Creating a Machine Learning Pipeline to Evaluate Employee Attrition

Presented by: Apoorv Anand, Ashmita Mukherjee, Sivaram Mandava

Prepared for: The University of Chicago MS in Applied Data Science, Machine Learning Operations Final Project

Date: August 15, 2024

Agenda

01

EDA + Preprocessing

Dataset overview with exploratory insights + cleaning

02

Pipeline + Modeling

Our Pipeline and Model choice after an initial AutoML run

03

Deployment

How we deployed for inference using a Databricks workflow along with endpointing

04

Monitoring + Dataset Change

How we deployed model monitoring and tracked the synthetic test data drift

We chose an employee attrition dataset where we predict churn based on various features

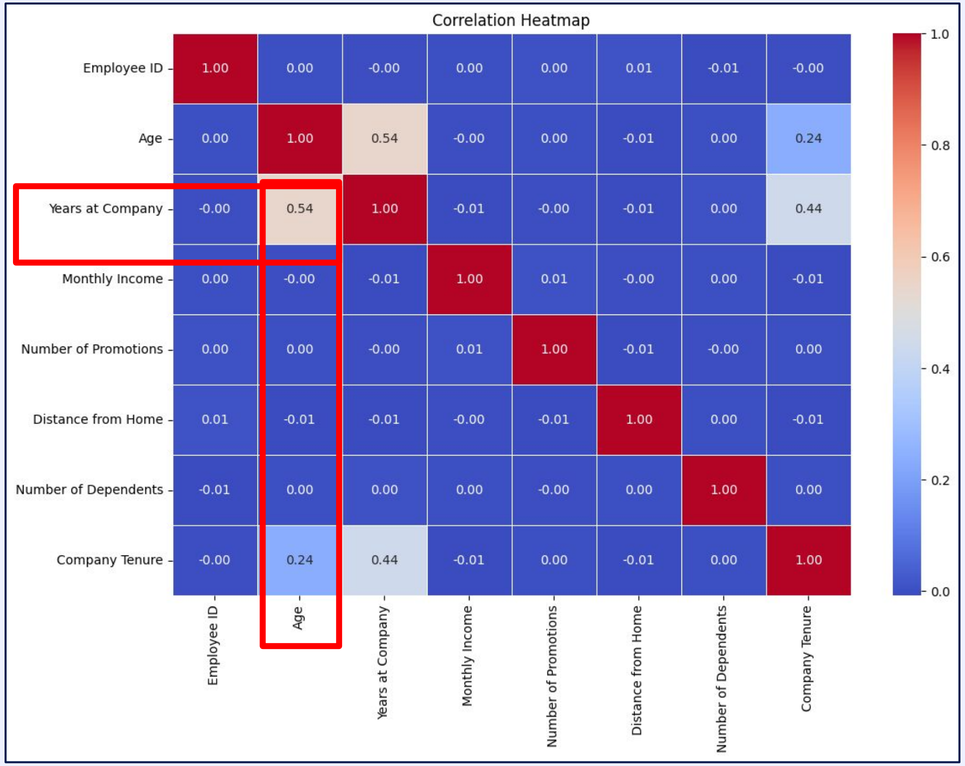
Employee Attrition Classification Dataset (from Kaggle)

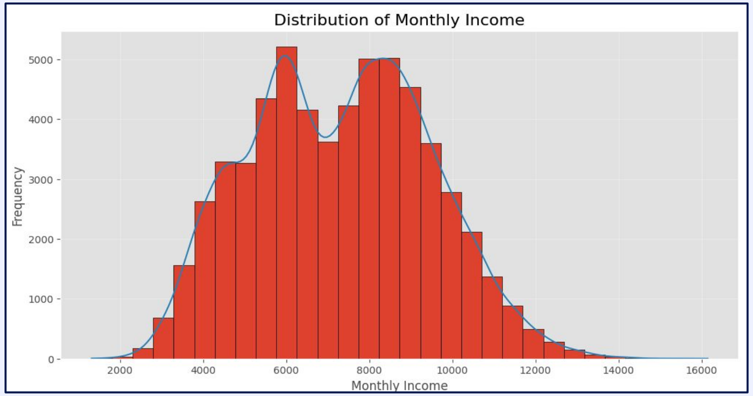
- Simulated dataset designed for the analysis and prediction of employee attrition
- ~75K samples split into train and test sets
- Primary key of employee id
- Useful features such as:
    - **Numeric**: age, years at company, distance from home
    - **Categorical:** gender, job role, job satisfaction, marriage status

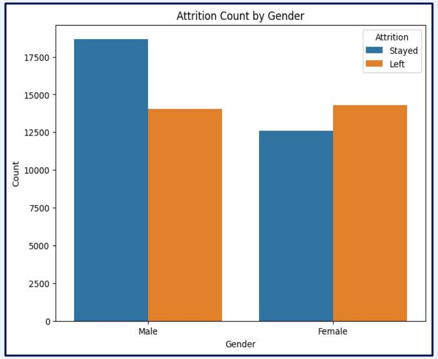
	Train	Test
# Observations	**59,598**	**14,900**
Split-Percentage	80%	20%
Target Variable	**Attrition (Stayed or Left)**	

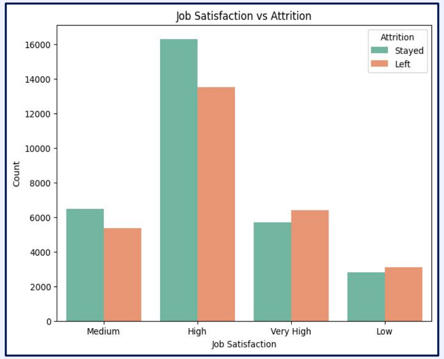
EDA + Preprocessing

Our exploratory analysis of the data indicates features that are relatively uncorrelated and ripe for pre-processing









We preprocessed our features using techniques like encoding and binning to improve model accuracy

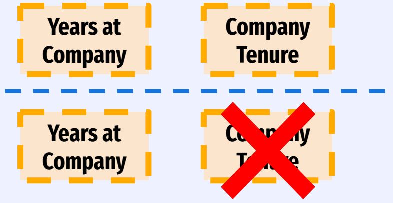
Label Encoding

Education		Education
High School	→	1
Bachelors		2

One Hot Encoding

Gender		Gender_ Male	Gender_ Female
Male	→	1	0
Female		0	1

Dropped Redundant Features



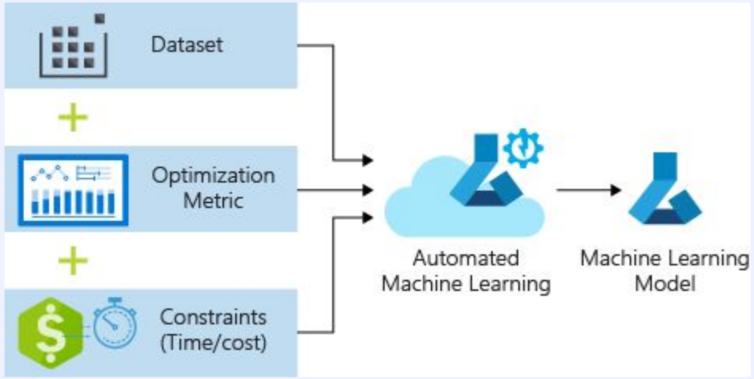
Binned Continuous Features

Age		Age_ 17-26	Age_ 43-51
16	→	1	0
45		0	1

After preprocessing, we pushed our train features to a feature store within Databricks itself

