COMPUTER PROGRAMMING AND
ARTIFICIAL INTELLIGENCE

Telecom Churn Prediction

Kaung Nyo Lwin st125066
Patsachon Pattakulpong st124952

Final Presentation

Methodology

Model Evaluation Results

Discussion

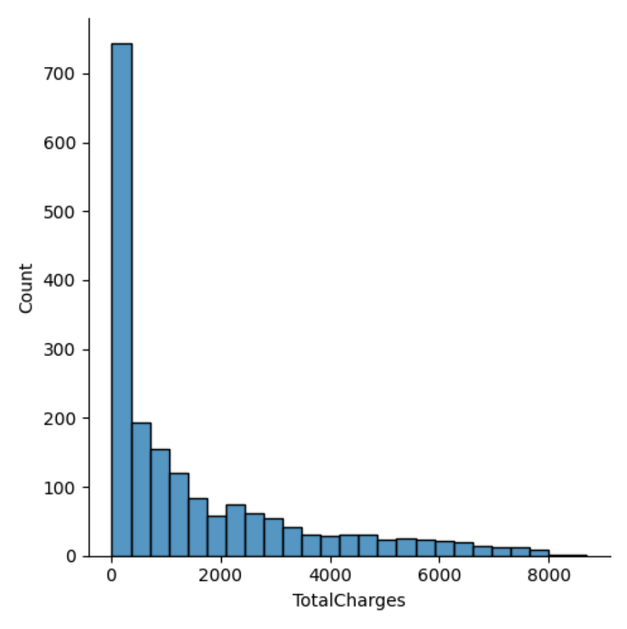
Conclusion

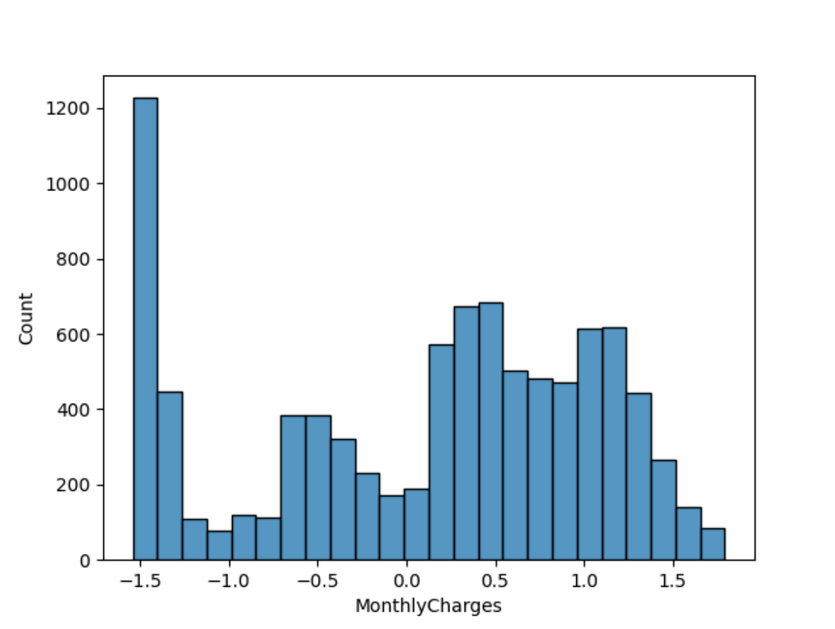
Methodology

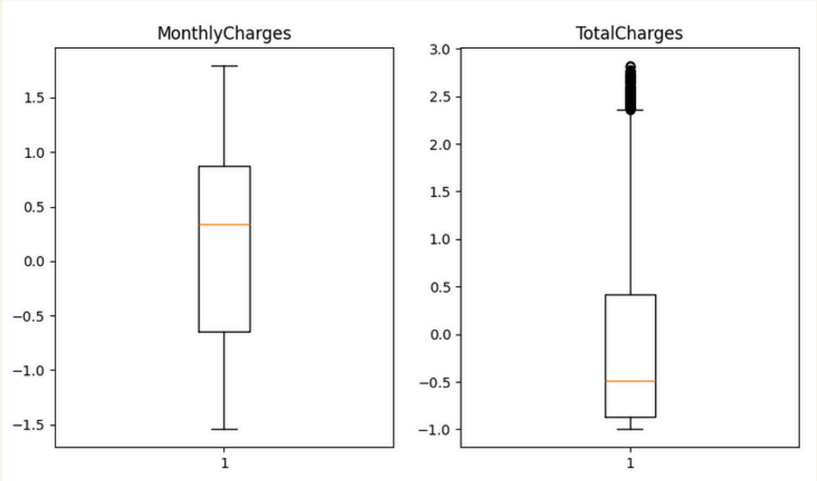
The methodology for this project encompasses exploratory data analysis (EDA), preprocessing, machine learning model and Pipeline with a focus on aligning predictive outcomes with business goals.

