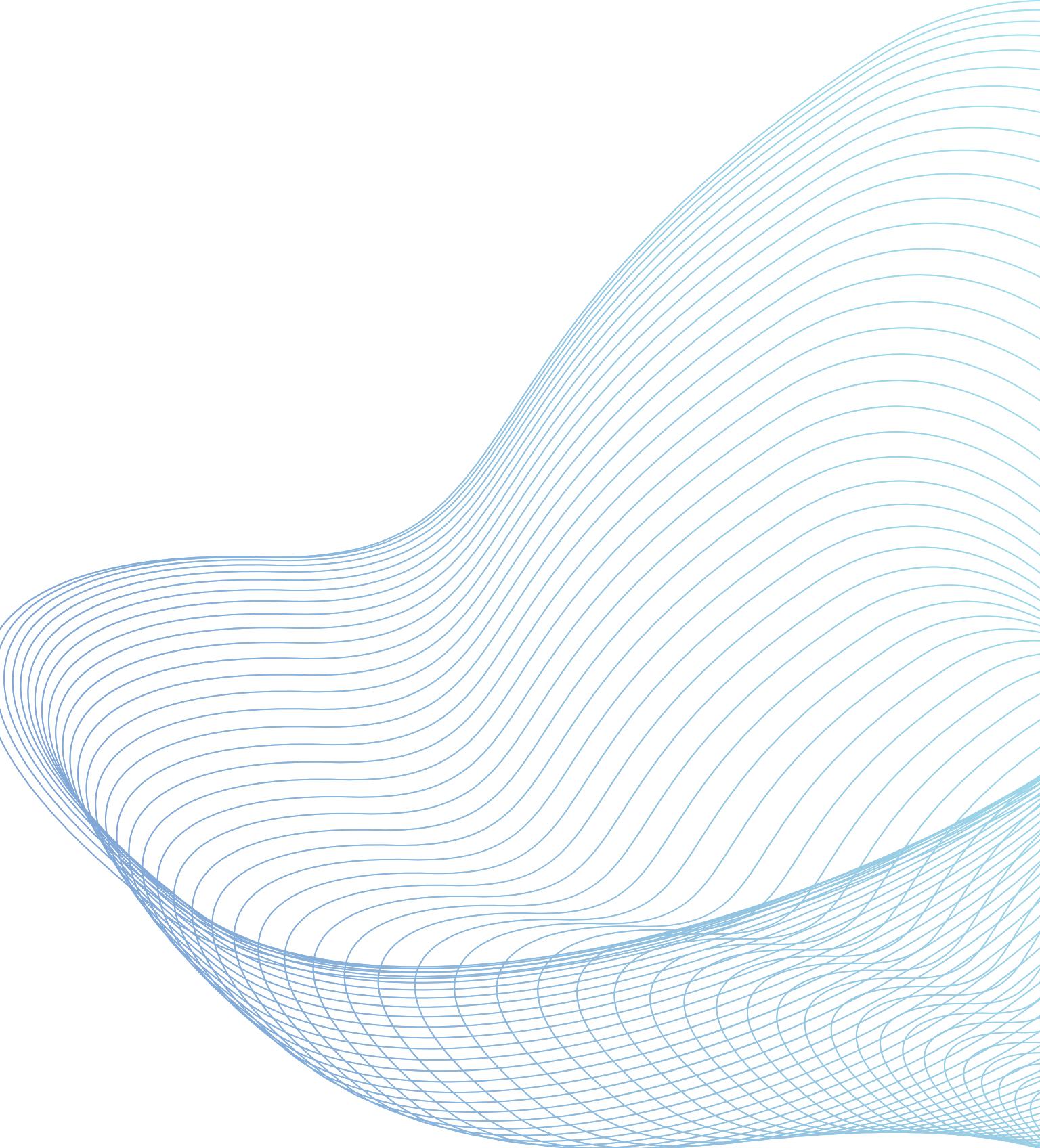


# **DSCI-552**

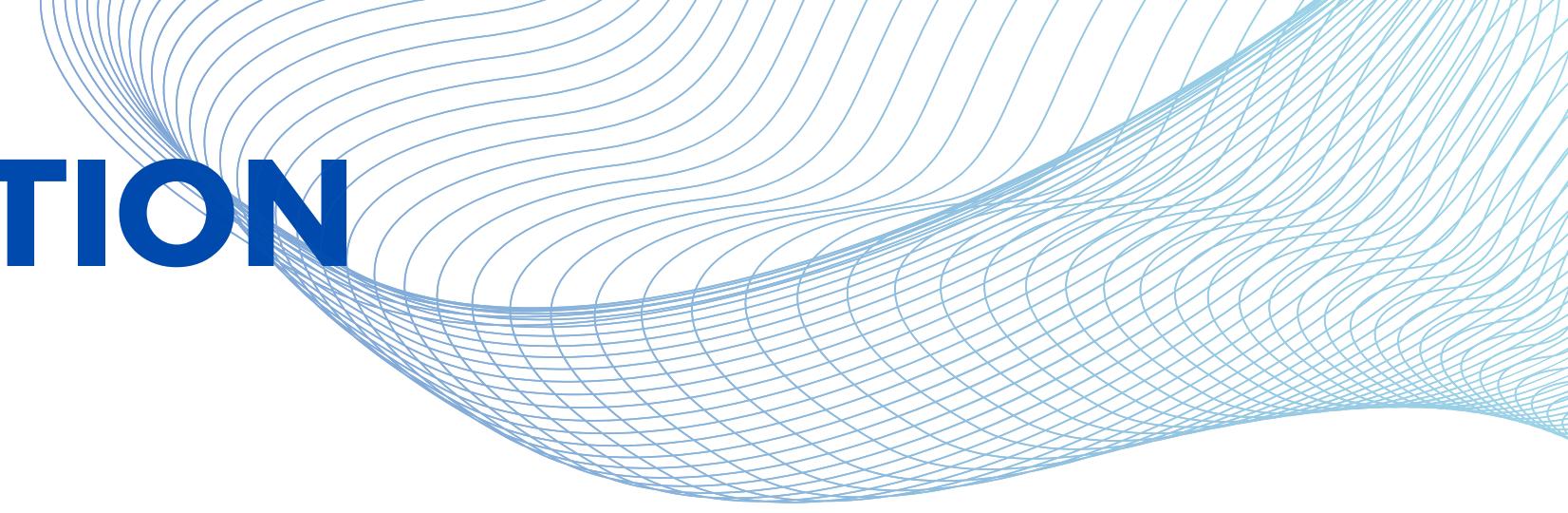
# **PROJECT**

## **GROUP-6**

- Aanandhi Sonduri Panthangi
- Prasad Malwadkar
- Shivani Rajesh Shinde
- Shreya Padmanabhan



# INTRODUCTION



## Home Credit's Role and Vision

1

- **Founded in 1997:** Home Credit is an established international consumer finance provider dedicated to responsible lending, particularly focusing on individuals with little or no credit history.
- **Commitment to Inclusion:** The company is committed to expanding financial inclusion and ensuring a positive, safe borrowing experience for the unbanked population.

## The Challenge of Financial Inclusion

2

- **Absence of Credit History:** Many individuals, especially younger ones or those preferring cash transactions, lack a traditional credit history, leading to frequent loan denials.
- **Critical Need for Accurate Data:** Effective loan repayment determination relies on robust data, which is often lacking for these underserved groups.

# Technological Solution Through Data Science

3

- **Innovation with Scorecards:** Consumer finance providers use advanced statistical and machine learning methods to develop predictive models, known as scorecards, for assessing loan risks.
- **Dynamic Adaptation Required:** The ever-changing behaviors of clients necessitate ongoing updates to these models to maintain their accuracy and relevance.

## Balancing Stability with Performance

4

- **Importance of Model Stability:** Stability in predictive models ensures reliable risk assessments over time, which is essential to prevent an increase in loan defaults.
- **Achieving Optimal Performance:** The development and implementation of these models require a careful balance between their stability and performance to ensure effective deployment.

# KEY ACCOMPLISHMENTS

Created a model to predict which clients are more likely to default on their loans with an accuracy of 57.8%.

## DSCI-552-Group6

Python · Home Credit - Credit Risk Model Stability

Notebook   Input   Output   Logs   Comments (0)   Settings



**Competition Notebook**

Home Credit - Credit Ris...

Run

778.1s - GPU P100

Public Score

0.578

Best Score

0.578 V5

⌚ 5 of 5

# DATASET DESCRIPTION

1

## Base Tables and Dataset Composition

### Fundamental Data:

- Base Tables: Store essential identifiers and basic information, which are fundamental for data linkage and preliminary analysis.
- Examples: `train_base.csv`, `test_base.csv` – these are pivotal for initiating all analyses.

