RSM 8421H1 AI and Deep Learning Tools: Individual Assignment Four: Keqing Wang 1006927337

1.Introduction:

1.1 Summarize the Problem:

This model explores the dataset 'promos' and 'customer purchase activities' by using random forest binary classification to improve the model prediction accuracy on customers activism rate. The methodology we used mainly includes handling an imbalanced dataset (PCA and SMOTE), feature engineering (dimensionality enrichment, efficient feature creation, and target encoding), random forest optimized model after efficient hyperparameter tunning.

Experiment results show that our model approaches lift the AUC score efficiently from 0.52 to 0.69, in particular through the feature engineering on the original dataset exploratory variables. Effective analysis through the exploitation of customer behavior-based segmentations and geographical characteristics has been inherently challenging to support customer value recognition decision-making, including precision marketing and churn prediction.

Our model specialized in engineering enriched features through advanced RFM score segmentations, customer purchase affinity, and upscale dimensionality by the promos dataset aggregation. Then, leveraging the K-fold target encoding to improve accuracy and speed up the model optimization on high-dimensional and large datasets.

1.2 Main points about the dataset:

The original baseline model was limited by the inability to identify effective predictors and optimize model architecture on the large-scale dataset. Through the exploratory data analysis phase, we observed that the three pre-feature engineered predictors, RFM – Recency, Frequency, and Monetary handling through the algorithm, seriously constrained the model prediction quality. This can be supported by the proportion of targeted mainstream classified value, which is over 90 percent in all variables. The numeric values on the RMF predictor variables: monetary mean 16475.66 (label=1) and 3690.39 (label=0). Frequency mean 0.21289(label=1) and 0.219078(label=0), respectively. Lack of effective variation on the predicting variables linked with no resonance on classification values.

Instead, compared with our model, the advantage of our modeling approaches is two-fold, firstly, we piped the unutilized dataset of promotion to manifest powerful and worthwhile features for fast upscaling on the model dimensionality. The Promos and Transaction dataset captures comprehensive characteristics on marketing activity information related to market regions, product categories, customer-promotion interactions, merchants, and brands.

Second, we introduce a modified data structure to rapidly access the most relevant variables through target features encoding and thoughtful feature design on variables formulation from the original RFM features for enhancing predicting usefulness on customer activism decision making. An exhaustive comparison of activism inference through all attributes is prohibited on a very large dataset. The interest in implementing our modified dataset structure is raising model accuracy with the most useful ranking features and comparable algorithms running efficiency.

Exhibit 1.1: **The Imbalanced Class Distribution in Customer Dataset**

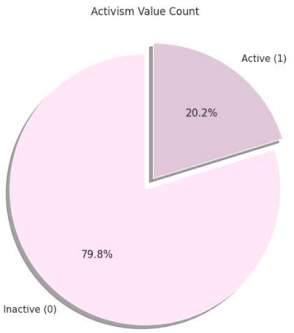
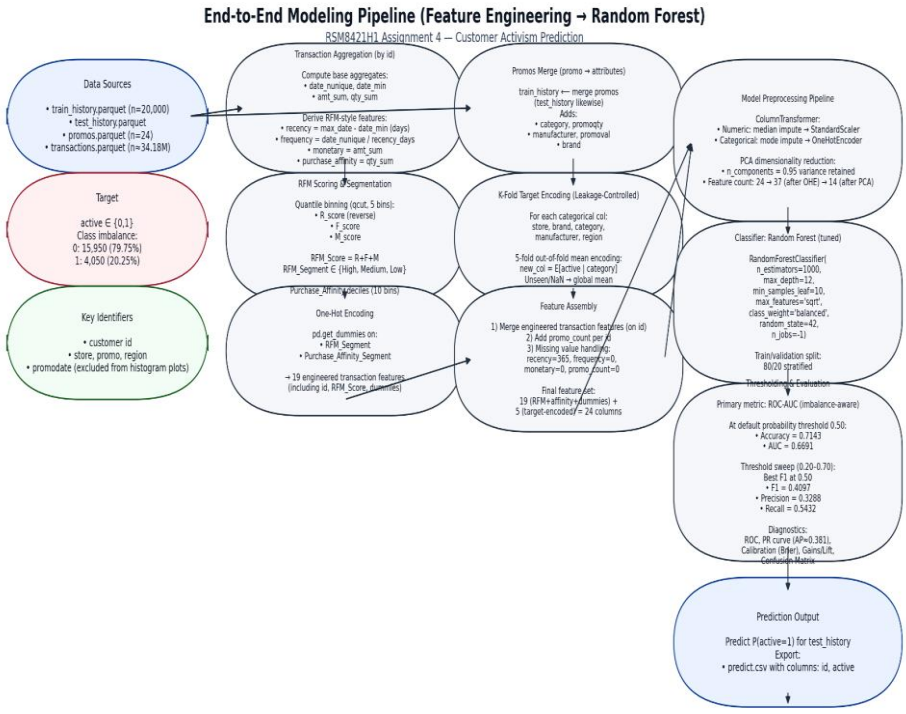


