

Inbound Agent Matching

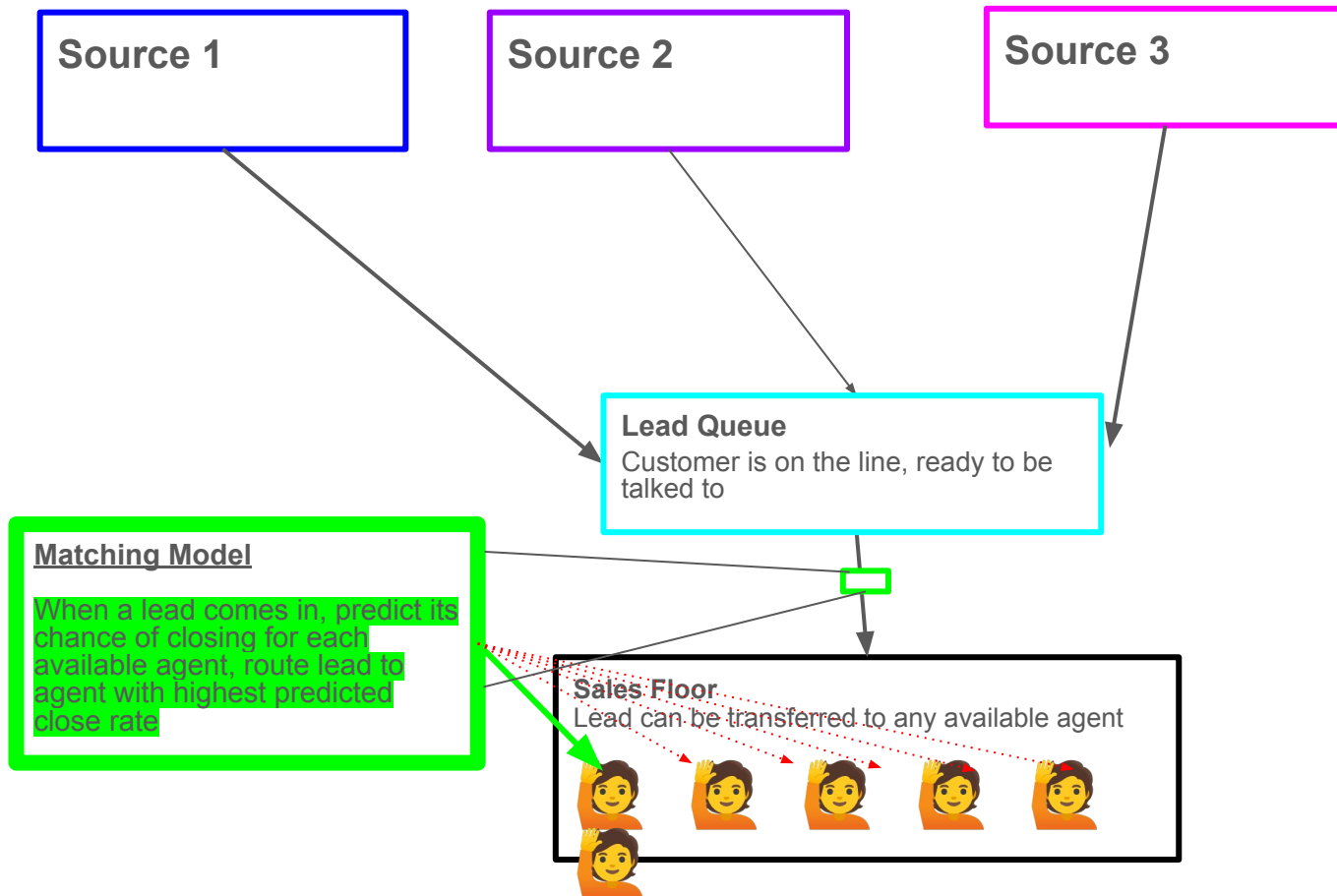
Farah Ahmed

08/15/2024

Outline

- Motivation
- Model
 - Data Filtering
 - Model Training
 - Model Output
- EDA
- Model Evaluation
- MLOps Cache
- Conclusion and Future Planning