### Depth-Specific Data Layers:

- Static Data (Depth 0): Consists of non-time-variant data from both internal and external sources.
- Historical Data (Depth 1): Contains records detailing past interactions and behaviors.
- Detailed Historical Data (Depth 2): Provides granular insights into detailed historical behaviors.

2

## Data Formats and Accessibility

### File Formats:

- Available in both CSV and Parquet formats, enhancing accessibility and processing efficiency across different computing platforms.

### Mirrored File Structure:

- The structure of the training and testing datasets mirrors each other, facilitating realistic testing environments and model validation.

# 3

## Key Features and Data Transformations

### Essential Columns:

- Identifiers and Metrics: Including `case_id`, `date_decision`, `WEEK_NUM`, `MONTH`, and `target`.
- Indexing for Depth: `num_group1` and `num_group2` are used for navigating through historical data layers.

### Transformations Applied:

- Transformation Types: Days Past Due (P), categorical masking (M), amount adjustments (A), date transformations (D), Unspecified Transform(T) and Unspecified Transform(L).

# 4

## Usage and Integration of Data for Predictive Analysis

### Unique Identifier (`case_id`):

- Facilitates the seamless integration of data across various tables, ensuring coherent data merging for comprehensive analysis.

### Data Layering:

- Utilizes `num_group1` and `num_group2` for aggregating historical data, enhancing the depth of analysis possible with the dataset.

## Predictive Analysis Enhancement

- **Foundation for Modeling:** Base tables serve as the backbone for the entire predictive modeling process, underpinning the development of models that estimate default risks.
- **Comprehensive Data Utilization:** By integrating diverse data sources, the dataset allows for a holistic view of client profiles, crucial for accurate risk prediction.



# DATA WRANGLING

## Pipeline function - Data Preprocessing

### **drop\_unnecessary\_columns:**

- Drops columns that are unnecessary for analysis to reduce complexity and improve model performance. (date columns)
- Drop columns with more than 70% missing values to reduce noise and improve data quality.
- Drop columns with only one unique value or more than 200 unique values which helps in reducing overfitting and improving model generalization.
- `if (freq == 1) | (freq > 200)`

# AGGREGATOR CLASS - FEATURE ENGINEERING

## 1 Numerical Features

- Aggregates numerical features to extract useful information for statistical analysis and modeling.

## 2 Date Features

- Aggregates date features to extract temporal patterns and significant for time-based analysis and modeling.

## 3 String, Count, and Other Features

- Aggregates string features to extract textual patterns.
- Aggregates count features to summarize frequency.
- Aggregates other features to extract miscellaneous information.
- Aggregating these features for comprehensive analysis and modeling.

# FEATURE ENGINEERING AND MEMORY OPTIMIZATION

## Feature engineering process:

- Add month and weekday features based on "date\_decision".
- Join additional depth DataFrames.
- Handle dates using the Pipeline method.

## Memory optimization techniques:

- Reduce memory usage: Iterate through all columns of a dataframe and modify the datatype to reduce memory usage.

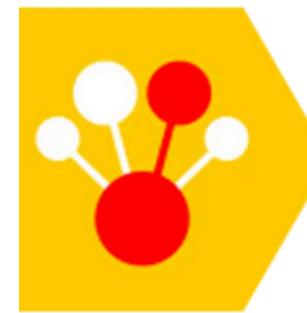


Importance of these functions in efficiently handling large datasets and optimizing memory usage

# OUR APPROACH: CATBOOST CLASSIFIER

## What is CatBoost?

- Open-source library for gradient boosting on decision trees
- Uses ordered boosting to optimize support of categorical features without the need for extensive preprocessing
- Builds symmetric tree architecture which serves as regularization and controls overfitting