Exhibit 1.2: **The Visualization of Model Architecture**



2. Exploratory Data Analysis:

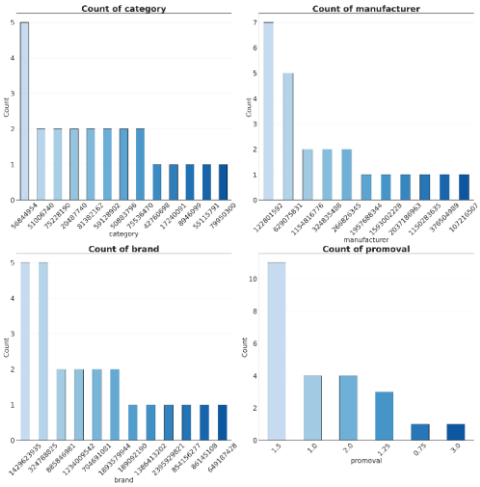
Promotion Dataset

The promotion dataset consists of 24 observations with six categorical variables, including promotional category, manufacturer, production category, promotion type, promotion dollar values and brand. By analyzing these promotional attributes, it becomes possible to identify segments of promotional activities that demonstrate stable and consistently strong activation performance across customers.

To operationalize the insight into feature enrichment, the promotion dataset was transformed using one-hot encoding and integrated into the customer-level training data using left joins. This approach allows each customer–promotion interaction to be represented in a higher-dimensional feature space, where we can efficiently analyze the segments of promotional activities that exhibit stable and excellent performance. This leads to further accurately maximizing activation rates for different complex feature combinations, conducting targeted promotional activities, and optimizing methods for ROI and other metrics. This further inspired us to aggregate the promos dataset using one-hot encoding and left joins, enriching the dataset's vector feature dimensions by analyzing each transaction in the customer's March transaction records training set.

Overall, incorporating promotional attributes into the modeling pipeline enhances more precise optimization of activation rates and provides a foundation for improving return on investment through the comination of promotion and the cutomers purchase tranctions with the highest expected effectiveness.