Overview



Motivation

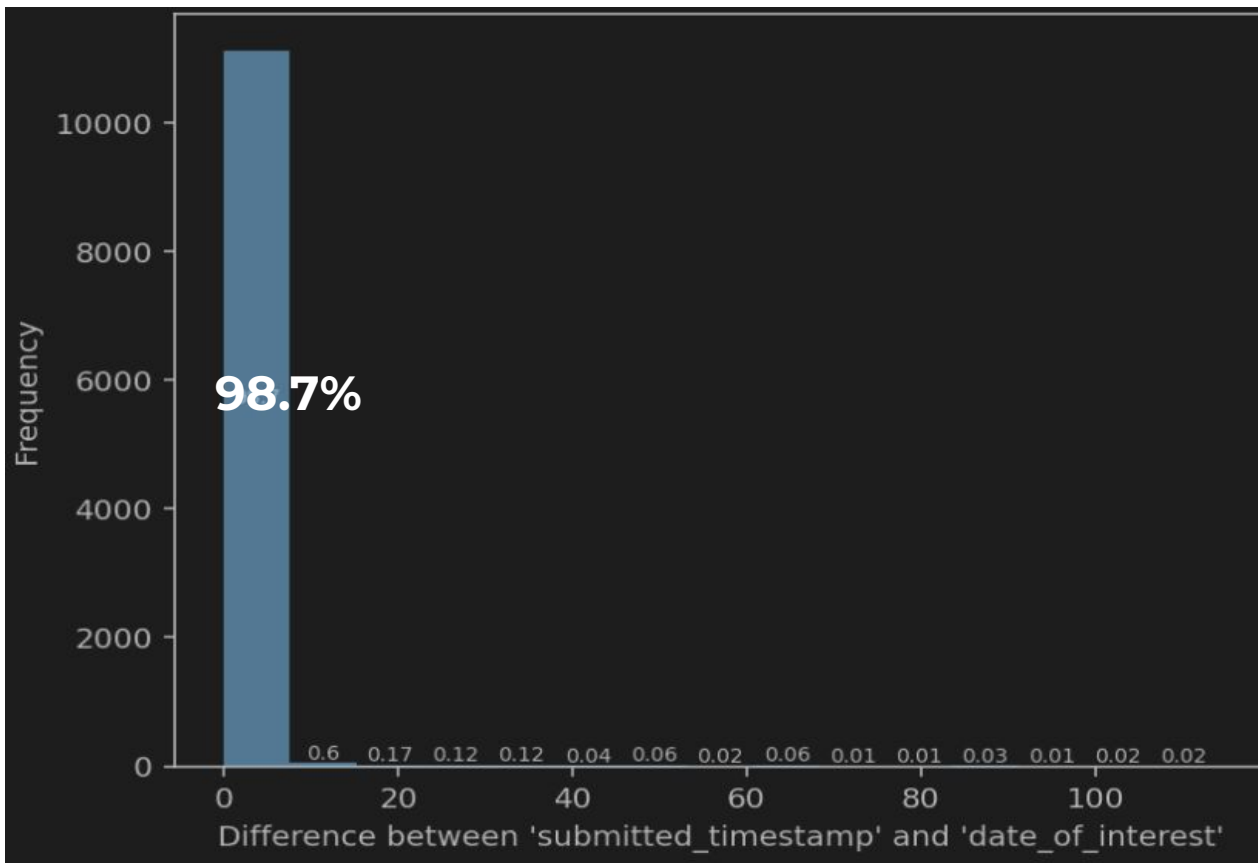
- Ensures leads connect with right agent to increase overall close rate
- Get lead to a compatible agent on first contact to reduce need for agent-to-agent transfers, wait time and increase efficiency
- Ensures fair and effective lead assignment
- Boost customer satisfaction and loyalty

Data Filtering

- Which agent to attribute lead
 - First agent that was connected to lead
- Which applications to include as target
 - Any policy
 - Keep only the first application per lead

```
and application.product in ('Medicare Advantage','Medicare Supplement')  
where lead_applications.application_number = 1
```

Data Filtering



Our Approach on Feature Engineering

- Collect as much features as we can
- Develop the model using all available features to understand their predictive power
- Even if all model features aren't consistently available in production, it is valuable to examine their effect on model performance

Model Type

- Binary Classification Model
 - XGBoost (Extreme Gradient Boosting) Classifier
- Why XGBoost?
 - Particularly effective for tabular data.
 - Provides insights into feature important
 - Optimized for speed and computational efficiency, making it well suited for large datasets
 - XGBoost can automatically detect and model non-linear relationships without needing to manually create new features.