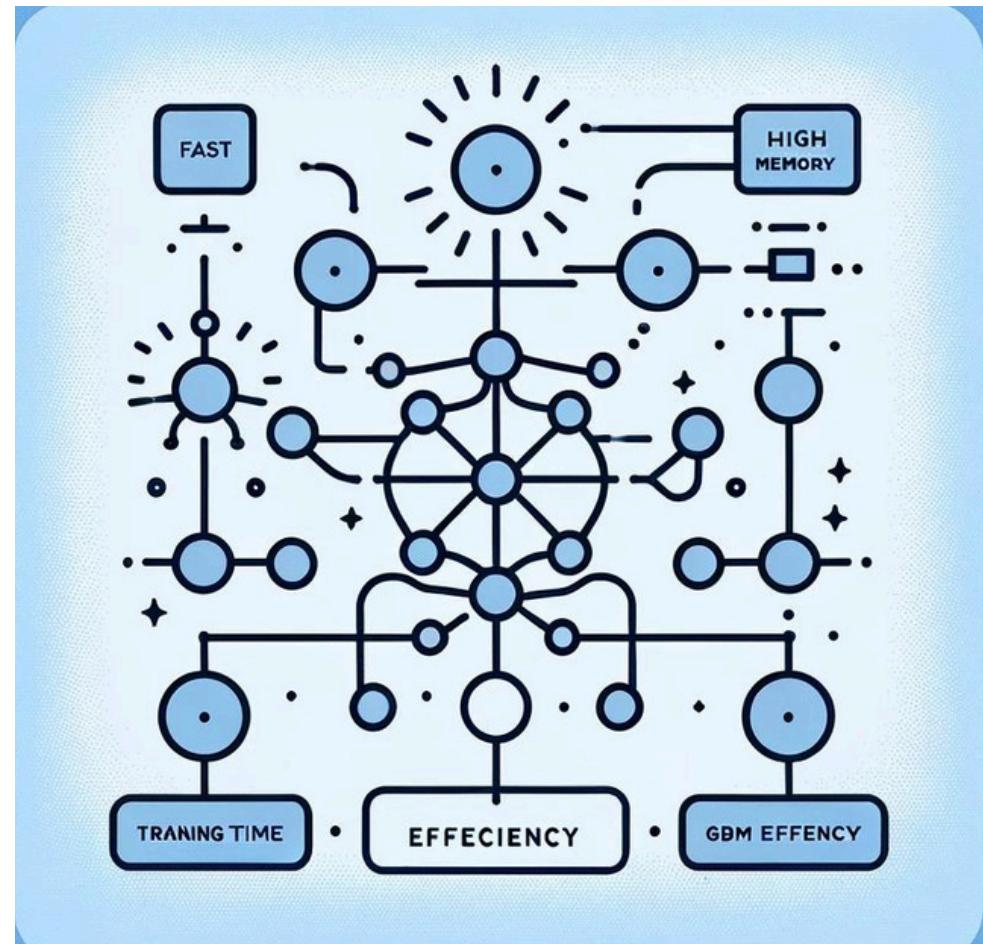


CatBoost

# OUR APPROACH: CHOOSING A MODEL

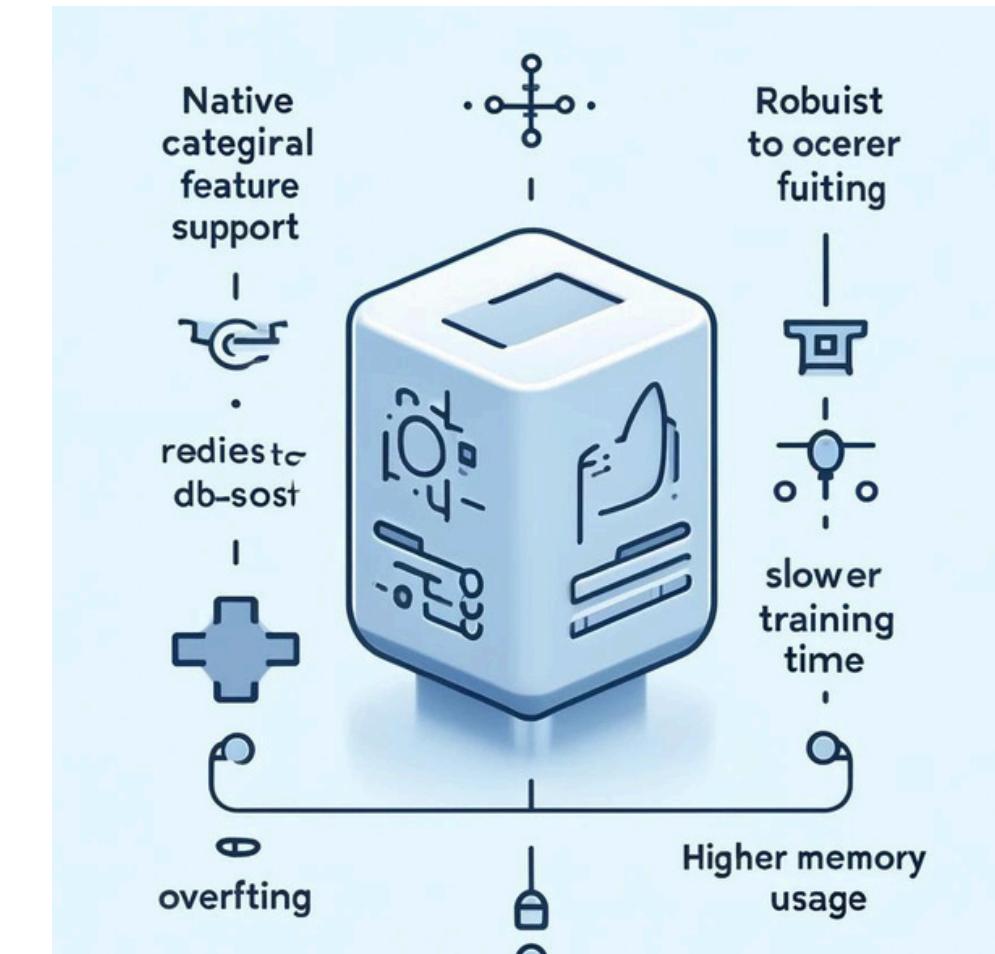
## LIGHT GBM

- Requires extensive feature engineering
- Vulnerable to overfitting
- Fast training time
- High memory efficiency



## CATBOOST

- Native categorical feature support
- Robust to overfitting
- Slower training time
- Higher memory usage



# PREPROCESSING AND FEATURE SELECTION FOR TRAINING DATA

1

## Data Loading and Preprocessing:

- Performing Feature Engineering on the dictionary that stores different dataframes generated from reading parquet files
- Using Pipeline for Data Preprocessing.
- Using Aggregator to extract features.

2

## Handling Missing Values:

- Identifying columns with missing values and grouping them based on the number of missing values.
- Iterating through the grouped missing value columns and applying correlation-based grouping.

3

## Feature Selection:

- Creating a list of selected features based on the grouping and selection criteria.
- Including categorical columns in the selected feature list.

# PREPROCESSING AND FEATURE SELECTION FOR TEST DATA

1

## Data Loading and Preprocessing

Loading and preprocessing test data performing Feature Engineering.  
Deleting unnecessary variables and garbage collecting to optimize memory usage

3

## Memory Optimization

Converting the test data to pandas DataFrame format. Reducing memory usage of the test DataFrame. Performing garbage collection to release memory resources.