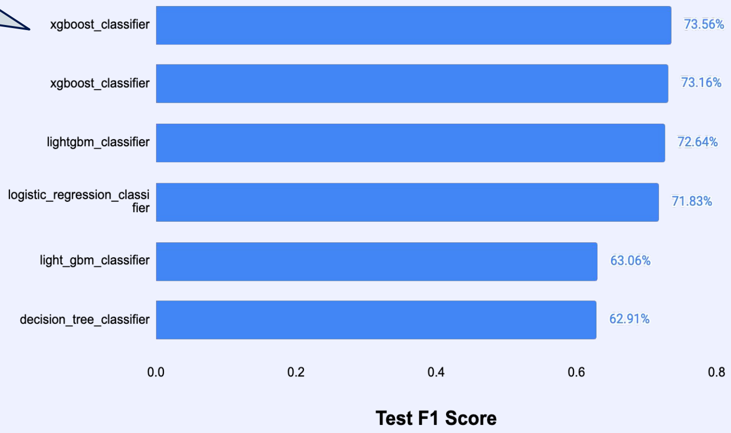
1. Data downloaded from Kaggle as .csv files (train, test)
2. Databricks ingestion into separate tables
3. Conduct preprocessing steps (binning, encoding, etc)
4. Push cleaned features to Feature Store in Databricks

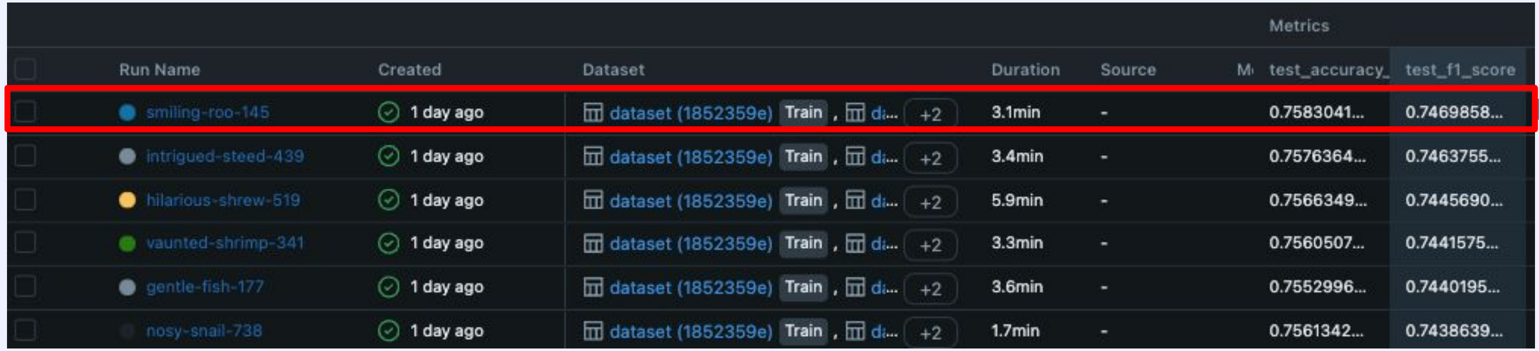
We deploy AutoML to explore model options and subsequently settled on XGBoost Classifier



https://softwareengineeringdaily.com/2019/05/15/introduction-to-automated-machine-learning-automl/

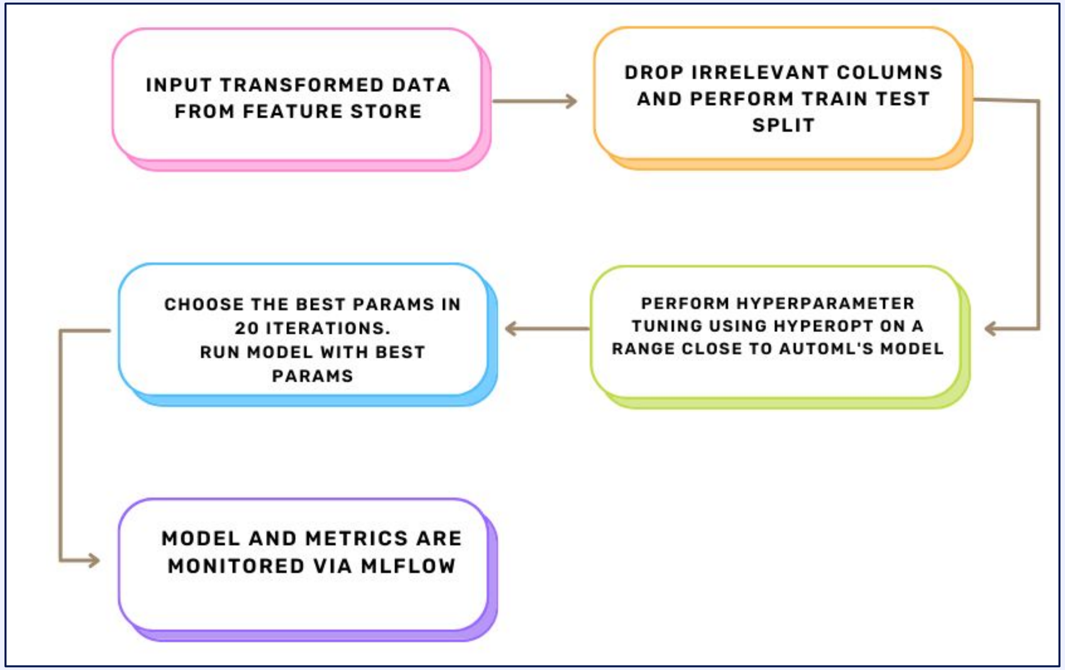
AutoML found XGBoost with an F1 score of ~74% as the best model