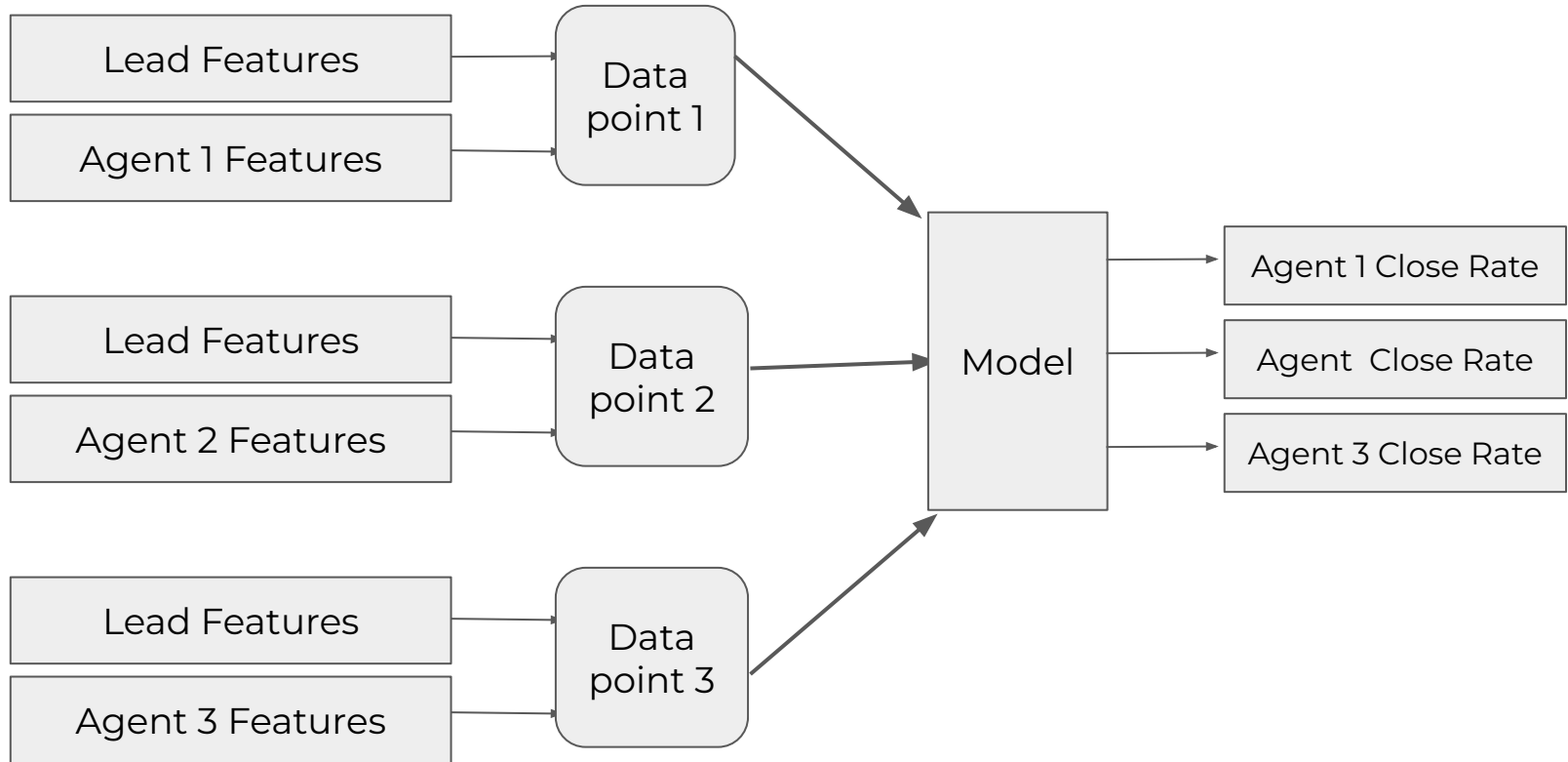
Primary Training Features

- **Lead Features** : Source, geographic features
- **Agent Features**: Agent username, start date
- **Historical Features**:
 - Close rates and applications per carrier for each lead sources, agents, queues and states
- **Derived Features**: Customer age, agent employment duration

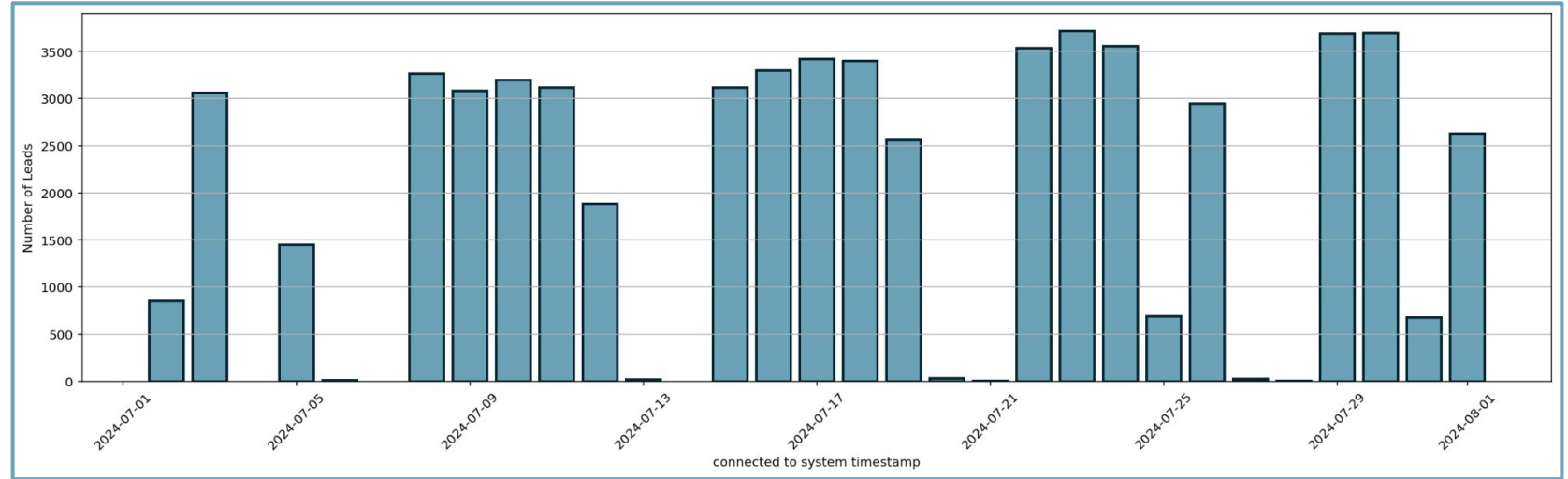
Target Variable and Model Output

- Target Variable
 - Indicates whether an application was closed successfully
 - Binary class, 1 if application exists otherwise 0.
- Model Output
 - The model predicts the probability of closing a lead for each available agents.