## Feature Selection

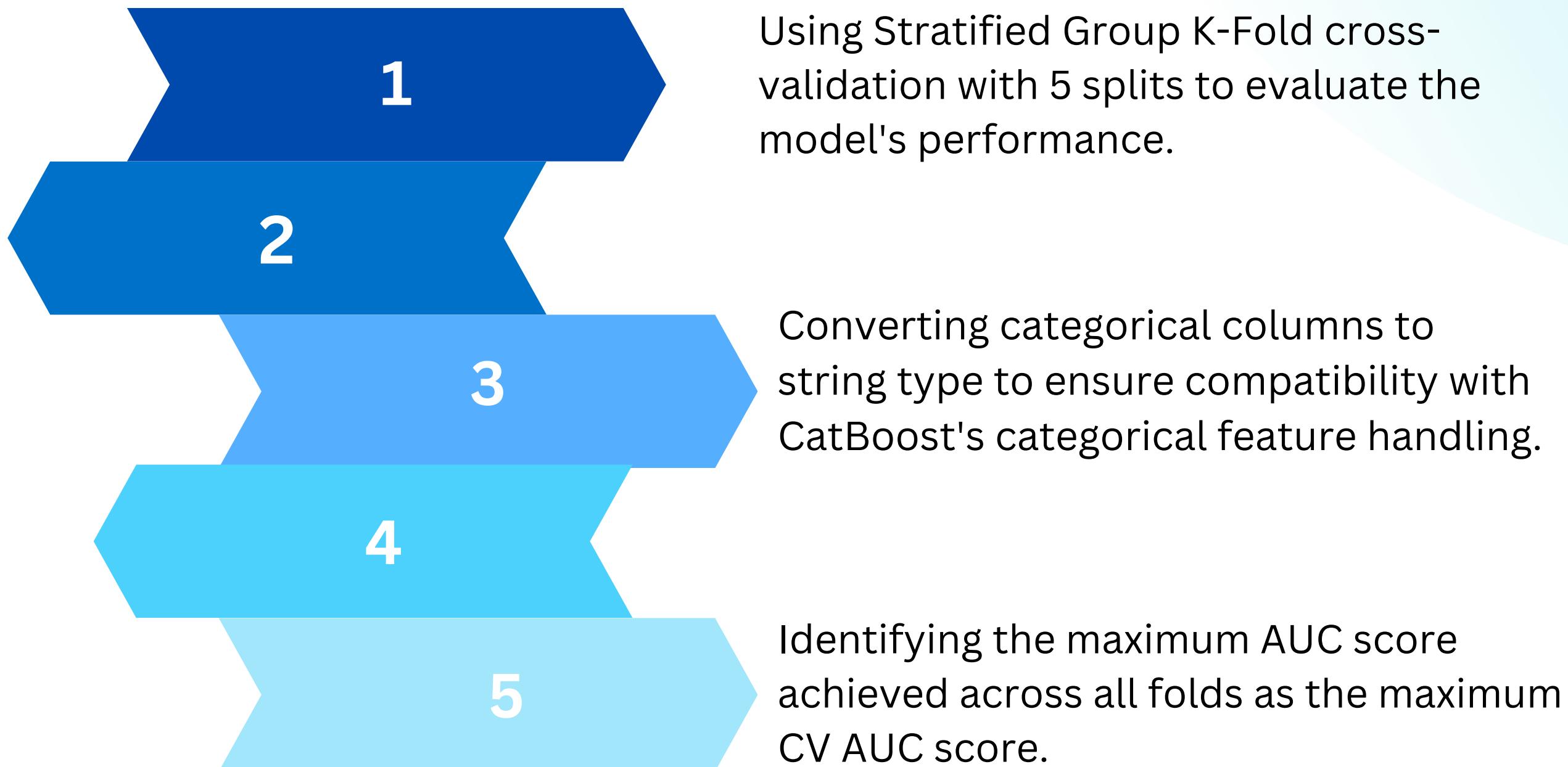
2

Selecting features for the test data that are present in the training data.  
Ensuring consistency between the features used in training and testing

# OUR APPROACH - CATBOOST CLASSIFIER

Dropping unnecessary columns such as "target", "case\_id", and "WEEK\_NUM" from the training data

Hyperparameter Tuning, Initializing and Training CatBoost Classifier



# OUR APPROACH - PARAMETERS USED

## Evaluation Metrics used

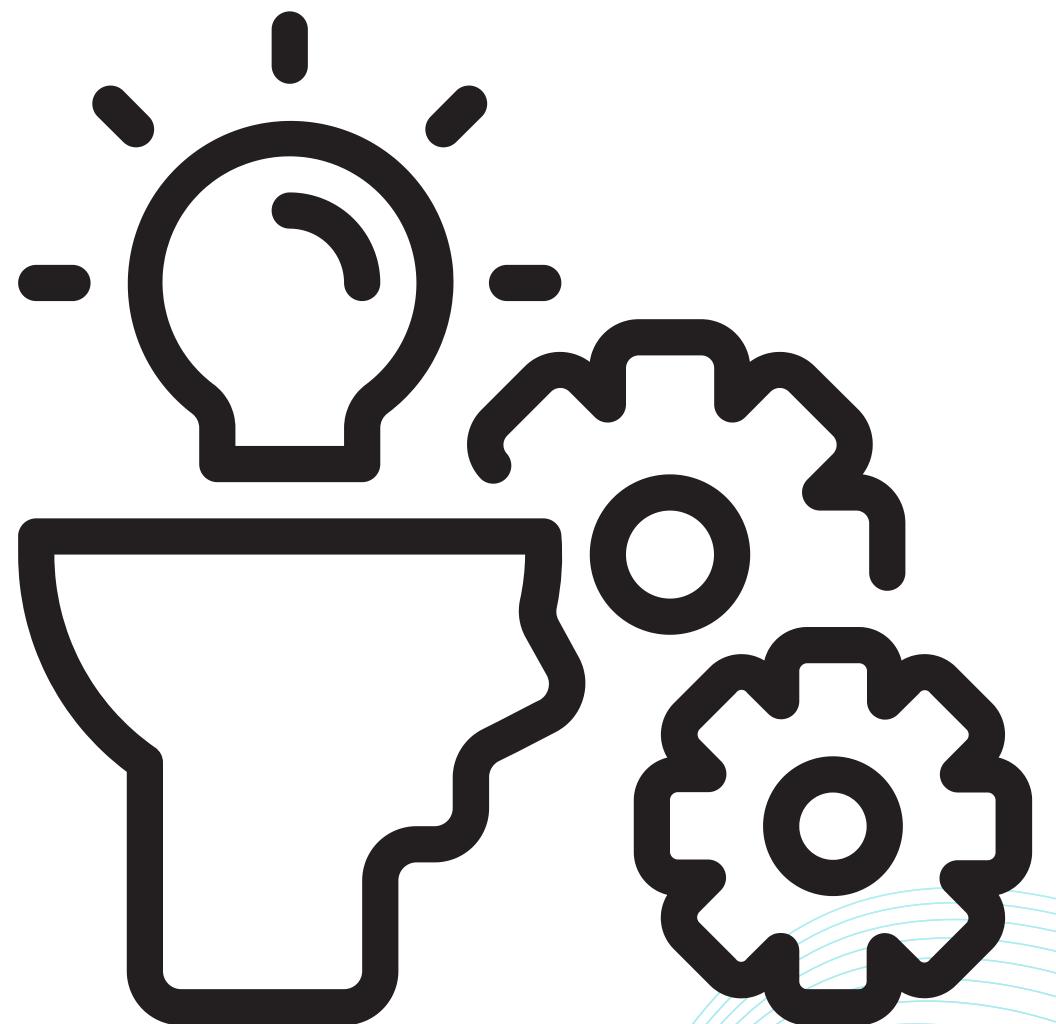
- Area Under the Receiver Operating Characteristic (ROC) Curve (AUC)

