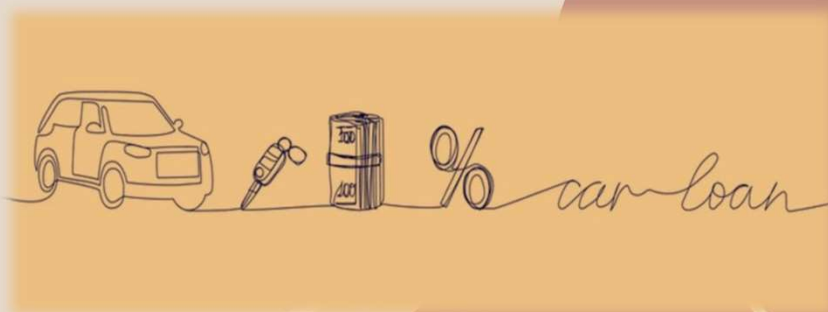
CAR LOAN DEFAULTER

Presentation by Group: 04
Khushi Agrawal
Eksimar Kaur
Diksha Yadav
Tejaswini Pathak
Ankita Roy



The Data Description

```
[ ] df=pd.read_csv(r'/content/drive/MyDrive/CAPSTONE/Train_Dataset.csv')
```

```
df.head()
```

	ID	Client_Income	Car_Owned	Bike_Owned	Active_Loan	House_Own	Child_Count	Credit_Amount	Loan_Annuity	Accompany_Client	Client_Income_Type	Client_Education
0	12142509	6750	0.0	0.0	1.0	0.0	0.0	61190.55	3416.85	Alone	Commercial	Secondary
1	12138936	20250	1.0	0.0	1.0	NaN	0.0	15282	1826.55	Alone	Service	Graduation
2	12181264	18000	0.0	0.0	1.0	0.0	1.0	59527.35	2788.2	Alone	Service	Graduation dropout
3	12188929	15750	0.0	0.0	1.0	1.0	0.0	53870.4	2295.45	Alone	Retired	Secondary
4	12133385	33750	1.0	0.0	1.0	0.0	2.0	133988.4	3547.35	Alone	Commercial	Secondary

```
print(f' Number of Columns :',df.shape[1])
print(f' Number of Rows :',df.shape[0])
df.info()
```

```
 Number of Columns : 40
 Number of Rows : 121856
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121856 entries, 0 to 121855
Data columns (total 40 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   ID                          121856 non-null  int64
 1   Client_Income               118249 non-null  object
 2   Car_Owned                   118275 non-null  float64
 3   Bike_Owned                  118232 non-null  float64
 4   Active_Loan                 118221 non-null  float64
 5   House_Own                   118195 non-null  float64
 6   Child_Count                 118218 non-null  float64
 7   Credit_Amount               118224 non-null  object
 8   Loan_Annuity                117044 non-null  object
 9   Accompany_Client            120110 non-null  object
 10  Client_Income_Type          118155 non-null  object
 11  Client_Education            118211 non-null  object
 12  Client_Marital_Status       118383 non-null  object
 13  Client_Gender               119443 non-null  object
 14  Loan_Contract_Type          118205 non-null  object
 15  Client_Housing_Type         118169 non-null  object
 16  Population_Region_Relative  116999 non-null  object
 17  Age_Days                    118256 non-null  object
 18  Employed_Days               118207 non-null  object
 19  Registration_Days           118242 non-null  object
 20  ID_Days                     115888 non-null  object
 21  Own_House_Age               41761 non-null   float64
 22  Mobile_Tag                  121856 non-null  int64
 23  Homephone_Tag               121856 non-null  int64
 24  Workphone_Working           121856 non-null  int64
 25  Client_Occupation           80421 non-null   object
 26  Client_Family_Members       119446 non-null  float64
 27  Cleint_City_Rating          119447 non-null  float64
 28  Application_Process_Day     119428 non-null  float64
 29  Application_Process_Hour    118193 non-null  float64
 30  Client_Permanent_Match_Tag  121856 non-null  object
 31  Client_Contact_Work_Tag     121856 non-null  object
 32  Type_Organization           118247 non-null  object
 33  Score_Source_1              53021 non-null   float64
 34  Score_Source_2              116170 non-null  float64
 35  Score_Source_3              94935 non-null   object
 36  Social_Circle_Default       59928 non-null   float64
 37  Phone_Change                118192 non-null  float64
 38  Credit_Bureau               103316 non-null  float64
 39  Default                     121856 non-null  int64
dtypes: float64(15), int64(5), object(20)
memory usage: 37.2+ MB
```

The dataset contains
121,856 rows and 40 columns, with a mix of numerical and categorical data, and some columns with missing values.