EDA (Exploratory Data analysis)

- Most features are categorical and need encoding; only three numerical features: **tenure, monthly charges, and total charges.**
- Most features are free from missing values.
- Distribution analysis (using histograms and boxplots) shows a class imbalance

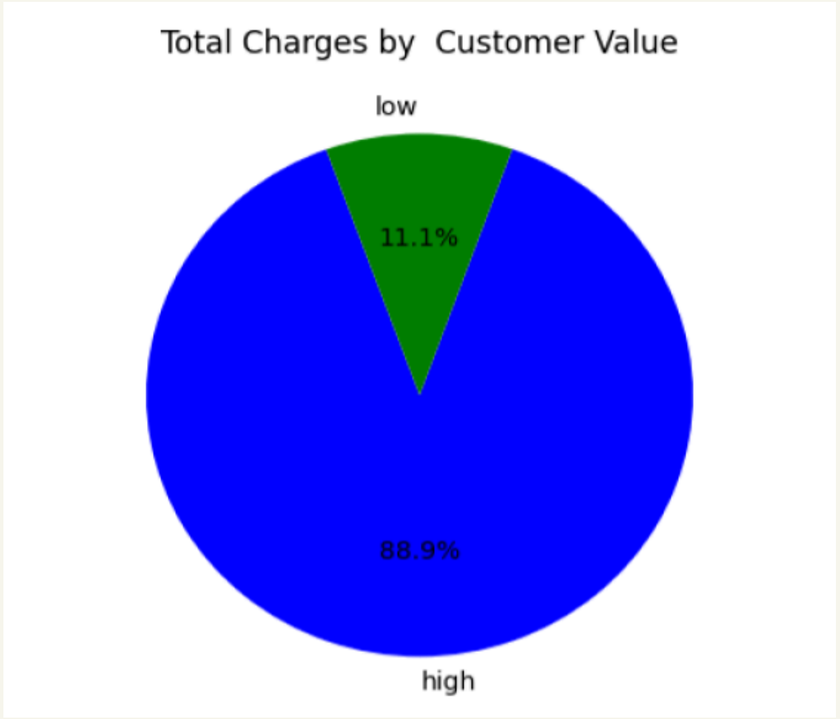






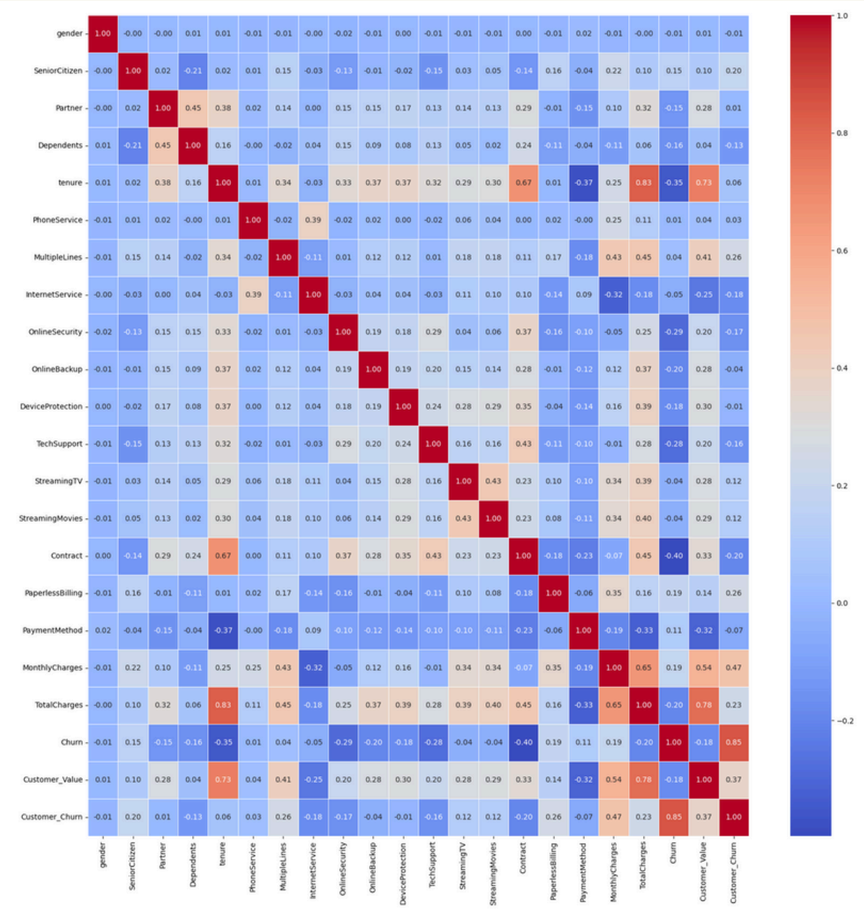
EDA (Exploratory Data analysis)