Exhibit 2.1: **Empirical Distribution of Key Promotional Feature Variables**

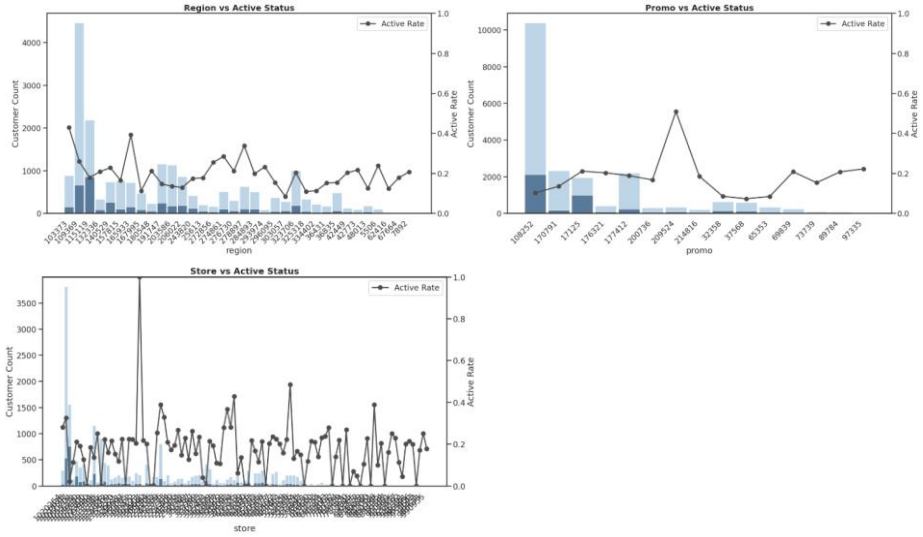


Dataset from the basic model

The training history dataset contains four basic features that record promotion delivery events. Exploratory analysis reveals two important structural patterns. First, the overall volume of promotions sent decreases over time, suggesting potential temporal effects or changes in campaign strategy that may influence customer responsiveness. Second, a substantial proportion of promotions with specific types are disproportionately concentrated within particular stores and regions.

We can rigorously point out the uneven distribution of feature types in the original model base dataset, the excessively large proportion of certain feature variable types, and the extreme predictive results of some feature variables that could serve as biased data for model training. This inspired us to conduct more in-depth feature engineering on the original RFM feature variables to learn more useful variables and further perform diversified feature engineering techniques such as k-fold target encoding and PCA.

Exhibit 2.2: **Distribution of Customer Counts and Activation Rates by Region, Promotion, and Store**



2.1 Feature Engineering:

Core Technique: Build-up Data Structure with Feature Enrichment and Target Feature Encoding

Feature Creating Level:

- **RFM Segmentation**
- **Purchase Price Affinity Segmentation**

Feature Encoding Level:

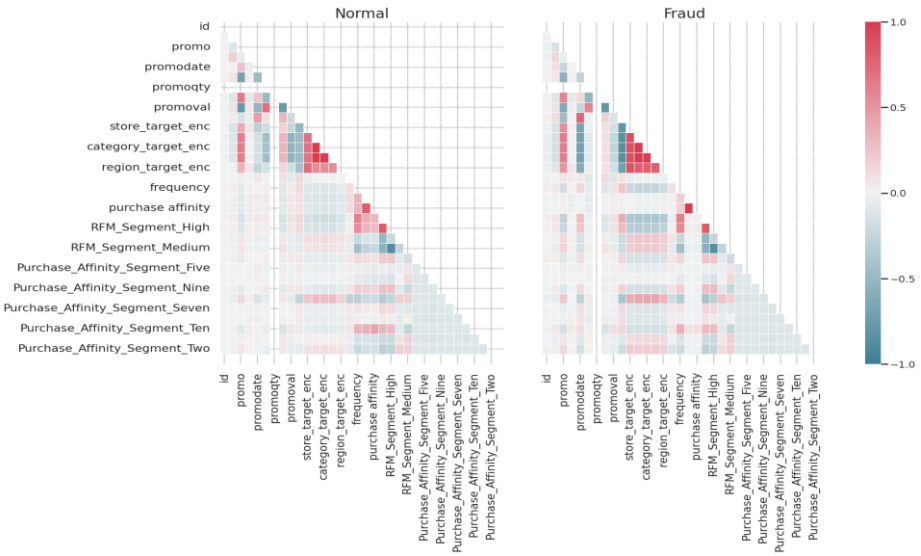
- **One-hot Encoding**
- **Target Encoding**

Targeted encoding is the central approach for us to improve the accuracy of the model. It enriches features through different high-dimensional spatial vector features and different meaningful perspectives, especially when facing high-

cardinality features, imbalanced datasets and application the random forest model. It is one of the most powerful methods in feature engineering. Our model primarily uses K-fold target encoding to reduce overfitting and improve prediction accuracy.

2.1.1 Enrich Data Features with RFM Segmentation and Purchase Price Affinity Segmentation

Exhibit 2.3: **Feature Correlation Structures Across Active and Inactive Customer Classes**



RFM Segmentation

The first step involves exploiting the transaction dataset to create a more advanced RFM-scored customer cluster. The enhanced RFM scores are then used to generate three quantiles using an equidistant method. High-value RFM customers are considered loyal customers (high_F, high_M), who are identified as loyal customers; medium-value customers are those with churn and moderate potential (high_M, low_R, low_F), indicating churn risk with upside potential; and the low-value RFM cluster includes low-potential customers (low_R, low_F, low_M) and dormant customers, indicating low engagement. The following methods enhance data processing capabilities and enrich data features. By effectively partitioning the RFM-based customer clusters to generate superior features.