EDA - Overview

•The dataset has **121,856 records** and **40 features**, with a mix of numerical and categorical data.
•**Data Cleaning**:
•Missing values were handled using:
  •**Mode** for categorical data.
  •**Median** for numerical data.
•Features with excessive missing values, like Own_House_Age and Score_Source_1, were dropped.
•**Outliers and Skewness**:
•Features like income, loan amount, and family size had significant **positive skewness** with extreme outliers.
•Features like age and registration days showed nearly symmetric distributions.
•**Encoding**:
•Categorical variables were converted into numerical representations using dummy encoding.

EDA SUMMARY

•**Numerical Features**:
•Most clients have low income and small loan amounts, with few high-value outliers.
•Short employment tenures are common, and a majority of clients come from less populated regions.
•**Categorical Features**:
•Most clients:
  •Do not own a car or bike.
  •Own a house and live in their own homes.
  •Are married and have secondary education.
•Tuesday saw the highest number of loan applications.
•**Correlation Analysis**:
•**Family Size ↔ Child Count**: Strong positive correlation.
•**Loan Amount ↔ Annuity**: Moderate positive correlation.
•Weak correlation between other variables, suggesting low multicollinearity.

STATISTICAL ANALYSIS

•**Dependency Analysis**:
•Features like Car_Owned, Client_Income_Type, Client_Education, and Loan_Contract_Type are significantly associated with loan default.
•Features like Bike_Owned, Active_Loan, and Mobile_Tag show no dependency on defaults.
•**Chi-Square Test Results**:
•Categorical variables such as Client_Gender, Client_Marital_Status, and Client_Housing_Type significantly influence the likelihood of default.
•Employment-related variables like Type_Organization also show strong dependency with defaults.
•**Class Imbalance Observations**:
•Defaulted loans (minority class) are significantly underrepresented, affecting statistical relationships and requiring adjustments during model building.

DATA PREPARATION STEPS

•**Encoding**:
•Applied **N-1 dummy encoding** to convert categorical variables into numerical representations for model compatibility.
•**Concatenation**:
•Combined **numerical** and **encoded categorical columns** into a single dataset for seamless training and evaluation.
•**Train-Test Split**:
•Split the data into **80% training** and **20% testing** sets to ensure robust model evaluation and avoid overfitting.

MODEL COMPARISON AND EVALUATION

LOGISTIC REGRESSION

- The model has high precision for class 0 (0.94) but struggles with class 1, showing low recall (0.53) and a very low F1 score for class 1 (0.17). Its overall accuracy is 59%, but it is biased towards predicting class 0. This results in significantly higher support for class 0 and poor performance for class 1.

DECISION TREE

- The model has high precision for class 0 (0.94) but low recall for class 1 (0.53), with a very low F1 score for class 1 (0.17). Its overall accuracy is 59%, and it is biased towards predicting class 0, leading to poor performance for class 1.

RANDOM FOREST

- The model performs well on class 0 with high precision and recall but fails to predict class 1, with a recall of 0.00. While overall accuracy is 92%, the poor performance for class 1 reduces the macro F1-score to 0.48, indicating class imbalance and poor generalization.

ADABOOST CLASSIFIER

- The model performs well on class 0 with a 93% F1-score but struggles with class 1, showing low precision and recall around 0.25–0.26. Despite 88% overall accuracy, the imbalance between classes highlights the need for better handling of the minority class.

XGB CLASSIFIER

- The model performs well for class 0 with high precision and recall but struggles with class 1, showing a recall of 0.03 and an F1-score of 0.06. Despite 92% overall accuracy, the imbalance heavily favors class 0, and the macro averages reflect poor performance on the minority class.

GRADIENT BOOSTING CLASSIFIER

- The model performs well for class 0 with high precision and recall but struggles with class 1, showing a recall of 0.03 and an F1-score of 0.06. Despite 92% overall accuracy, the imbalance skews results, and macro averages highlight poor performance on the minority class.

LGBM CLASSIFIER

- The model's performance shows imbalanced class handling, with high false positives for class 1 due to low precision and suboptimal recall for class 0. The macro F1-score of 0.54 highlights difficulty in achieving balanced precision and recall across both classes.

CATBOOST CLASSIFIER

- The model shows strong performance for class 0 with a high F1-score but struggles significantly with class 1 due to low precision and moderate recall, resulting in many false positives. The macro F1-score of 0.64 indicates poor class balance, emphasizing the need for better handling of the minority class.

MODEL TUNING –APROACH

- **Addressing Class Imbalance**:
    - Implemented techniques like **class weighting** and **SMOTE** to balance the minority class.
    - Used **cost-sensitive learning** to prioritize recall for the default class.
- **Parameter Optimization**:
    - Applied **GridSearchCV** and **RandomizedSearchCV** to fine-tune key parameters:
        - For boosting models: learning_rate, max_depth, n_estimators, and scale_pos_weight.
        - For Random Forest: max_depth, min_samples_split, and min_samples_leaf.
- **Threshold Tuning**:
    - Adjusted probability thresholds to improve sensitivity and recall for default predictions.