Flowchart of Model Input and Output

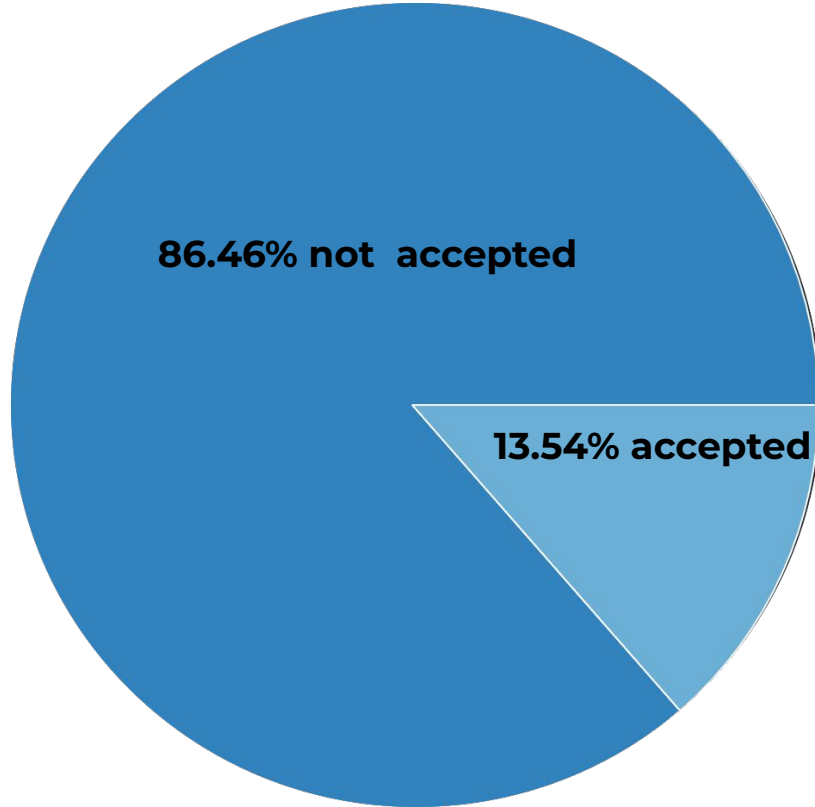


Exploratory Data Analysis

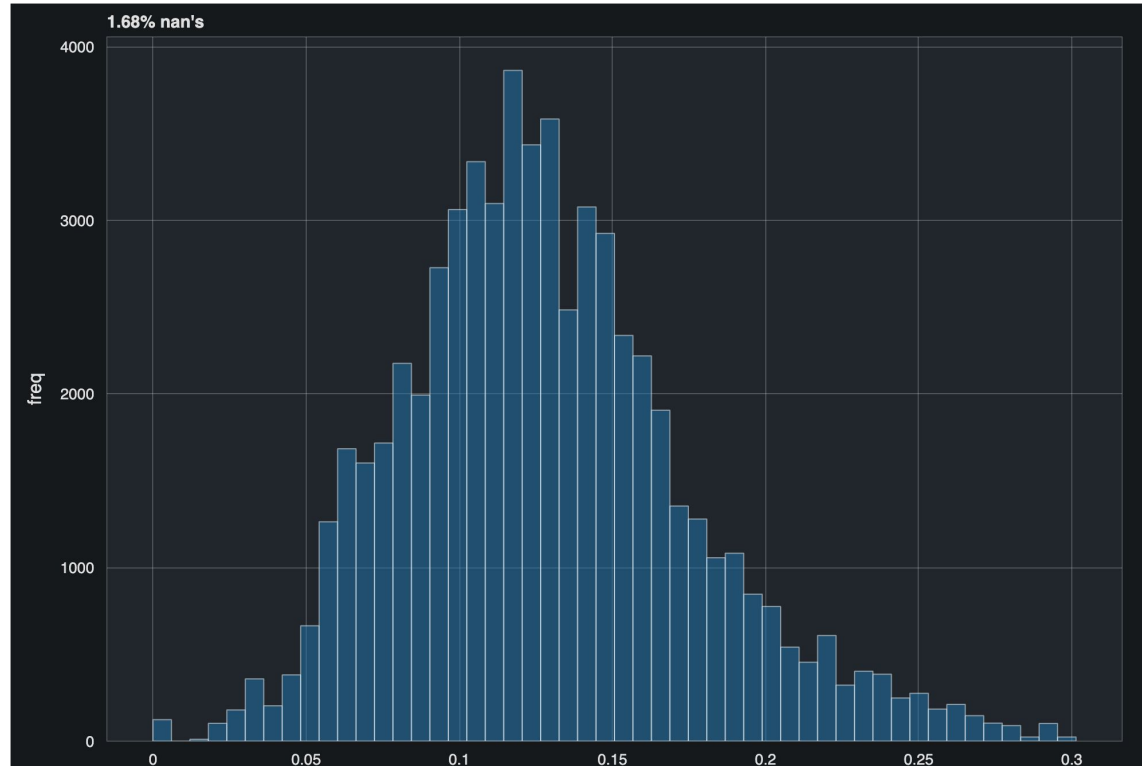


- Number of leads get connected to the system each day

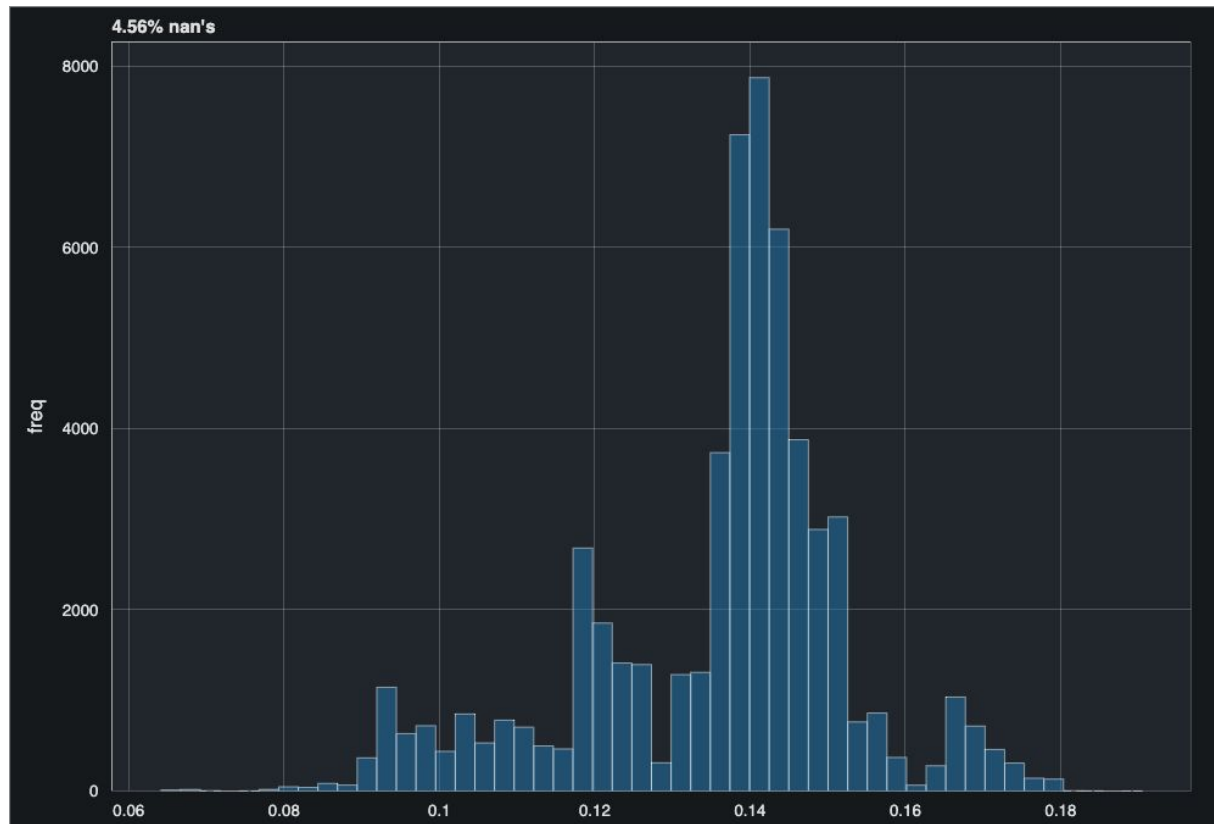
Exploratory Data Analysis (2/6)