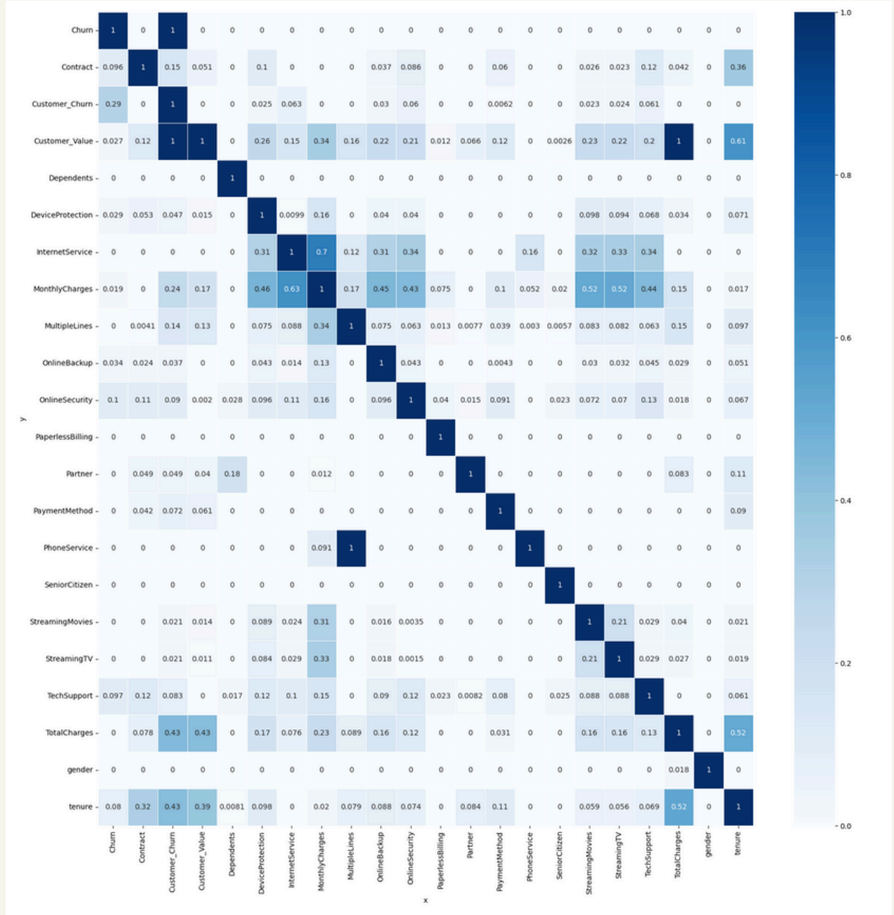
- The model will be evaluated using business-oriented metrics derived from total customer revenue.
- Analysis of total charges reveals
  - Median total charges: 1394.55
  - Over 80% of total revenue comes from customers with total charges above the median.