## Learning Rate

- 0.03

## AUC Score

- 0.82



# OUR APPROACH - MODEL FITTING

1

## Custom Voting Model Definition

- Capable of aggregating predictions from a list of base estimators

2

## Model Initialization and Training

- Initializing the VotingModel with a list of fitted models (`fitted_models`), typically obtained from cross-validation or individual model training.

3

## Prediction Generation

- Generating predictions (`y_pred`) using the custom VotingModel for the test data.

# OUR APPROACH - ABLATION STUDIES

1

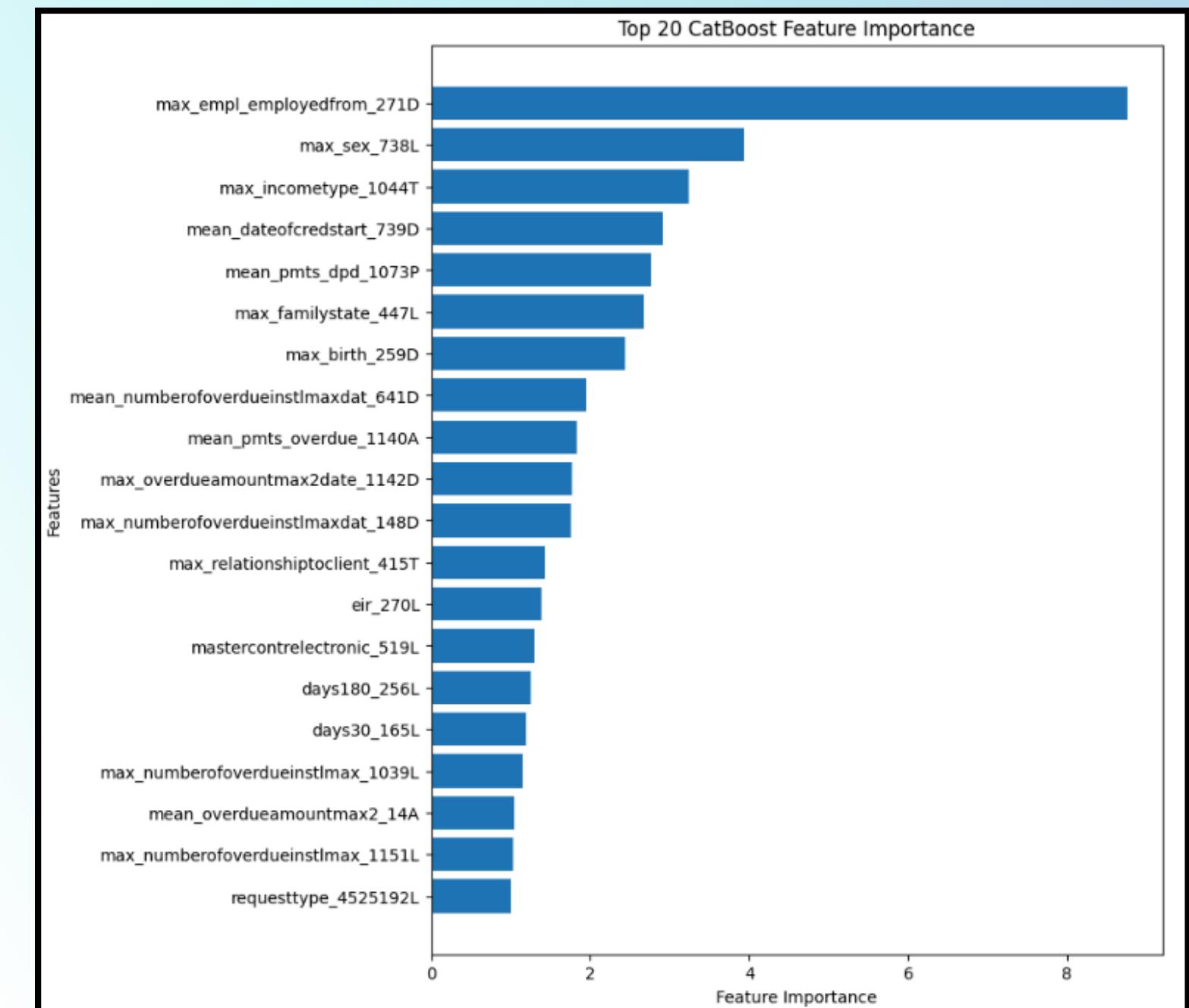
## Feature Ablation

- Determined feature importance based on change in prediction values

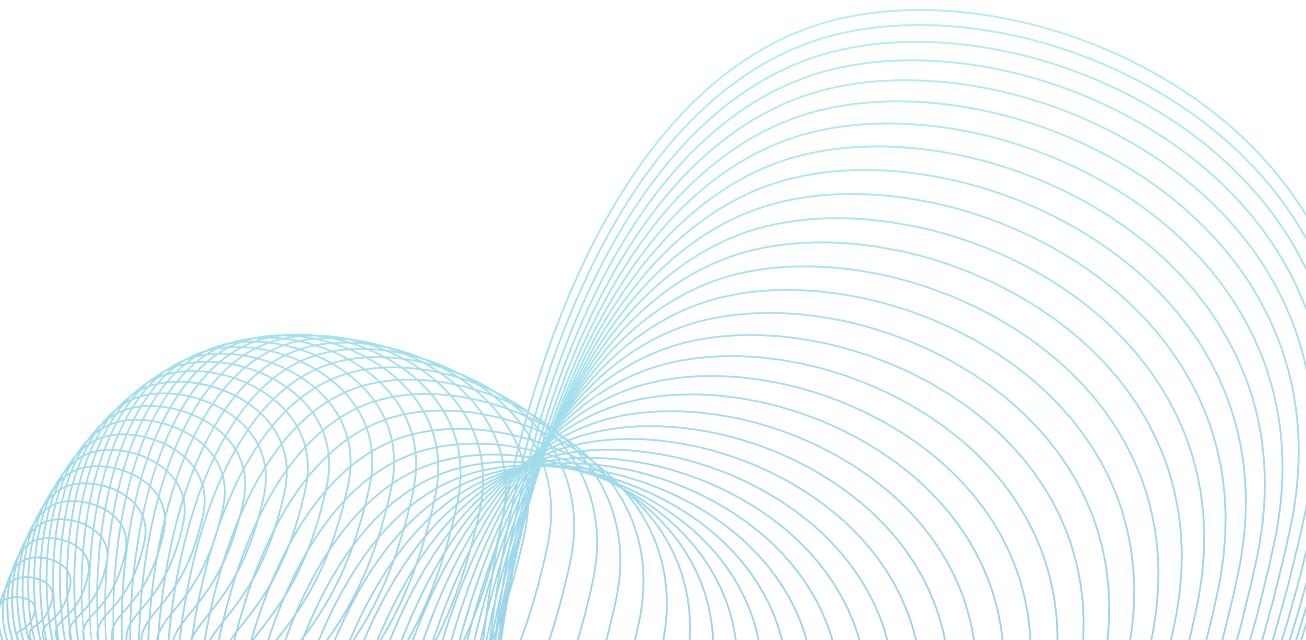
2

## Complexity Ablation

- Dropped unnecessary or excessively sparse columns
- Evaluated the impact of dropping columns with regards to overfitting and noise



# Q & A



# THANK YOU