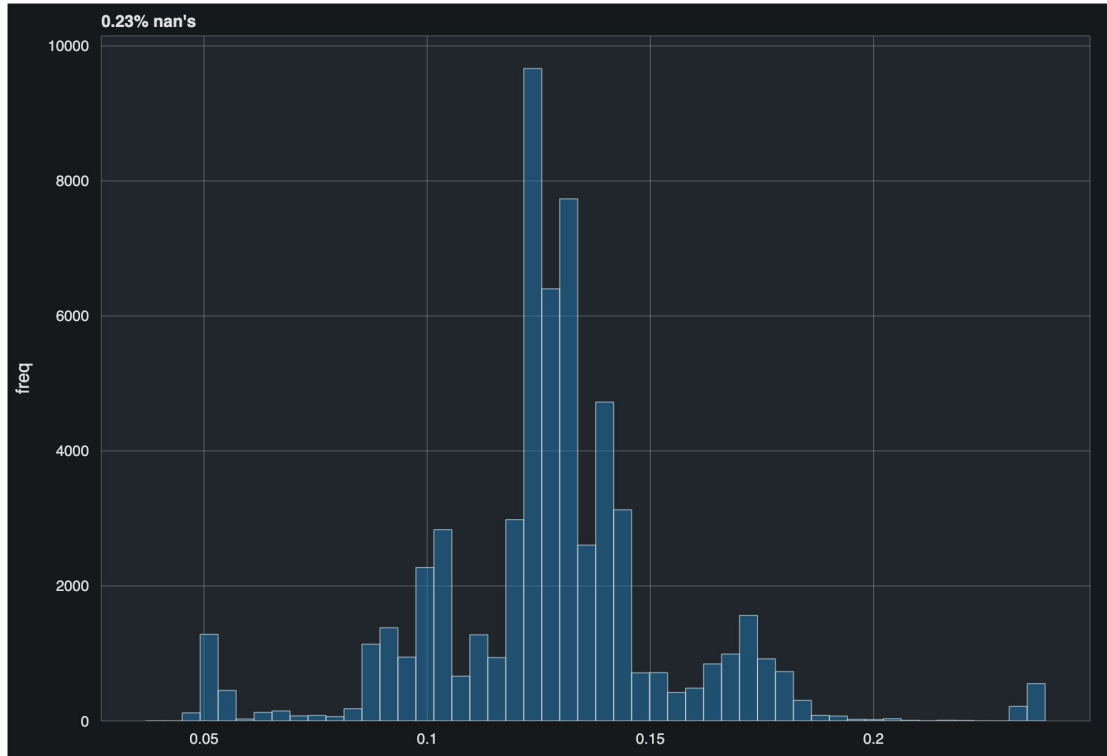
Exploratory Data Analysis (3/6)



Exploratory Data Analysis (6/6)



Exploratory Data Analysis (4/6)



Model Training

- Data Split
 - Used **train_test_split** with 80/20 ratio for training and testing
- Cross Validation
 - StratifiedKFold
- Hyperparameter Tuning
 - Randomized Grid Search with 160 iterations
- We got a BSS of 0.0374 on our test set

Model Evaluation

- Evaluation metric

- Brier Skill Score (BSS)

$$BSS = 1 - \frac{BS}{BS_{ref}}$$

- What is (BSS)

- BSS is a measure of how well our model performs compared to a reference model

- What is BS (Brier Score)

- A strictly proper score function that measures the accuracy of probabilistic predictions