EDA (Exploratory Data analysis)

- Feature selection was performed **using correlations** and **predictive power score**.
- Heatmaps were plotted to identify relationships among features.
  - The **tenure feature** was excluded due to high correlation with other features.
  - **All other features were retained.**





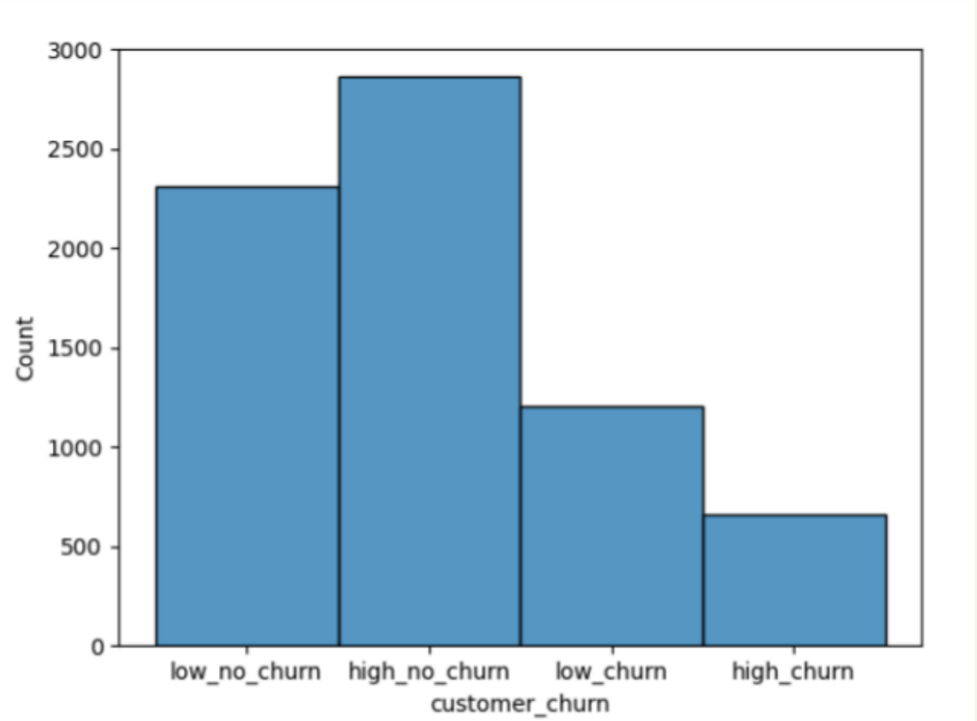
Pre-Processing

Imputing missing values

- Missing values in total charges were imputed using values from the **monthly charges** feature, assuming these customers have not been charged yet.
- All other features are free from missing values.

Feature Engineering

- Created two new columns for model evaluation
  a. **Customer_value**: Categorized as "high" (above median) or "low" (below median).
  b. **Customer_churn**: Derived from Customer_value and churn, forming four classes: low_churn, high_churn, low_no_churn, high_no_churn



Pre-Processing

Train-Test Split

- Used 10% of data for the test set **(~800 samples)**.

Scaling

- Applied StandardScaler() to scale numerical features, as charges are unbounded despite right-skewed distributions.

Encoding

- Used **label encoding** for categorical features with 2-3 unique values instead of one-hot encoding.

Modeling

- **Model Selection**
- **Upsampling with SMOTE**
- **Hyperparameter Tuning**
- **Experiment Class Weights**

Cross_val with four classes training data

1. Logistic Regression
2. Support Vector Classifier
3. Decision Tree Classifier
4. Random Forest Classifier
5. AdaBoost Classifier
6. **Gradient Boosting Classifier**

Modeling

- Model Selection
- Upsampling with SMOTE
- Hyperparameter Tuning
- Experiment Class Weights

```
from imblearn.over_sampling import SMOTEN
smote = SMOTEN(random_state=0)

X_train_up,y_train_up = smote.fit_resample(X_train,y_train)
```

Modeling

- Model Selection
- Upsampling with SMOTE
- Hyperparameter Tuning
- Experiment Class Weights