Purchase Price Affinity Segmentation

The second step includes introducing a new feature engineered by Purchase Affinity. From the original over 30 million transaction historical records, we extracted customer purchasing quantities at the customer–store level and aggregated customer purchasing records from March 2016. Purchase Affinity was defined as the total transaction amount (amt_sum) divided by the total quantity purchased (qty_sum) for each customer at a given store, then clustered into ten different purchase affinity segments.

Exhibit 2.4: **Scenario-Based Intuition Customer Purchase Scenarios on Purchase Affinity Feature**

Incorporating enhances the model's ability to analyze pricing effectiveness, product attractiveness, and customer willingness to spend. Effective use of this variable can enhance the analysis of average order value, generating valuable inferences about marketing opportunities, including customer purchasing habits and consumption perceptions, such as the difference between purchasing baking molds and luxury handbags. It supports insightful feature analysis by identifying customers with higher upsell potential and by informing decisions on promotional intensity and product positioning. Finally, we used one-hot encoding for the after-feature engineering RFM baseline model.

2.1.2 Target Feature Encoding with Promotion Dataset

The model connects all variables in the Promos dataset to the training and test sets through target feature encoding and manual iteration for selecting significant attributes. Finally, the optimal data conditions were found, and the model was retrained with feature importances adjusted.

- **Global: Code Line**

```
train_encoded[new_col] = train_encoded[new_col].fillna(global_mean)
```

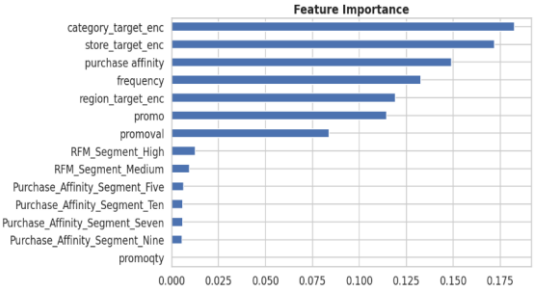
Insights from Feature Importance:

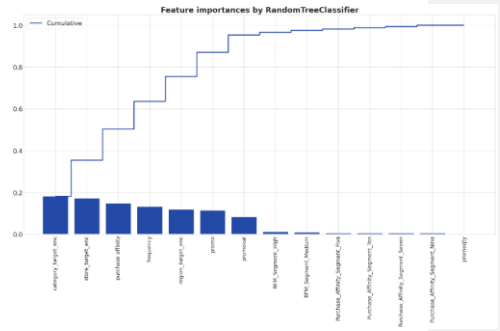
By manually exploring and merging variables from the entire engineered dataset for baseline models, then using t**arget feature** encoding to process the promotion-related variables. After constructing and collecting predictive variables with potential application value, the model finally selects effective variables to include: the store's own business performance (basic activity level), promotional activities (whether there are promotions affects purchase intention), and store location (foot traffic affects activity level). A new evaluation index is generated after weighting for prediction. Its core principle is that the original model only focuses on individual users, while the new method integrates store and promotional factors to improve discriminative power.

Code Snippet shoot:

```
Merging all features...
Final feature list: ['id', 'recency', 'frequency', 'monetary', 'purchase affinity', 'RFM_Score', 'RFM_Segment_High', 'RFM_Segment_Low', 'RFM_Segment_Medium', 'Purchase_Affinity_Segment_Eight', 'Purchase_Affinity_Segment_Five', 'Pu
```

Exhibit 2.5: **Contribution of Engineered Features to Model Predictions (Individual Importance vs. Cumulative Contribution)**





Challenges from Feature Selection:

The Variable `promos_count` was dropped but finally added. We removed extremely inactive customers with no transaction records and accurately imputed missing transaction data. Removing it would reduce the model's accuracy. Therefore, after screening the most important potential target variables, we reintroduced the `promos_count` variable after all retained variables, which effectively improved the model's accuracy.