- What is BS

- The brier score of our reference model is the overall close rate

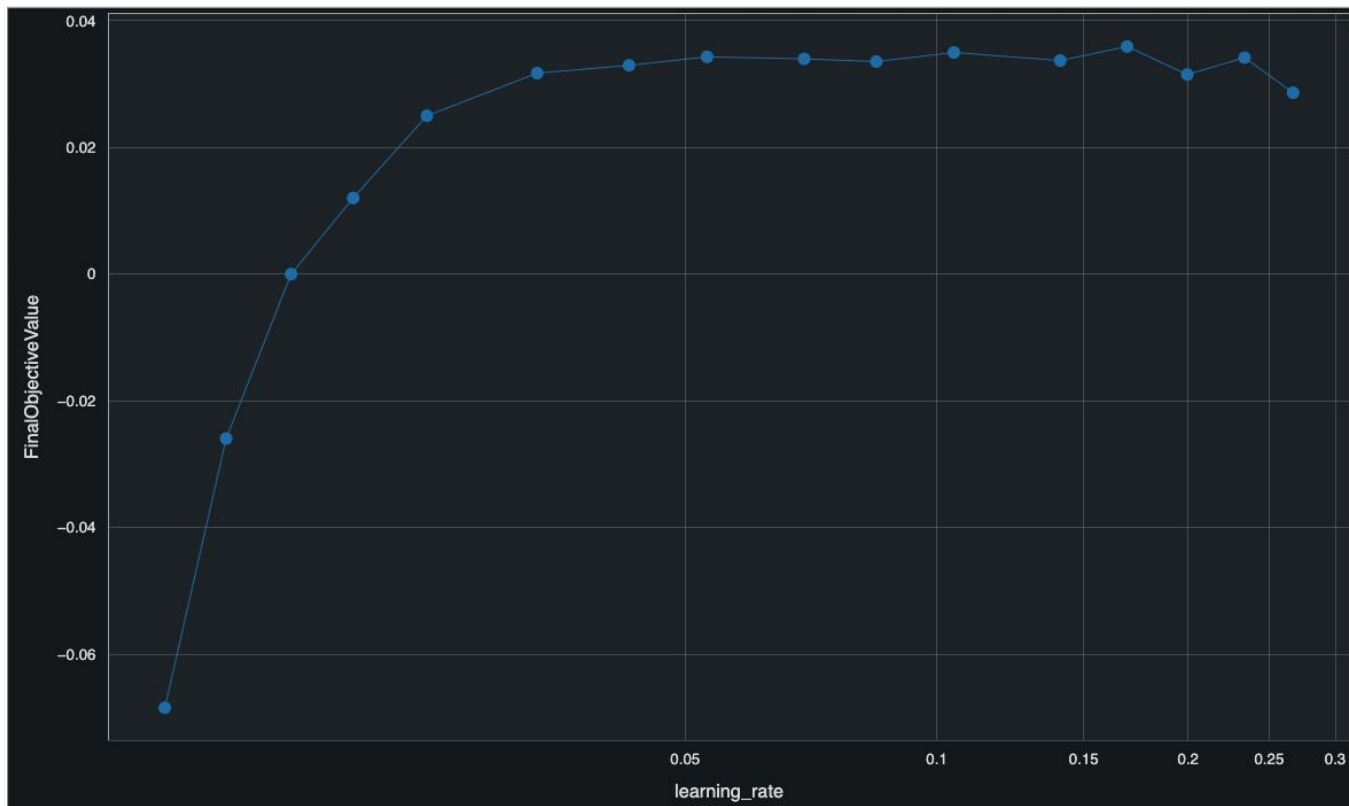
$$BS = \frac{1}{N} \sum_{t=1}^N (f_t - o_t)^2$$

the prediction values will be the

Model Evaluation

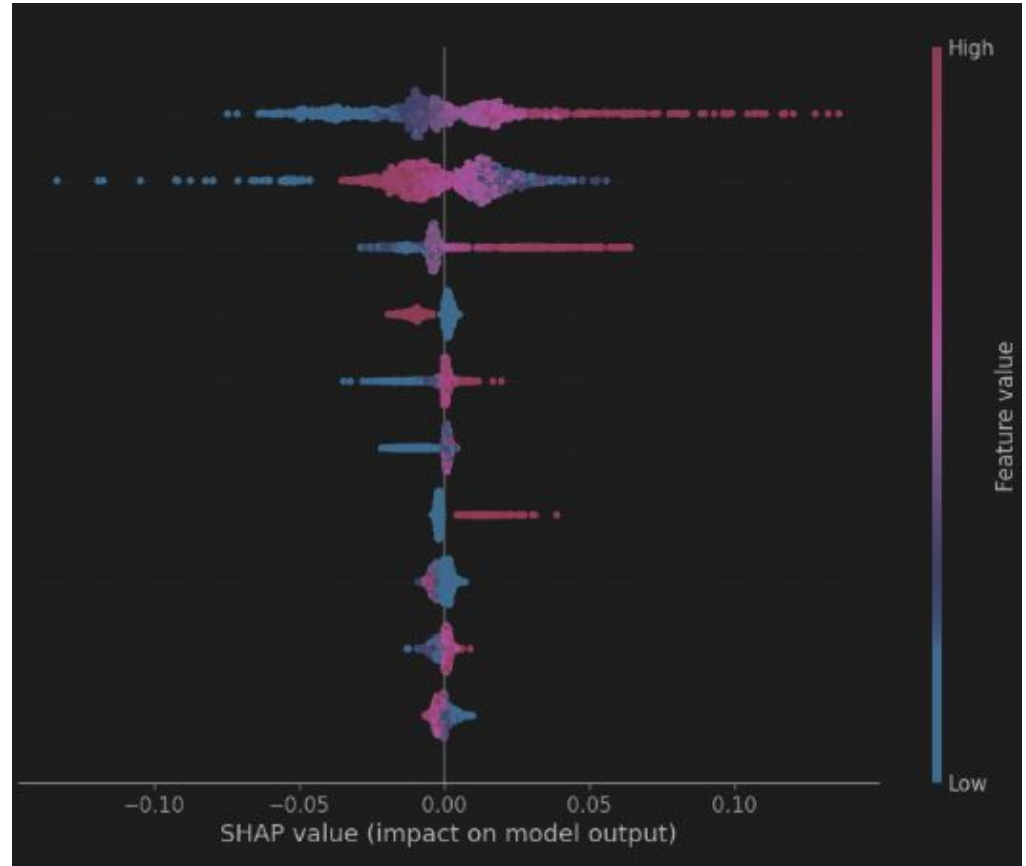
- Why use BSS?
 - Ensures accurate evaluation of agent/lead matches by providing well-calibrated probabilities
- What is well-calibrated model?
 - A model is considered well-calibrated if it's predicted probabilities correspond closely to the actual probabilities of the outcomes
- BSS of our model
 - BSS value **0.03573** indicates that our model is around 3.6% better than the baseline model