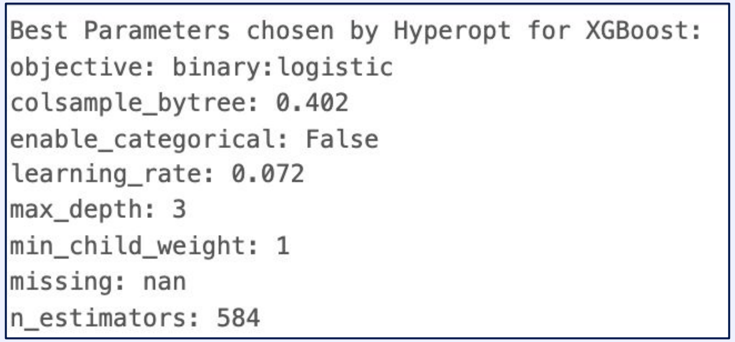


Run Name	Created	Dataset	Duration	Source	M	Metrics test_accuracy_	test_f1_score
smiling-roo-145	1 day ago	dataset (1852359e) Train , da... +2	3.1min	-		0.7583041...	0.7469858...
intrigued-steed-439	1 day ago	dataset (1852359e) Train , da... +2	3.4min	-		0.7576364...	0.7463755...
hilarious-shrew-519	1 day ago	dataset (1852359e) Train , da... +2	5.9min	-		0.7566349...	0.7445690...
vaunted-shrimp-341	1 day ago	dataset (1852359e) Train , da... +2	3.3min	-		0.7560507...	0.7441575...
gentle-fish-177	1 day ago	dataset (1852359e) Train , da... +2	3.6min	-		0.7552996...	0.7440195...
nosy-snail-738	1 day ago	dataset (1852359e) Train , da... +2	1.7min	-		0.7561342...	0.7438639...

We then create our XGBoost model using Hyperopt and MLFlow pipeline to generate our predictions and metric evaluations


INPUT TRANSFORMED DATA FROM FEATURE STORE
DROP IRRELEVANT COLUMNS AND PERFORM TRAIN TEST SPLIT
PERFORM HYPERPARAMETER TUNING USING HYPEROPT ON A RANGE CLOSE TO AUTOML'S MODEL
CHOOSE THE BEST PARAMS IN 20 ITERATIONS. RUN MODEL WITH BEST PARAMS
MODEL AND METRICS ARE MONITORED VIA MLFLOW

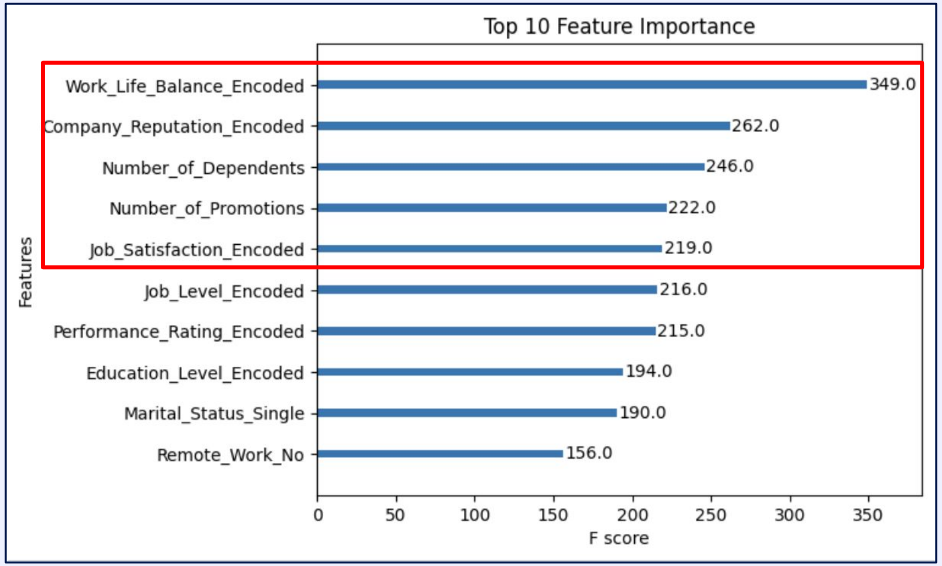
Modelling Process

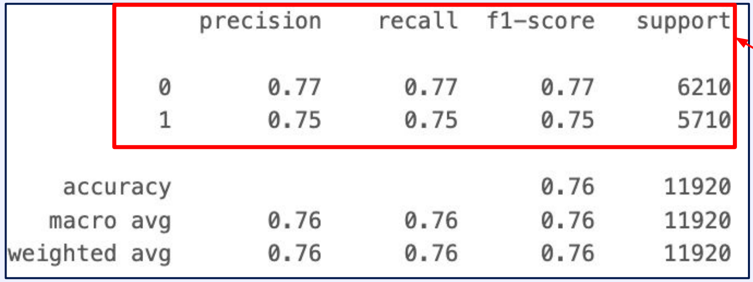


```
Best Parameters chosen by Hyperopt for XGBoost:
objective: binary:logistic
colsample_bytree: 0.402
enable_categorical: False
learning_rate: 0.072
max_depth: 3
min_child_weight: 1
missing: nan
n_estimators: 584
```