Tuning one hyperparameter at a time

```
# gradeint boosting parameters
gbc_parameters = {'loss': 'log_loss',
'learning_rate': 0.1,
'min_samples_split': 0.17272727272727273,
'min_samples_leaf': 0.17272727272727273,
'max_depth': 3,
'max_features': 'log2',
'criterion': 'friedman_mse',
'subsample': 0.95,
'n_estimators': 50}

# histogram-based gradeint boosting parameters
hgc_parameters = {'learning_rate': 0.1,
        'min_samples_leaf': 10,
        'max_depth': None,
        'max_features': 0.5,
        'l2_regularization': 0.1,
        'max_iter': 130}
```

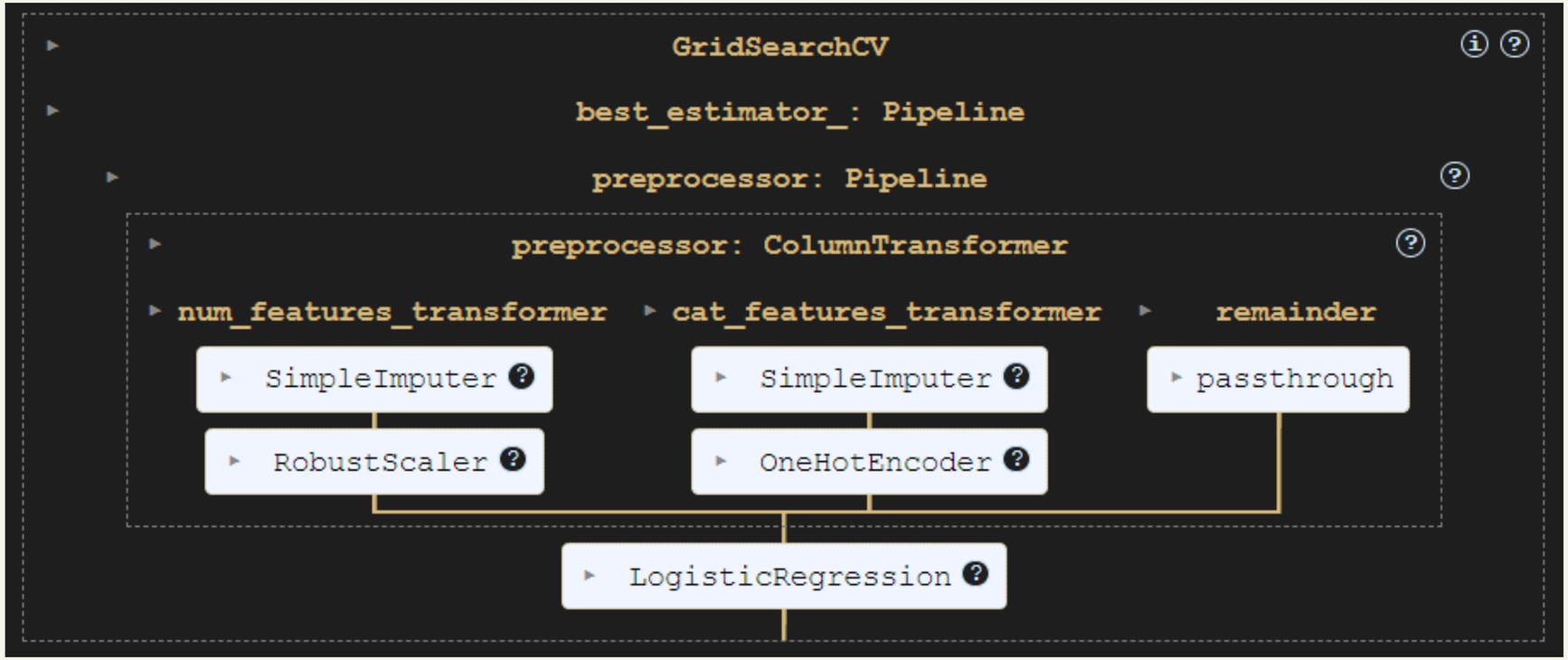
Modeling

- Model Selection
- Upsampling with SMOTE
- Hyperparameter Tuning
- Experiment Class Weights

Train HGboost with class weights

```
cls_weights1 = {0 : 2, 1 : 1, 2 : 1, 3 : 1}
cls_weights2 = {0 : 3, 1 : 1, 2 : 2, 3 : 1}
cls_weights3 = {0 : 4, 1 : 1, 2 : 1, 3 : 1}
cls_weights4 = {0 : 4, 1 : 1, 2 : 1, 3 : 1}
```

Pipeline


GridSearchCV
best_estimator_: Pipeline
preprocessor: Pipeline
preprocessor: ColumnTransformer
num_features_transformer
cat_features_transformer
remainder
SimpleImputer
SimpleImputer
passthrough
RobustScaler
OneHotEncoder
LogisticRegression

Model Evaluation Results

The model evaluation focuses on minimizing potential revenue losses from false negatives (misclassified non-churners), especially among high-value customers, whose departure poses a significant financial risk.