Hyperparameter Plot

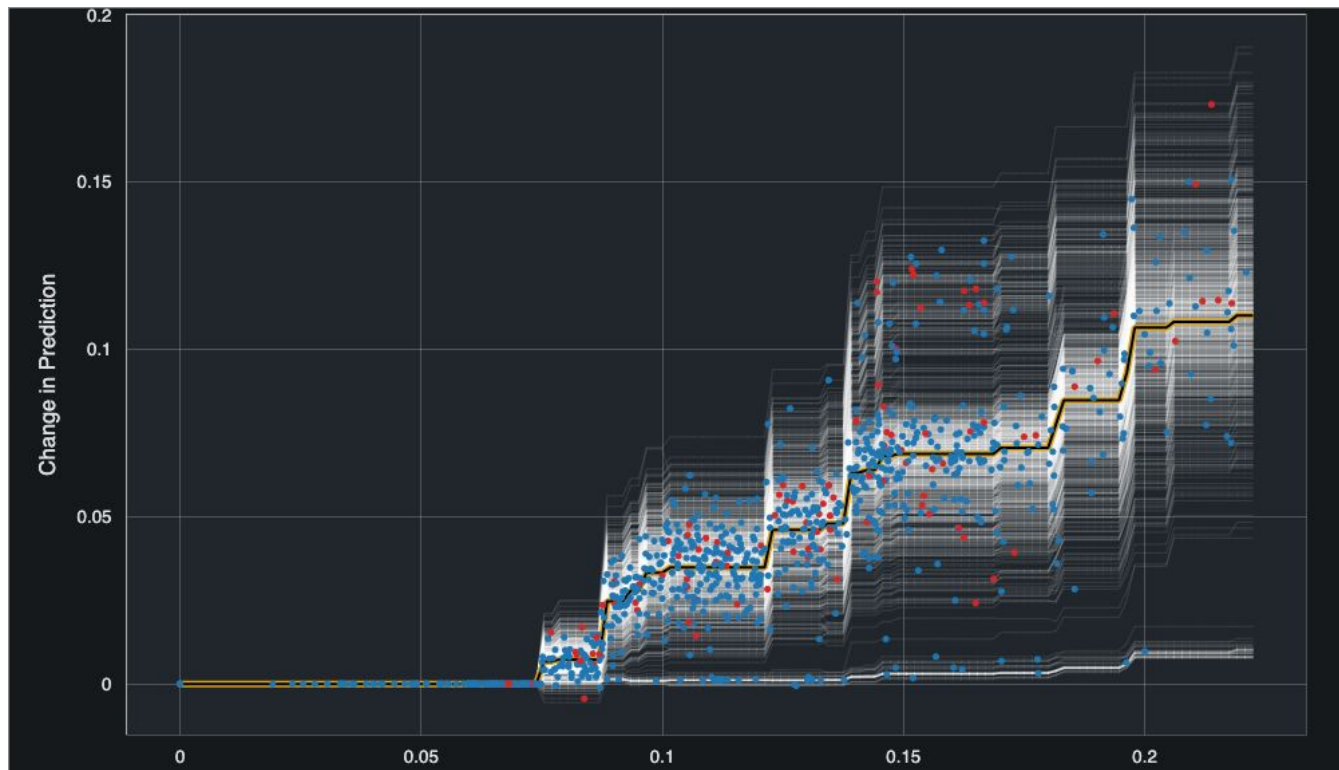


SHAP Plot

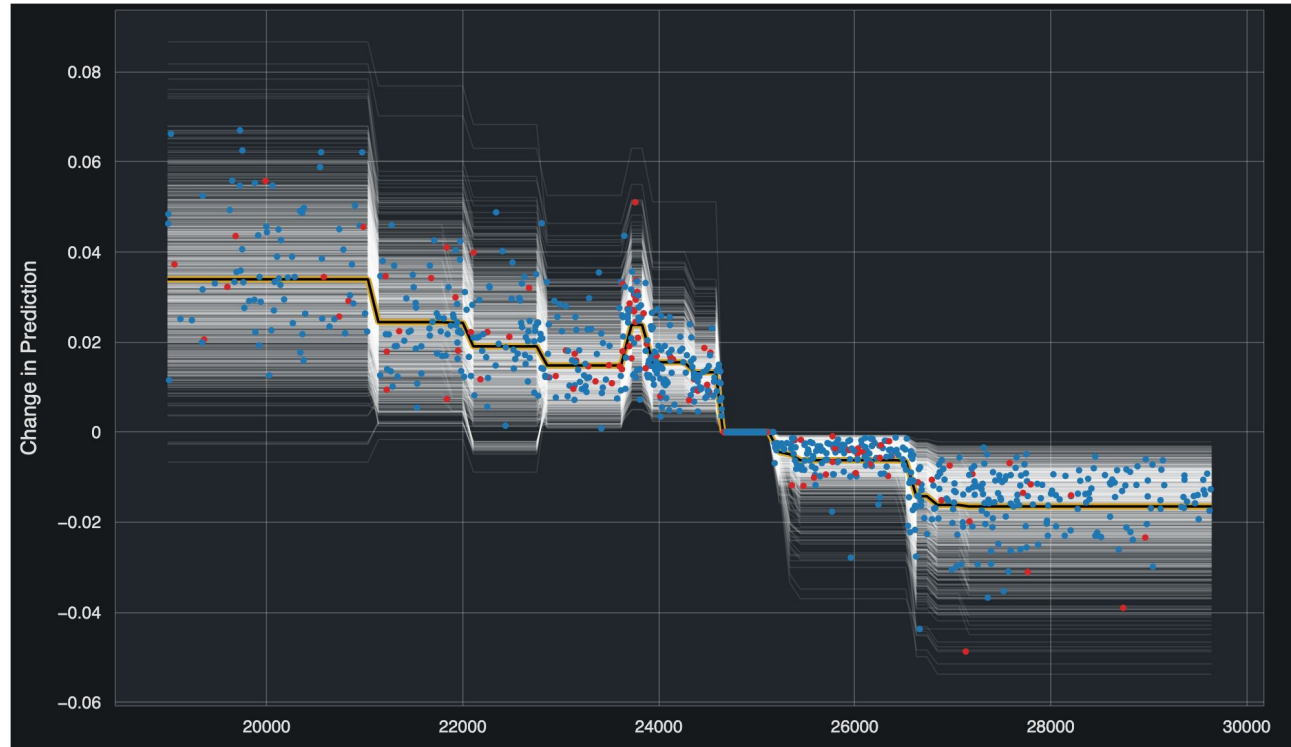
Feature 1
Feature 2
Feature 3



Ice Plot (1/3)



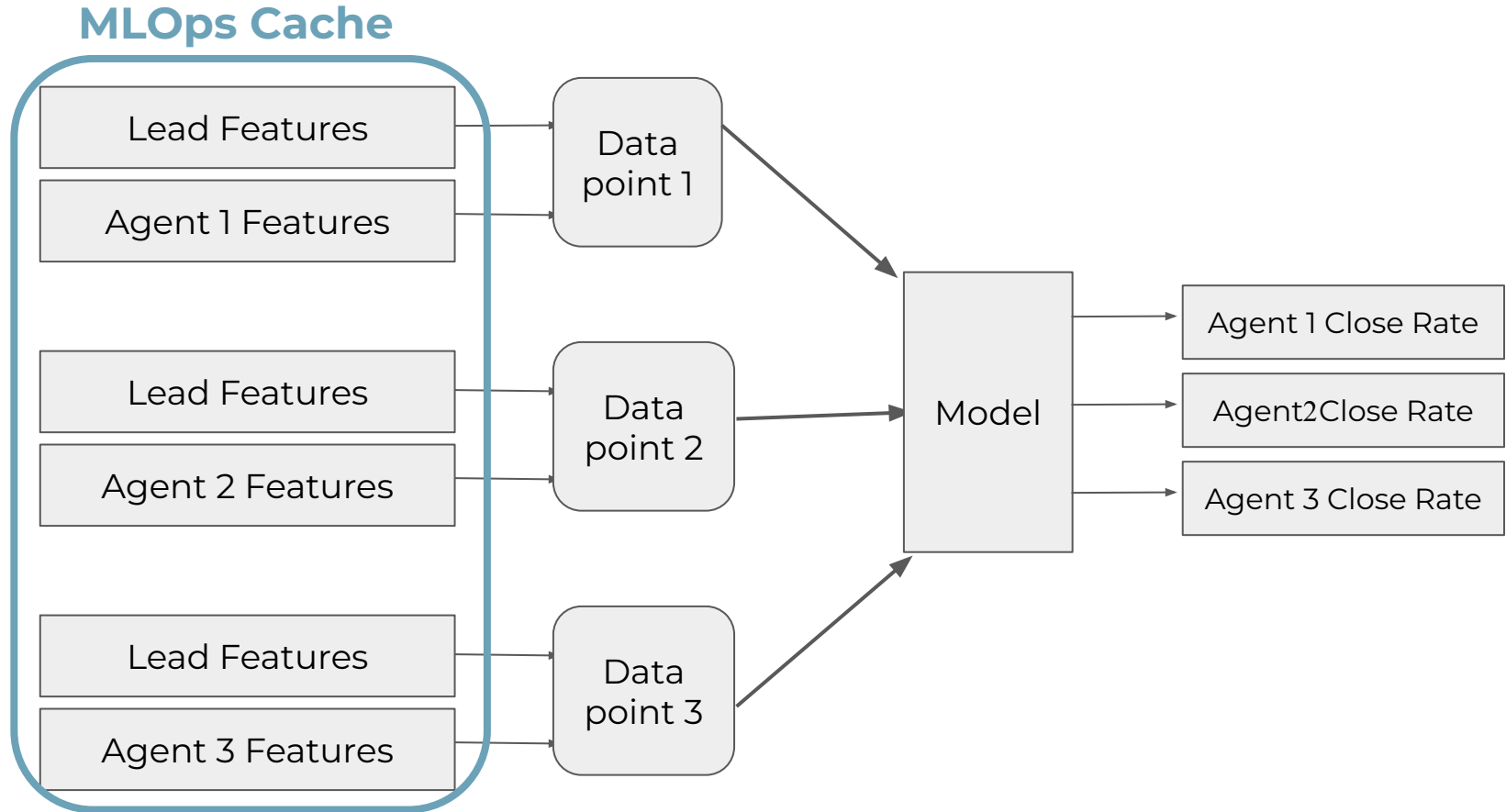
Ice Plot (2/3)



MLOps Cache

- Why use MLOps cache?
 - Easy and fast to access the pre-processed data
- Features we put in MLOps cache
 - Close rates for each lead sources, agents, queues and states
- How does MLOps works?
 - Cache will be updated every night with the previous 30 days of historical data

Flowchart of Model with MLOps



Conclusion

- Potential problem
 - Agent with previously higher close rate might get most leads
 - Hard to tell which agent is improving as good agents get best leads
- Possible solution
 - Keep 10% holdout set to randomly assign leads to the agents
- Expectation from the solution
 - Get better idea if agent is improving or not
 - All agents have equal chance to get best leads

Future Plan

- Building reports to check model performance over the time to have an idea about how often the model needs to be re-trained
- Try new algorithms/optimization methods for better performance

Thank you