Best Parameters chosen by Hyperopt with values ranges based on baseline AutoML

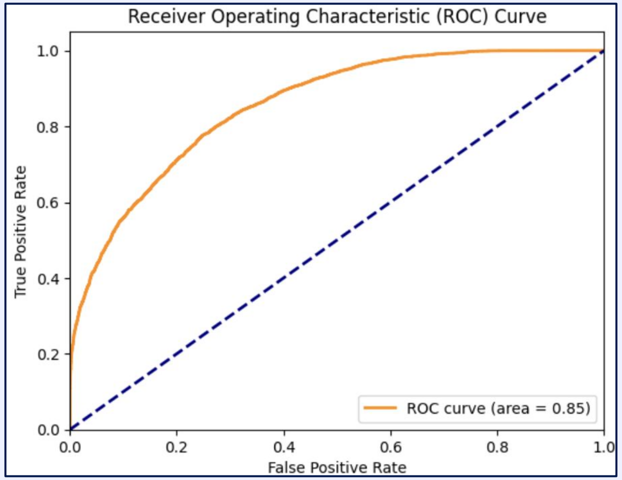
Metrics used for Evaluation and Results

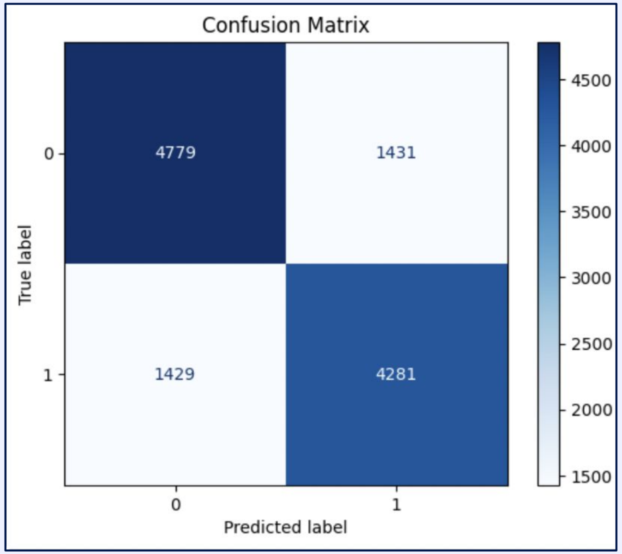




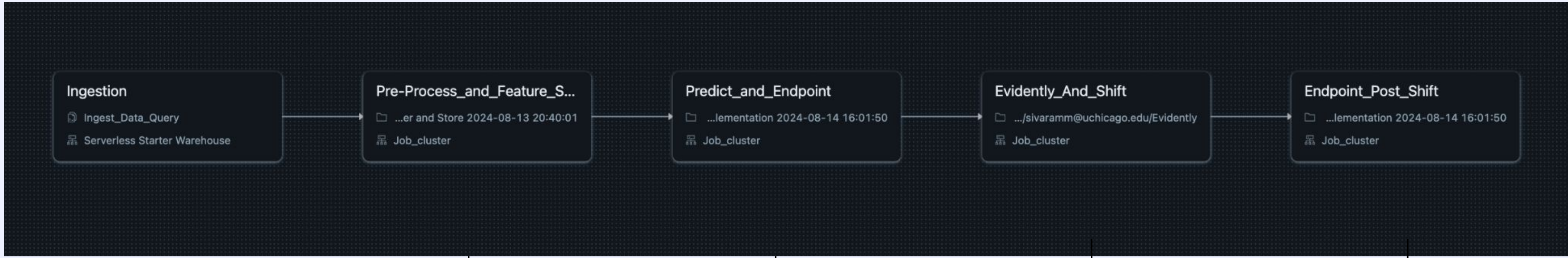
```
              precision    recall  f1-score   support

           0       0.77      0.77      0.77      6210
           1       0.75      0.75      0.75      5710

    accuracy                           0.76     11920
   macro avg       0.76      0.76      0.76     11920
weighted avg       0.76      0.76      0.76     11920
```

F1 score for both classes have similar performance indicating no bias





A Databricks workflow allows us to deploy our model in an environment that continuously ingests new data



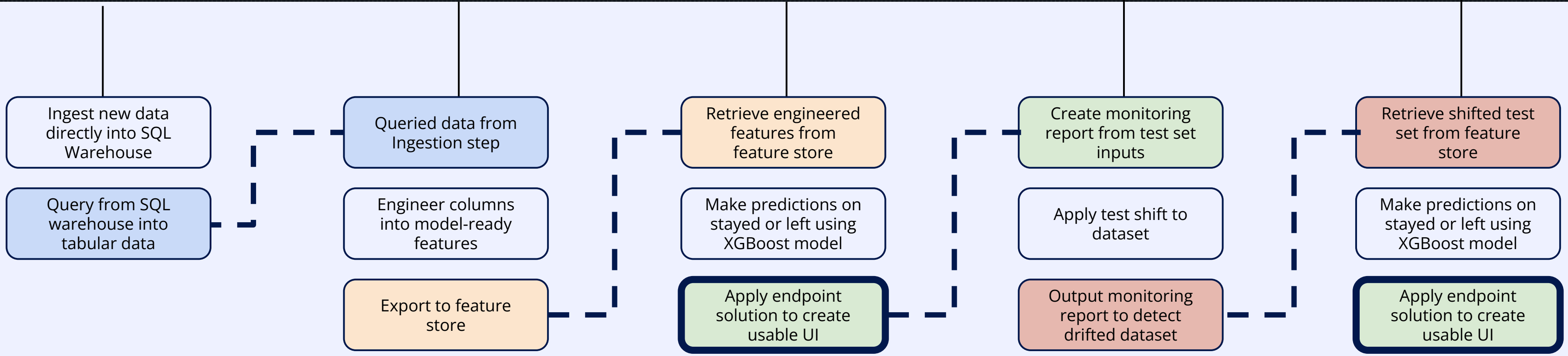
Ingestion
Ingest_Data_Query
Serverless Starter Warehouse

Pre-Process_and_Feature_S...
...er and Store 2024-08-13 20:40:01
Job_cluster

Predict_and_Endpoint
...lementation 2024-08-14 16:01:50
Job_cluster

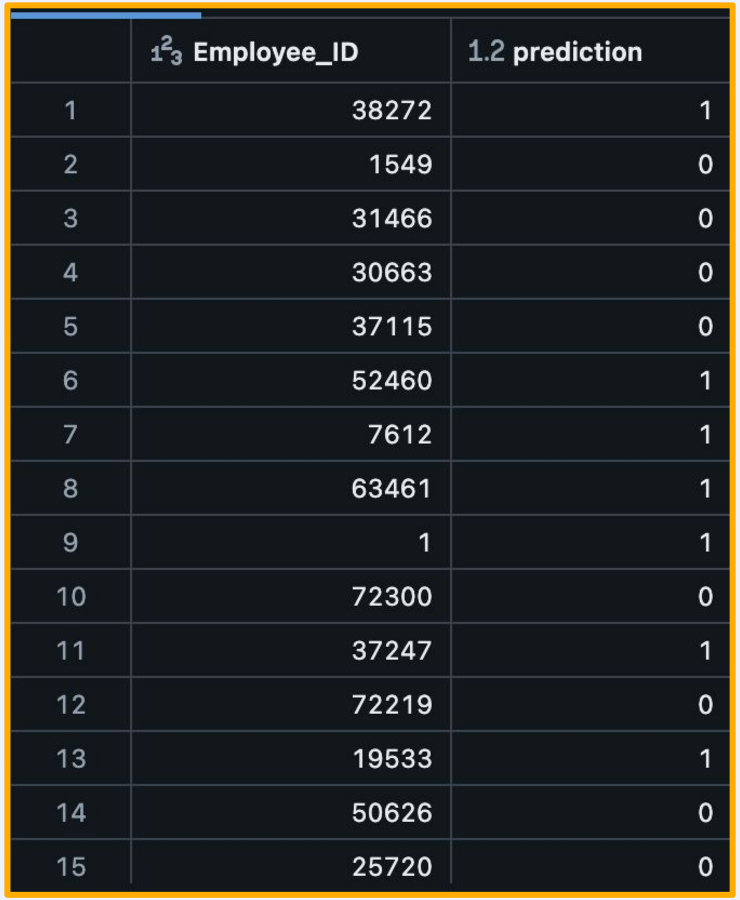
Evidently_And_Shift
.../sivaramm@uchicago.edu/Evidently
Job_cluster

Endpoint_Post_Shift
...lementation 2024-08-14 16:01:50
Job_cluster

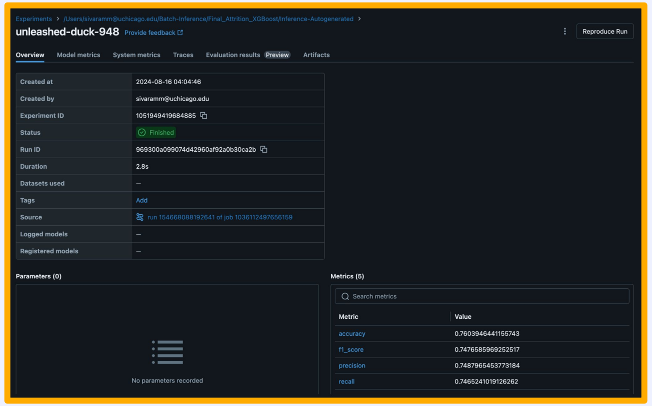

Ingest new data directly into SQL Warehouse
Query from SQL warehouse into tabular data
Queried data from Ingestion step
Engineer columns into model-ready features
Export to feature store
Retrieve engineered features from feature store
Make predictions on stayed or left using XGBoost model
Apply endpoint solution to create usable UI
Create monitoring report from test set inputs
Apply test shift to dataset
Output monitoring report to detect drifted dataset
Retrieve shifted test set from feature store
Make predictions on stayed or left using XGBoost model
Apply endpoint solution to create usable UI