3. Random Forest Model Set Up:

The third step involved conducting hyperparameter tuning on the baseline Random Forest model, which produced an accuracy of approximately 0.57 even under optimized parameter settings. Consequently, we shifted our focus from model optimization to a deeper investigation of the dataset itself, aiming to identify structural deficiencies, ineffective features, and potential sources of information loss that prevented the model from effectively distinguishing between positive and negative classes.

Code Snippet shoot:

Original baseline model: Test AUC: 0.52:

```
print('Test Accuracy: {:.2f}'.format(model.score(X_test, y_test)))
print('Test AUC: {:.2f}'.format(roc_auc_score(y_test, model.predict_proba(X_test)[:, 1])))
```

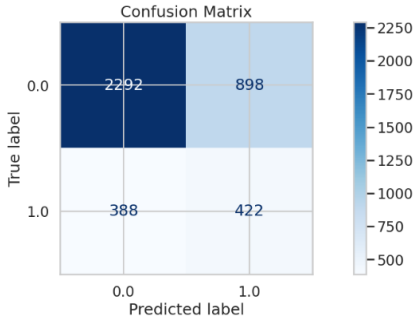
```
Test Accuracy: 0.78
Test AUC: 0.52
```

Comparison with Optimized Random Forest Model: Test AUC: 0.69:

```
print('Test Accuracy: {:.2f}'.format(model.score(X_test, y_test)))
print('Test AUC: {:.2f}'.format(roc_auc_score(y_test, model.predict_proba(X_test)[:, 1])))
```

```
Test Accuracy: 0.68
Test AUC: 0.69
```

Exhibit 3: **Confusion Matrix and Precision–Recall–F1 Performance Summary**



	precision	recall	f1-score	support
0	0.86	0.72	0.78	3190
1	0.32	0.52	0.40	810
accuracy	0.68	0.68	0.68	0.6785
macro avg	0.59	0.62	0.59	4000
weighted avg	0.75	0.68	0.70	4000

4. Experimental Setup and Evaluation

4.1 Data Splitting

Used an 80/20 train-test split.
Used stratified sampling for the target variable, maintaining a consistent ratio of positive to negative samples.
The test set was completely independent and used only for final evaluation.

批注 **[KW1]:** Jay Cao-Suggestion on using 'time_series_split' for preventing future information leakaging into training data.

4.2 Controlled Experiment Design

The experiment followed strict variable control principles:
The model structure remained unchanged.
Only the performance of different feature sets was compared.
Baseline: RFM features
Enhanced: RFM + Target Encoding + Promotional Attributes

- **Random Forest with optimized hyperparameter:**

 # Use GridSearchCV to automatically search for optimal random forest parameters

```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
# Define the parameter grid
param_grid = {
'n_estimators': [300, 500, 800],
'max_depth': [6, 10, None],
'min_samples_split': [5, 10, 20],
'min_samples_leaf': [2, 5, 10],
'max_features': ['sqrt'],
'class_weight': ['balanced']
}
# Create the base model
rf = RandomForestClassifier(random_state=42, n_jobs=-1)
# Perform grid search using 5-fold cross-validation
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
grid = GridSearchCV(
```

rf,
param_grid,
scoring='roc_auc', **# Use AUC as the scoring metric**
cv=cv,
n_jobs=-1, **# Use all CPU cores**
verbose=1 **# Show progress**

- **Out-of_fold Encoding avoid data leakage:**

```
for train_index, val_index in kf.split(train_df):
    X_tr, X_val = train_df.iloc[train_index], train_df.iloc[val_index]
    means = X_tr.groupby(col)[target_col].mean()
    train_encoded.loc[val_index, new_col] = X_val[col].map(means)
```

