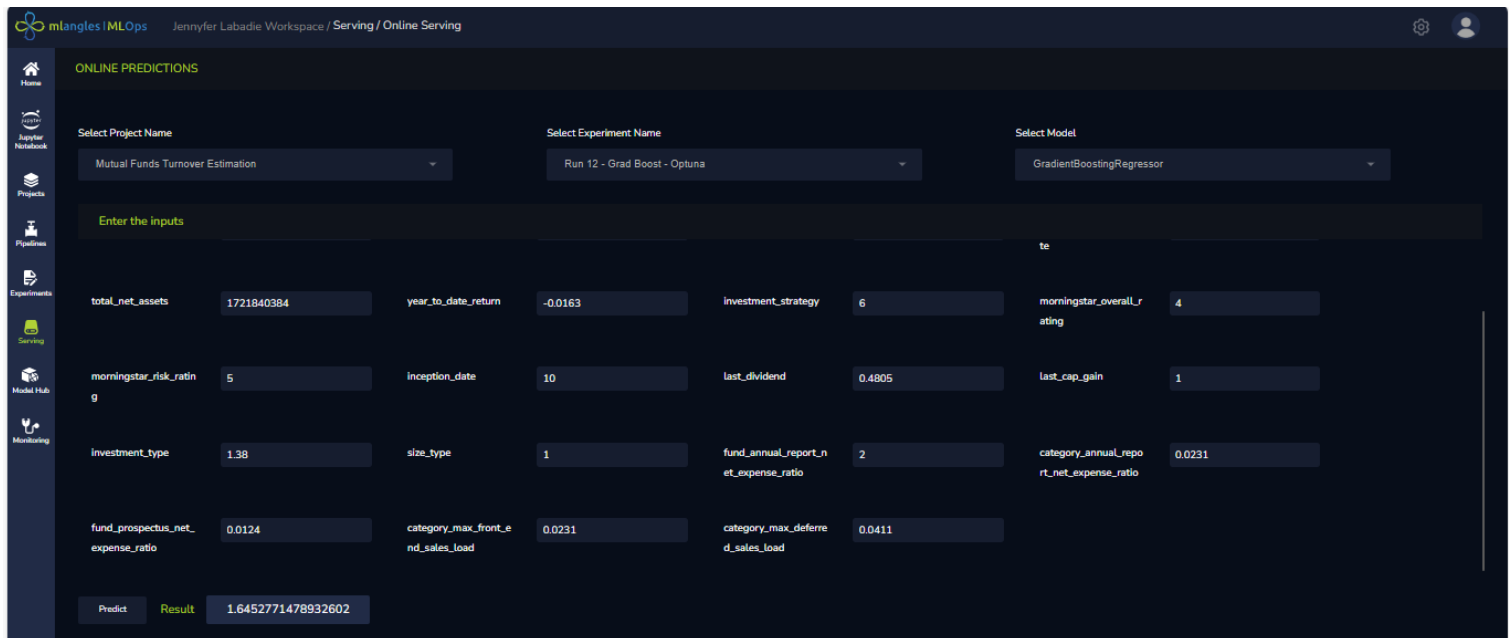


Model Versioning:

- ▶ 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.