The deployed model is used for inference to predict attrition

Deployment



	Employee_ID	prediction
1	38272	1
2	1549	0
3	31466	0
4	30663	0
5	37115	0
6	52460	1
7	7612	1
8	63461	1
9	1	1
10	72300	0
11	37247	1
12	72219	0
13	19533	1
14	50626	0
15	25720	0



Experiments > /Users/sivaramm@uchicago.edu/Batch-Inference/Final_Attrition_XGBoost/Inference-Autogenerated >

unleashed-duck-948 Provide feedback

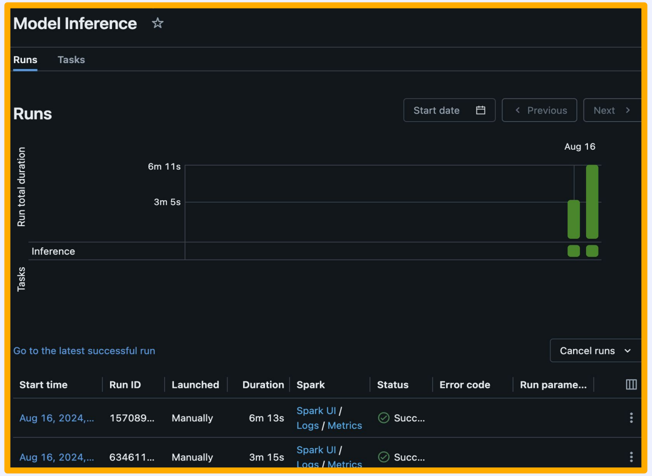
Created at	2024-08-16 04:04:46
Created by	sivaramm@uchicago.edu
Experiment ID	1051949419684885
Status	Finished
Run ID	969300a099074d42960af92a0b30ca2b
Duration	2.8s
Datasets used	—
Tags	Add
Source	run 154668088192641 of job 1036112497656159
Logged models	—
Registered models	—

Parameters (0)

No parameters recorded

Metrics (5)

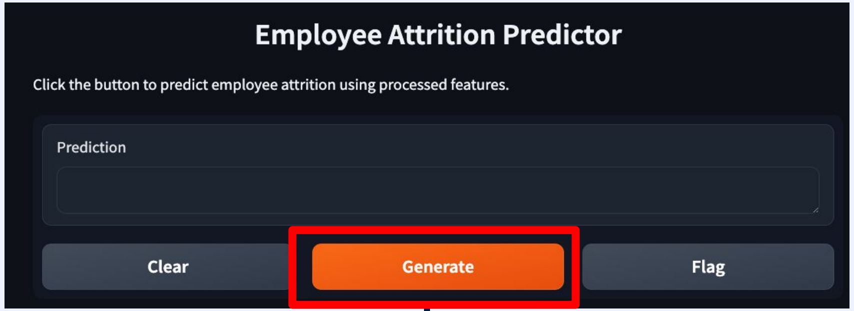
Metric	Value
accuracy	0.7603946441155743
f1_score	0.7476585969252517
precision	0.7487965453773184
recall	0.7465241019126262



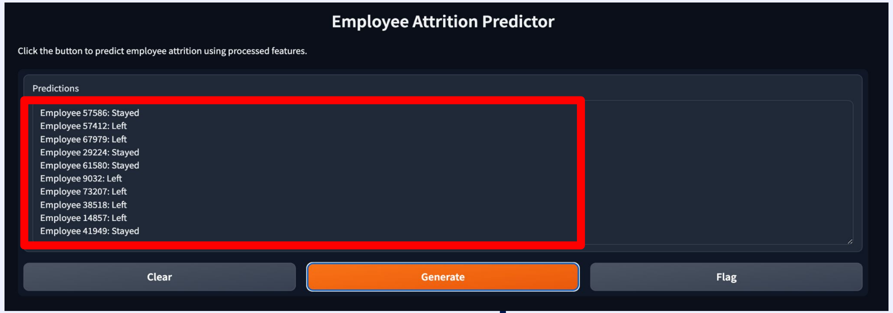
Model Inference

Start time	Run ID	Launched	Duration	Spark	Status	Error code	Run parame...
Aug 16, 2024,...	157089...	Manually	6m 13s	Spark UI / Logs / Metrics	Succ...		
Aug 16, 2024,...	634611...	Manually	3m 15s	Spark UI / Logs / Metrics	Succ...		

Gradio offers a simple endpoint solution for our pipeline, outputting our inference results in an intuitive user interface


Employee Attrition Predictor

Click the button to predict employee attrition using processed features.


Employee Attrition Predictor

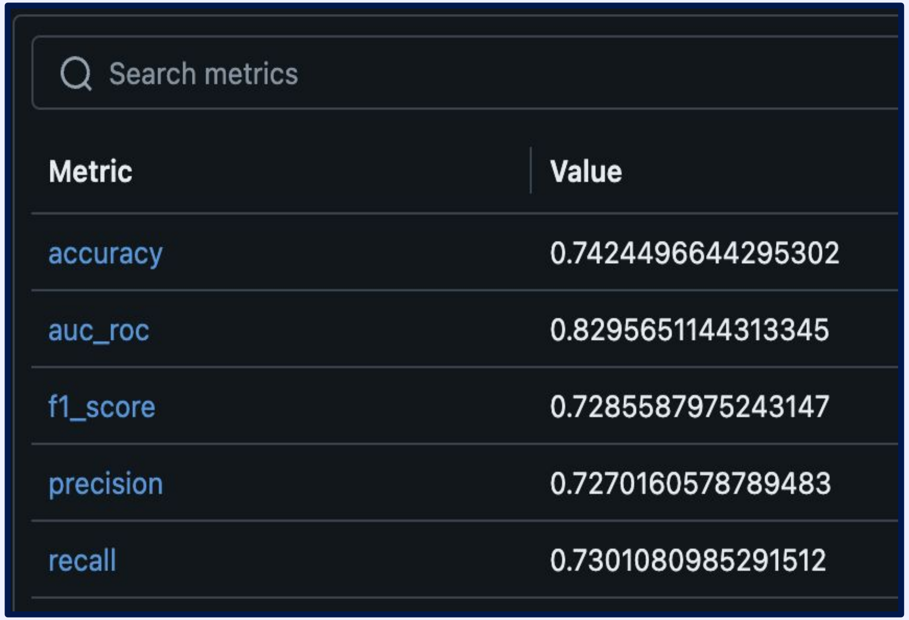
Click the button to predict employee attrition using processed features.