- **SMOTE rebalanced dataset:**

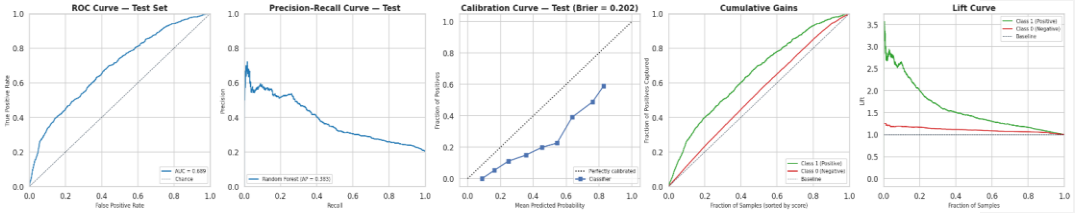
```
from imblearn.over_sampling import SMOTE
from sklearn.impute import SimpleImputer
```

```
imputer = SimpleImputer(strategy='median')
X_train_imputed = imputer.fit_transform(X_train)
 # Initialize SMOTE
smote = SMOTE(random_state=42)

# Apply SMOTE to the training data only
X_train_res, y_train_res = smote.fit_resample(X_train_imputed, y_train)

# Check the new class distribution
print(f'Original training class distribution:\n{y_train.value_counts()}')
print(f'Resampled training class distribution:\n{y_train_res.value_counts()}')
```

4.3 Evaluation Metrics

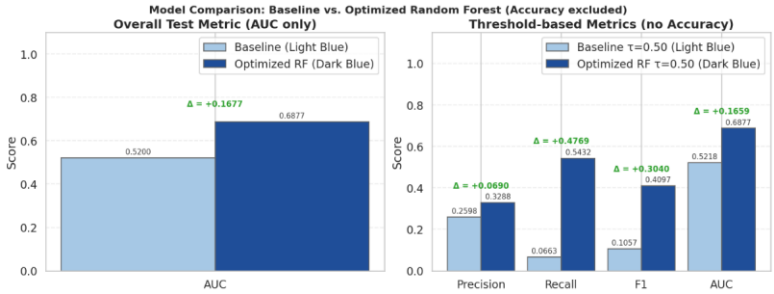
AUC as the primary metric.
This task is essentially a ranking problem (finding the most worthwhile users to promote), and AUC directly reflects the model's discriminative ability.
Accuracy is used as a secondary metric, supported by comprehensive evaluation diagnostics including the confusion matrix, precision, recall, F1-score, ROC analysis, calibration, cumulative gains, and lift.

Exhibit 4.1 **Model Evaluation Visualization**
(ROC Curve, Calibration Analysis, Cumulative Gains, and Lift Performance)



4.4 Summary of Experimental Results

Exhibit 4.2 **Comparison Table for Models Performance on Test Set**
(Baseline Random Forest Model vs. Experimented Random Forest Model)



	Baseline Random Forest Model	Experimented Random Forest Model	Changes
Test Accuracy	0.78	0.68	-0.10
Test AUC	0.52	0.69	-0.17

With the original features, AUC ≈ 0.52–0.57, close to randomness.
After introducing Target Encoding and feature fusion, AUC improved to approximately 0.69.
The results show that the performance improvement mainly comes from feature engineering, not model complexity.

5. Future Directions:

The potential database we can explore: — This model only uses data from the promos dataset for data merging and joining. Therefore, the variables related to the transaction database, such as the market regions, product shapes and quantities, product brand, and store location information, can be further leveraged and explored through applying a one-hot target encoder to improve the model precision and the quality of prediction. The goal is to increase the AUC to over 75%, with a longer time and process possibility.

The potential variable we can explore: — Variables may be constructed through feature encoding approaches that capture customers' purchasing habits, particularly those inferred from the frequency and recency of specific product types. For example, developing the "product_last_purchase" variable reflects item-level recency and can enhance the model's understanding of product engagement patterns. Further variables could be engineered using market-group information to highlight geographical and customer segmentation effects, including indicators of "location-specific activism," which may reveal meaningful differences in purchasing behavior across regions. Combining the lag of time created columns to reflect the time effects on purchasing behaviors, another good example for feature engineering to create meaningful variables is: "The last shopping weekend."

The potential metrics and potential models we can apply – Potential algorithms include CatBoost, XGBoost, and LightGBM models better suited to imbalanced classification and high-dimensional feature spaces, as well as effective and strong performance gains when modeling complex categorical and numerical interactions. Further mathematical metrics that can be applied include F1, Precision, and Recall, which measure the classification targeted task's labeling accuracy through different evaluation aspects. The business level metrics that can be used in the customer purchase activism include Customer Sensitivity, Business ROI, Promotion Cost efficiency, which effectively measures the service effectiveness and the real commercial value of the specific activity or physical information (chain stores, customer behaviours, time of promotions, etc.)