MODEL TUNING RESULTS

- **LightGBM (Best Model)**:
- Achieved **71% accuracy** and **64% recall**, balancing false positives and false negatives effectively.
- Handles the imbalanced dataset better than other models.
- **Other Models**:
- **XGBoost**: Good accuracy but struggled with recall for defaults.
- **Random Forest**: Overfitting issues and poor performance on minority class.
- **Logistic Regression**: Limited recall improvement despite tuning.
- **Next Steps**:
- Focus on further optimizing LightGBM with advanced techniques like ensemble stacking.
- Explore real-time applications with the tuned model

SCORECARD

Model	Accuracy	Precesion	Recall	F1 Score	Cohen Kappa	ROC AUC
LogisticRegression(class_weight={0: 0.5440076787428291, 1: 6.180826781638347})	0.588339	0.102109	0.528841	0.171169	0.042099	0.581329
DecisionTreeClassifier(class_weight={0: 0.5440076787428291,\n 1: 6.180826781638347},\n max_depth=5)	0.616937	0.131408	0.671261	0.219789	0.098606	0.690881
RandomForestClassifier(max_depth=5, max_features=None, n_estimators=150)	0.919621	0.000000	0.000000	0.000000	0.000000	0.704694

Model	Accuracy	Precesion	Recall	F1 Score	Cohen Kappa	ROC AUC
AdaBoostClassifier(estimator=DecisionTreeClassifier(class_weight={0: 0.5440076787428291,\n 1: 6.180826781638347}),\n learning_rate=0.2, n_estimators=200)	0.876375	0.247847	0.264421	0.255866	0.188530	0.597141
XGBClassifier(base_score=None, booster=None, callbacks=None,\n colsample_bylevel=None, colsample_bynode=None,\n colsample_bytree=None, device=None, early_stopping_rounds=None,\n enable_categorical=False, eval_metric=None, feature_types=None,\n gamma=1, grow_policy=None, importance_type=None,\n interaction_constraints=None, learning_rate=0.2, max_bin=None,\n max_cat_threshold=None, max_cat_to_onehot=None,\n max_delta_step=None, max_depth=None, max_leaves=None,\n min_child_weight=None, missing=nan, monotone_constraints=None,\n multi_strategy=None, n_estimators=150, n_jobs=None,\n num_parallel_tree=None, random_state=None, ...)	0.920113	0.553571	0.031649	0.059874	0.051629	0.755579
LGBMClassifier(class_weight={0: 0.5440076787428291, 1: 6.180826781638347},\n max_depth=5, num_leaves=28)	0.713852	0.166777	0.640633	0.264656	0.157149	0.745606
<catboost.core.CatBoostClassifier object at 0x7d4d38b8a680>	0.879000	0.309615	0.410924	0.353148	0.287858	0.752419

Model	Accuracy	Precesion	Recall	F1 Score	Cohen Kappa	ROC AUC
GradientBoostingClassifier(learning_rate=0.2, max_depth=10, n_estimators=200)	0.923355	0.570983	0.18683	0.281538	0.251888	0.765534

CONCLUSION

Among all the models evaluated, LightGBM stands out as the best-performing model for predicting customer defaults. It achieves a good balance between recall (64%) and accuracy (71%), making it effective in identifying most default cases while maintaining a reasonable overall classification performance. Additionally, LightGBM outperforms other models in handling the trade-off between false positives and false negatives, which is critical in imbalanced datasets like this one.

While its precision (17%) is relatively low, this can be addressed with techniques like oversampling, undersampling, or cost-sensitive learning to further improve the model's ability to correctly classify positive cases. Overall, LightGBM is a robust and reliable model compared to others in this analysis, demonstrating its suitability for this predictive task with further refinement and optimization

LIMITATION

1. Class Imbalance Issue: Most models show low precision, recall, or F1-scores for predicting defaults (positive cases), which indicates that the class imbalance in the dataset has impacted their ability to correctly classify minority cases.

2. LowPrecision for Positive Class: Many models, such as Logistic Regression, Decision Tree, and Random Forest, demonstrate low precision, meaning that a high proportion of predicted positive cases are false positives.

3. LowRecall for Positive Class: Models like Random Forest and XGBoost have particularly low recall, indicating that they fail to identify a significant number of actual default cases.

4. Overfitting or Bias: Models such as Random Forest, XGBoost, and Gradient Boosting have high accuracy scores but perform poorly in terms of recall and F1-score for the positive class, suggesting potential overfitting or bias toward the majority class.

5. Model-Specific Weaknesses: RandomForest failed to identify any positive cases (0% recall). AdaBoost and Logistic Regression have moderate F1-scores but still struggle with precision.

6. Trade-Off Between Precision and Recall: Models like LightGBM and Gradient Boosting achieve good recall but lower precision, reflecting a trade-off that might lead to many false positives.

Thank You