Predictions

```
Employee 57586: Stayed
Employee 57412: Left
Employee 67979: Left
Employee 29224: Stayed
Employee 61580: Stayed
Employee 9032: Left
Employee 73207: Left
Employee 38518: Left
Employee 14857: Left
Employee 41949: Stayed
```

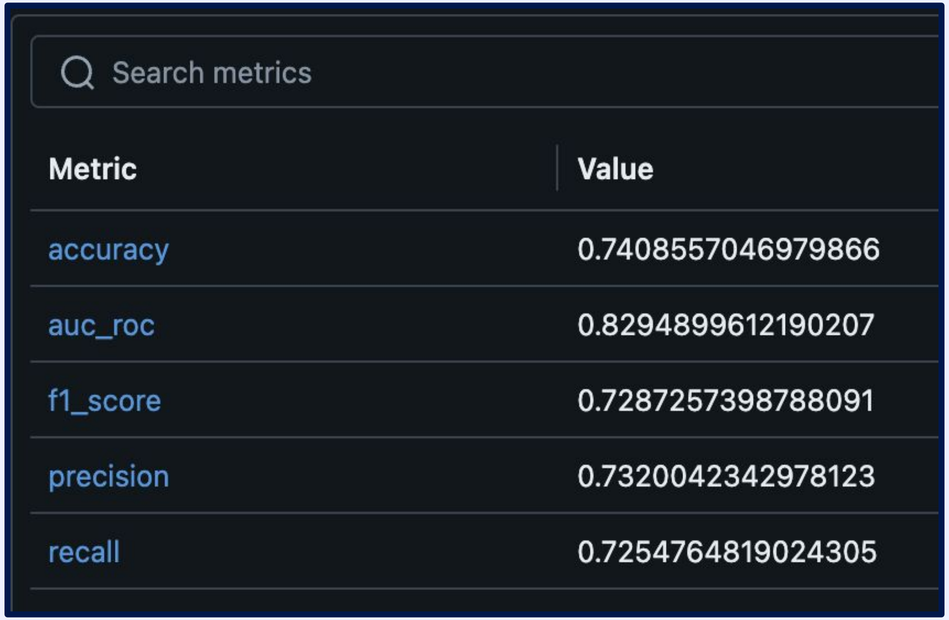
*Example of **10 attrition inferences** our model makes from newly ingested data in the Gradio endpoint UI*

Model Monitoring with MLFlow for two separate runs



Search metrics

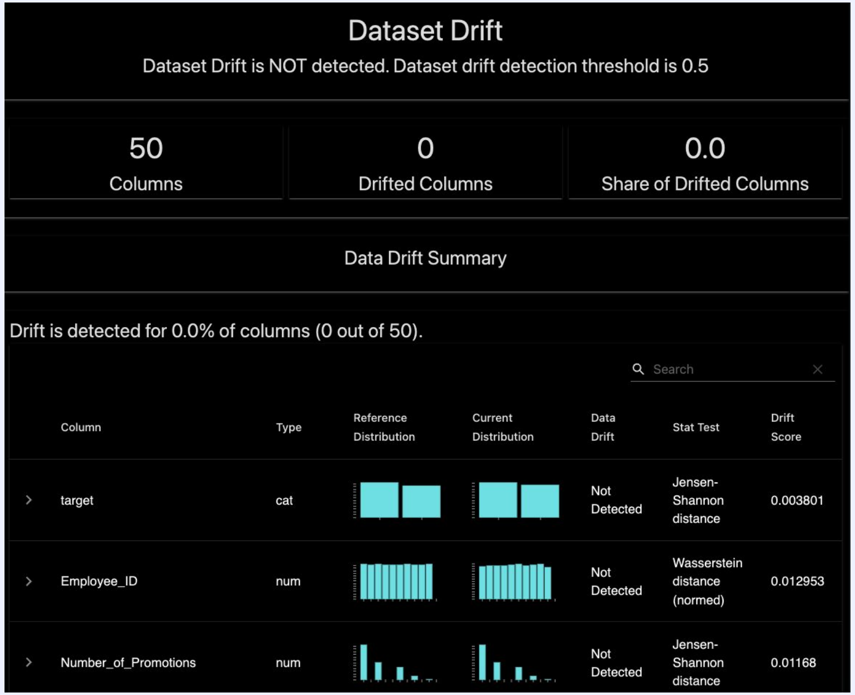
Metric	Value
accuracy	0.7424496644295302
auc_roc	0.8295651144313345
f1_score	0.7285587975243147
precision	0.7270160578789483
recall	0.7301080985291512



Search metrics

Metric	Value
accuracy	0.7408557046979866
auc_roc	0.8294899612190207
f1_score	0.7287257398788091
precision	0.7320042342978123
recall	0.7254764819024305

We spin up Evidently AI to generate monitoring reports to track our model and its performance



Dataset Drift

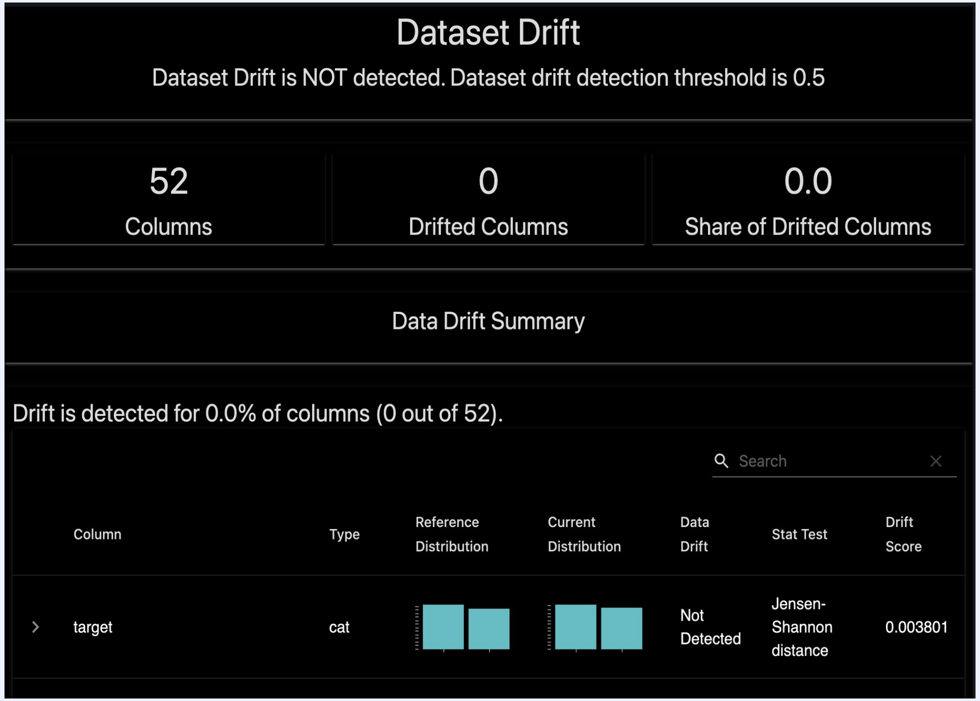
Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

50	0	0.0
Columns	Drifted Columns	Share of Drifted Columns

Data Drift Summary

Drift is detected for 0.0% of columns (0 out of 50).