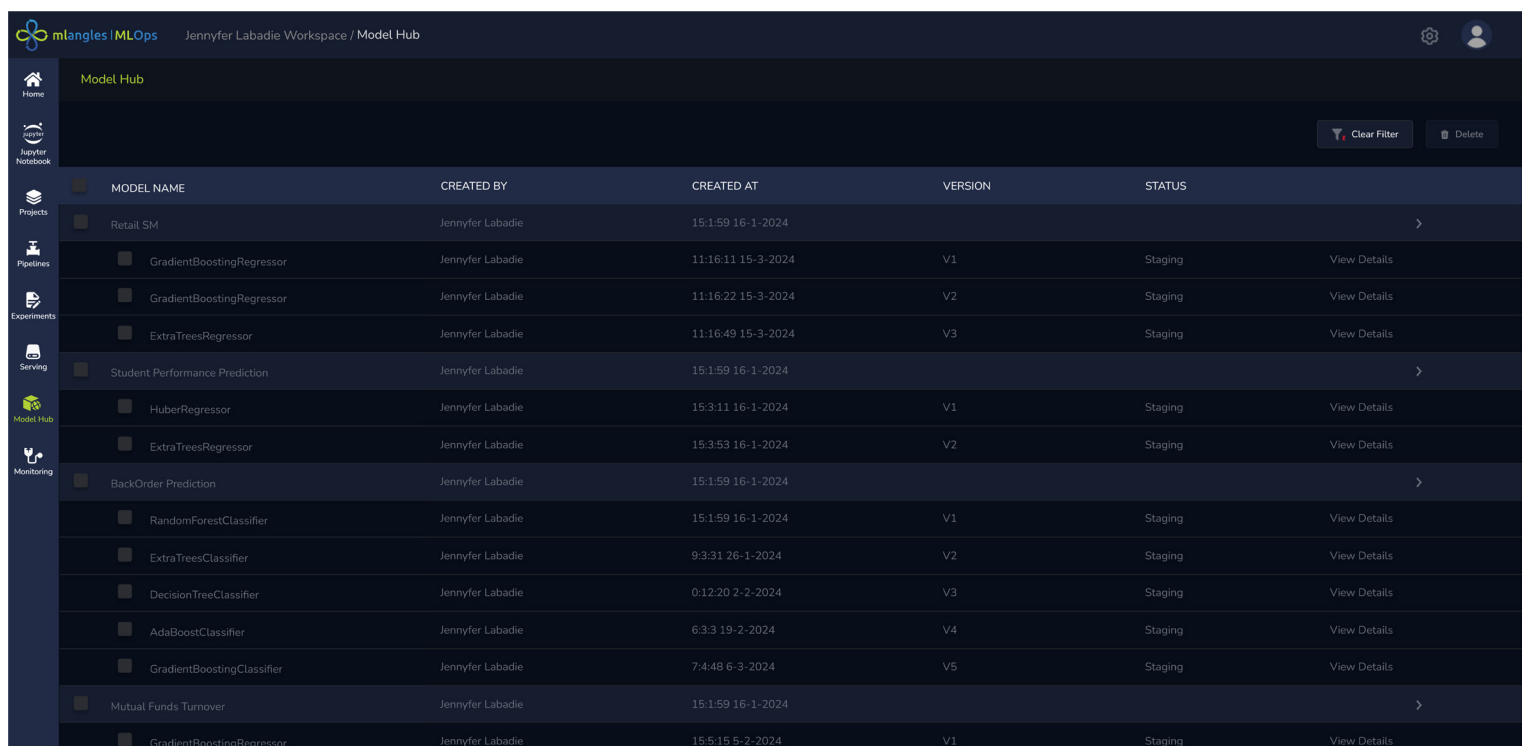
Step 3: Serving

We performed a sanity check on our model using an isolated data point. Thus, the predicted annual holdings turnover, which is the percentage rate at which a fund replaces its investment holdings on an annual basis is estimated to be 1.6. Since the turnover percentage rate in this case is positive, the corresponding mutual fund can possibly be seen as a viable investment.



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.



MODEL NAME	CREATED BY	CREATED AT	VERSION	STATUS
Retail SM	Jennyfer Labadie	15:1:59 16-1-2024		
GradientBoostingRegressor	Jennyfer Labadie	11:16:11 15-3-2024	V1	Staging
GradientBoostingRegressor	Jennyfer Labadie	11:16:22 15-3-2024	V2	Staging
ExtraTreesRegressor	Jennyfer Labadie	11:16:49 15-3-2024	V3	Staging
Student Performance Prediction	Jennyfer Labadie	15:1:59 16-1-2024		
HuberRegressor	Jennyfer Labadie	15:3:11 16-1-2024	V1	Staging
ExtraTreesRegressor	Jennyfer Labadie	15:3:53 16-1-2024	V2	Staging
BackOrder Prediction	Jennyfer Labadie	15:1:59 16-1-2024		
RandomForestClassifier	Jennyfer Labadie	15:1:59 16-1-2024	V1	Staging
ExtraTreesClassifier	Jennyfer Labadie	9:3:31 26-1-2024	V2	Staging
DecisionTreeClassifier	Jennyfer Labadie	0:12:20 2-2-2024	V3	Staging
AdaBoostClassifier	Jennyfer Labadie	6:3:3 19-2-2024	V4	Staging
GradientBoostingClassifier	Jennyfer Labadie	7:4:48 6-3-2024	V5	Staging
Mutual Funds Turnover	Jennyfer Labadie	15:1:59 16-1-2024		
GradientBoostingRegressor	Jennyfer Labadie	15:5:15 5-2-2024	V1	Staging

Step 4: Monitoring

Data drift refers to the phenomenon where the statistical properties of the data change over time in a deployed machine learning model. This could be due to changes in the underlying data distribution, data collection process, or external factors influencing the data. When data drift occurs, the relationships between features and the target variable may change, impacting the model's performance and reliability. The share of drifted features refers to the proportion of features in the dataset that have experienced a significant change or drift in their statistical properties. As these features undergo drift, their relationships with the target variable may become less relevant or even misleading, leading to decreased model accuracy and effectiveness.

Therefore, monitoring and addressing data drift are essential to maintain the model's performance and ensure its continued relevance in production environments.

The monitoring screen displays that 23 features have drifted out of the 25 features that we had utilized for our model. Financial indicators are heavily dependent on several extraneous factors and tumultuous, thus leading to increased possibilities of data drift and shift in statistical properties. This leads to a loss of model efficacy and requires continuous monitoring and retraining of the model, leading to ever increasing robustness and avoidance of systemic bias.



Conclusion

In conclusion, training tree-based ensemble models like Gradient Boosting against our dataset has yielded promising results in estimating annual holdings turnover percentage for any mutual fund with a high degree of accuracy. This comes with a caveat that the data and model is continuously monitored for drift and adequate steps of analyzing and retraining are taken upon drift detection.

To setup Demo

Info.mlangles@cloudangles.com 

Visit: www.mlangles.ai