Model Evaluation Metrics

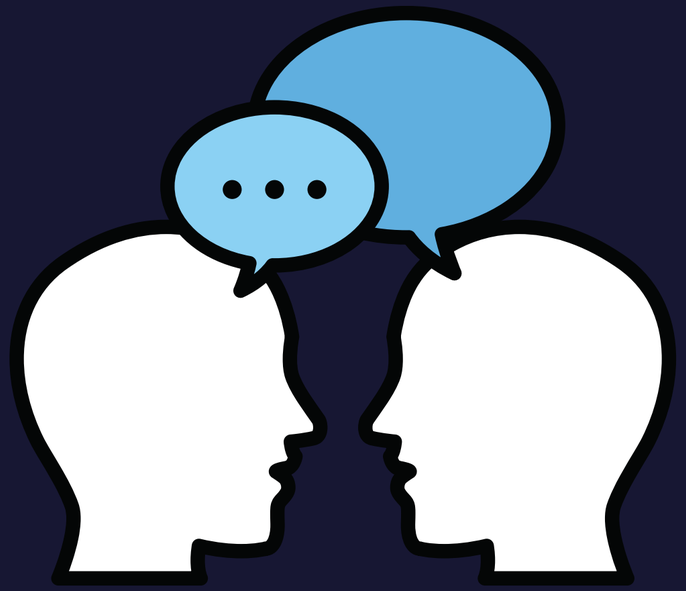
$$Potential\ Risk\ in\ Currency\ = \sum_{i=0}^{m} (TC)\ if\ FN$$

$$Risk\ Factor\ = \frac{Potential\ Risk\ in\ Currency}{\sum_{i=0}^{m} (TC)\ where\ target = "churn"}$$

	Models	Potential Risk in Currency	Risk Factor	Number of wrong predictions	False Negatives
0	gbc_4class_upsample	168376.30	0.520904	191	64
1	hgbc_4class_upsample_class_weight4	185005.20	0.572349	190	77
2	hgbc_4class_upsample_class_weight3	186503.45	0.576984	193	69
3	hgbc_4class_upsample_class_weight1	187950.35	0.581460	168	76
4	hgbc_4class_upsample_class_weight2	188011.10	0.581648	197	65
5	gbc_2class_upsample	190178.70	0.588354	178	63
6	hgbc_4class_upsample	236796.05	0.732573	167	90
7	hgbc_4class	246049.50	0.761201	157	89
8	gbc_2class	247608.90	0.766025	147	100
9	gbc_4class	277095.30	0.857247	145	105

Evaluation Results

- Variations of Gradient Boosting Classifier (GBC) and Histogram-Based Gradient Boosting Classifier (HGBC) were assessed.
- Upsampling techniques enhance performance by reducing revenue loss, e.g., gbc_4class_upsample shows a lower risk factor (0.5209) compared to gbc_4class (0.8572).
- Class weight adjustments improve outcomes, as seen in hgbc_4class_upsample_class_weight1, achieving a lower risk factor (0.5814) and potential risk (187950.34).
- Trade-offs include increased false negatives and risk factors in models like gbc_2class (0.7660), highlighting the need to balance accuracy with financial impact.



Discussion

EVALUATION METRIC

Risk factor shows how well the model predicts on the high-valued customer churns

PERFOMANCE

Risk factor - improvement (0.5)

Reduce the risk of potential loss of revenue

RECATEGORIZING AND UPSAMPLING

Effective for unbalanced data and the interest is in least dominant class

Conclusion

- A MORE ROBUST MODEL
- EVALUATION METRIC
- A GENERAL APPROCH FOR UMBALANCED DATA