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
target	cat			Not Detected	Jensen-Shannon distance	0.003801
Employee_ID	num			Not Detected	Wasserstein distance (normed)	0.012953
Number_of_Promotions	num			Not Detected	Jensen-Shannon distance	0.01168



Dataset Drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

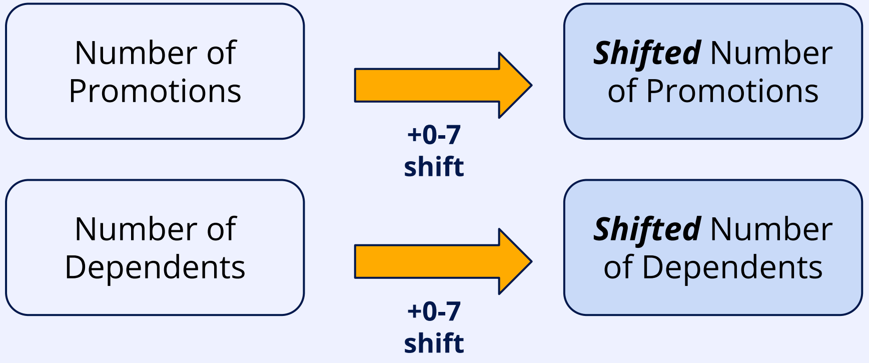
52	0	0.0
Columns	Drifted Columns	Share of Drifted Columns

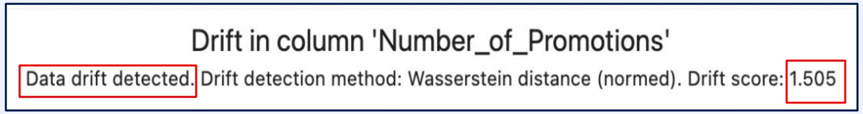
Data Drift Summary

Drift is detected for 0.0% of columns (0 out of 52).

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
target	cat			Not Detected	Jensen-Shannon distance	0.003801

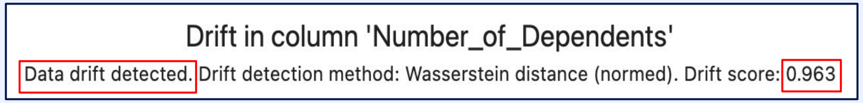
To explore "changed" test data, we shift Number of Promotions and Number of Dependents in our dataset


Number of Promotions
+0-7 shift
Shifted Number of Promotions
Number of Dependents
+0-7 shift
Shifted Number of Dependents



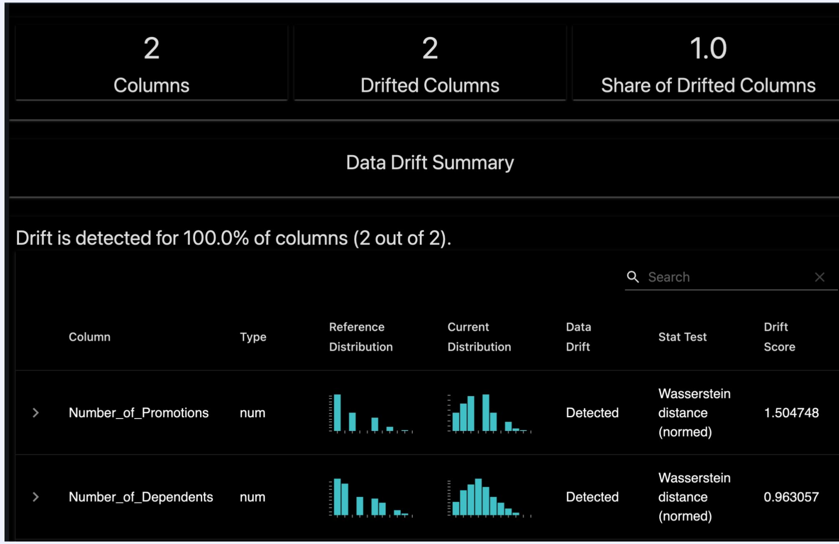
Drift in column 'Number_of_Promotions'

Data drift detected. Drift detection method: Wasserstein distance (normed). Drift score: 1.505



Drift in column 'Number_of_Dependents'

Data drift detected. Drift detection method: Wasserstein distance (normed). Drift score: 0.963



2	2	1.0
Columns	Drifted Columns	Share of Drifted Columns

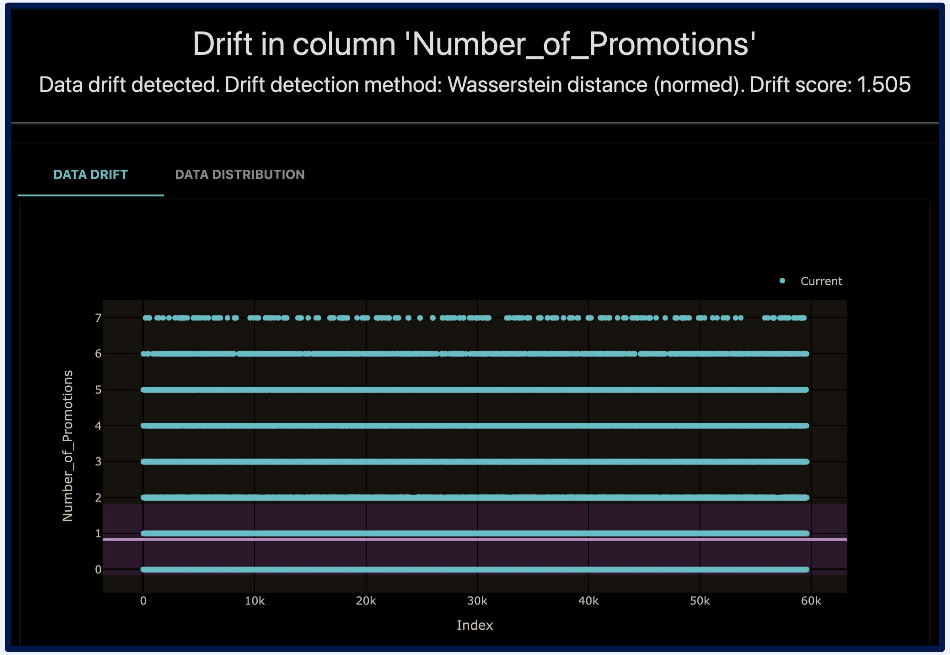
Data Drift Summary

Drift is detected for 100.0% of columns (2 out of 2).

Search

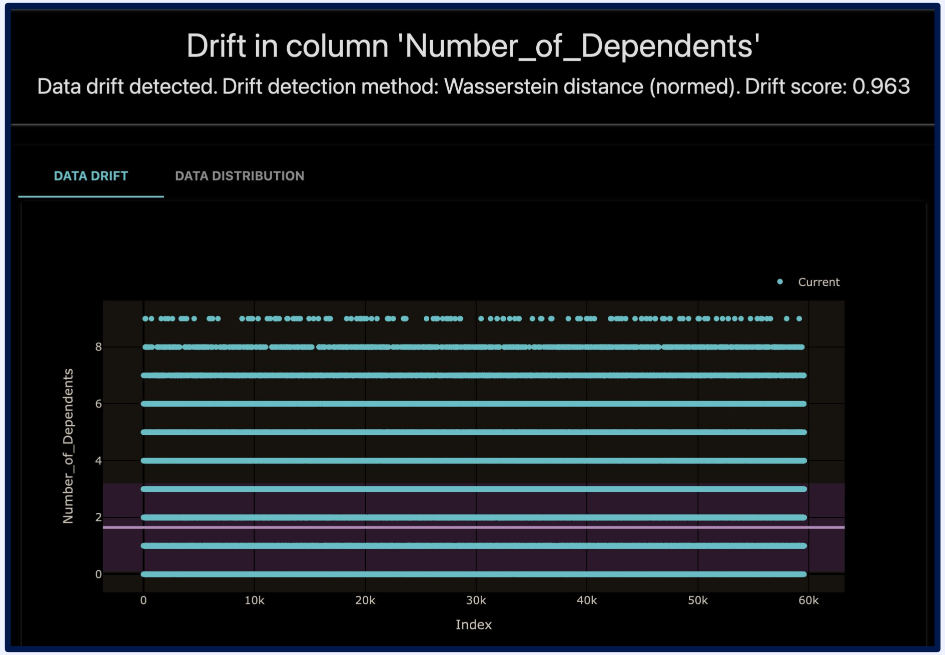
Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
Number_of_Promotions	num			Detected	Wasserstein distance (normed)	1.504748
Number_of_Dependents	num			Detected	Wasserstein distance (normed)	0.963057

Evidently AI automatically displays drift insights for our changed features



Drift in column 'Number_of_Promotions'

Data drift detected. Drift detection method: Wasserstein distance (normed). Drift score: 1.505



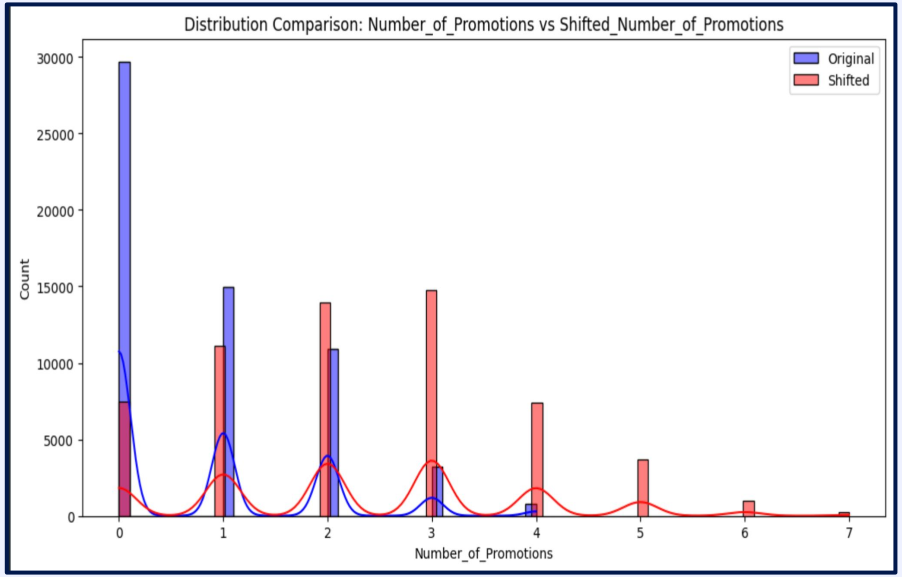
Drift in column 'Number_of_Dependents'

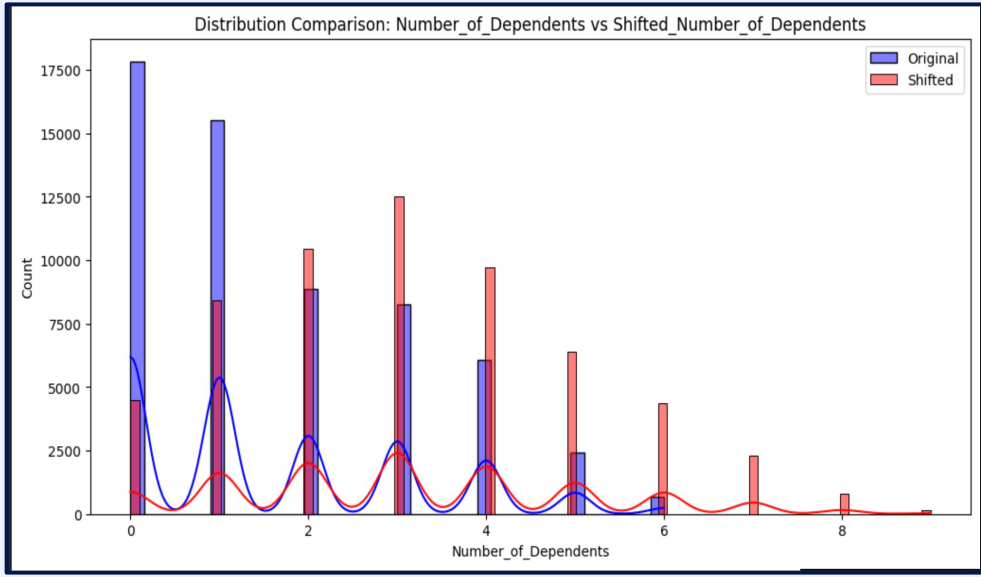
Data drift detected. Drift detection method: Wasserstein distance (normed). Drift score: 0.963

Drift Scores Higher than the Threshold of 0.5, hence Drift is detected

Monitoring + Test Data Drift

Evidently AI also displays charts to effectively monitor specific shifts in features





Link to Video Demonstration

https://drive.google.com/file/d/1RghT9TW1hIk8lsRimOdat9OhvWh3DfIV/view?usp=share_link

Link to Notebooks

Link to Notebooks

https://uchicago-team2-databricks.cloud.databricks.com/browse/folders/1051949419685084?o=29437060
2992426

Questions?

For any follow-ups, please email:

Apoorv Anand - apanand@uchicago.edu

Sivaram Mandava - sivaramm@uchicago.edu

Ashmita Mukherjee - ashmitam@uchicago.edu