ROSSMANN STORE SALES

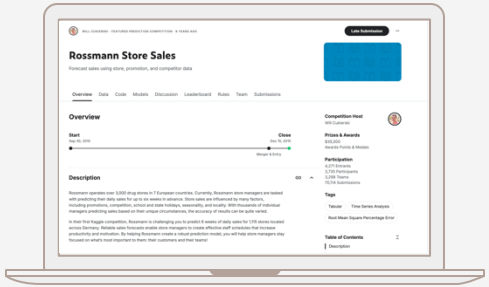
LẬP TRÌNH PYTHON CHO MÁY HỌC - CS116

PHẠM NGUYÊN TƯỜNG – 23521751
NGUYỄN THANH TÙNG – 23521745
CHƯỚNG HỒNG VĂN – 23521769
TĂNG HOÀNG PHÚC – 23521219

TABLE OF CONTENTS

01

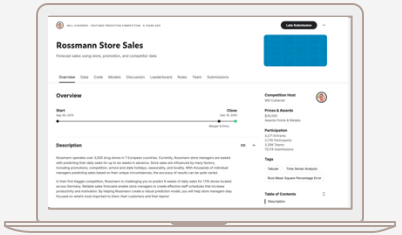
INTRODUCTION



INTRODUCTION

➢ GIỚI THIỆU ĐỀ TÀI

- Dự đoán doanh thu 1115 cửa hàng của chuỗi bán lẻ Rossmann ở Đức.
- Bộ dữ liệu gồm 1,017,209 dòng với 2 file train.csv và store.csv
- **train.csv** 9 cột (Store, DayOfWeek, Date, Sales, Customers, Open, Promo, StateHoliday, SchoolHoliday)
- **store.csv** 10 cột (Store, StoreType, Assortment, CompetitionDistance, CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo2, Promo2SinceWeek, Promo2SinceYear, PromoInterval)



INTRODUCTION

➢ LÝ DO CHỌN ĐỀ TÀI

- Dữ liệu thực tế, quy mô lớn
- Thích hợp cho việc áp dụng Machine Learning Pipeline
- Rèn luyện khả năng xử lý các yếu tố chu kỳ, thời vụ, xu hướng trong dữ liệu Time Series

➢ INPUT

- Dataset đã được gán nhãn gồm khoảng 1 triệu dòng về lịch sử bán hàng để huấn luyện mô hình
- Test set chứa thông tin trong tương lai gần, dùng để dự đoán kết quả đầu ra

➢ OUTPUT

- Doanh thu trong 48 ngày tiếp theo cho 1115 cửa hàng

02
EDA

EDA

```
[ ]  train.isnull().sum()
```

CompetitionDistance	2642
CompetitionOpenSinceMonth	323348
CompetitionOpenSinceYear	323348
Promo2SinceWeek	508031
Promo2SinceYear	508031
PromoInterval	508031

```
[ ]  test.isnull().sum()
```

CompetitionDistance	96
CompetitionOpenSinceMonth	15216
CompetitionOpenSinceYear	15216
Promo2SinceWeek	17232
Promo2SinceYear	17232
PromoInterval	17232

MISSING VALUE ANALYSIS

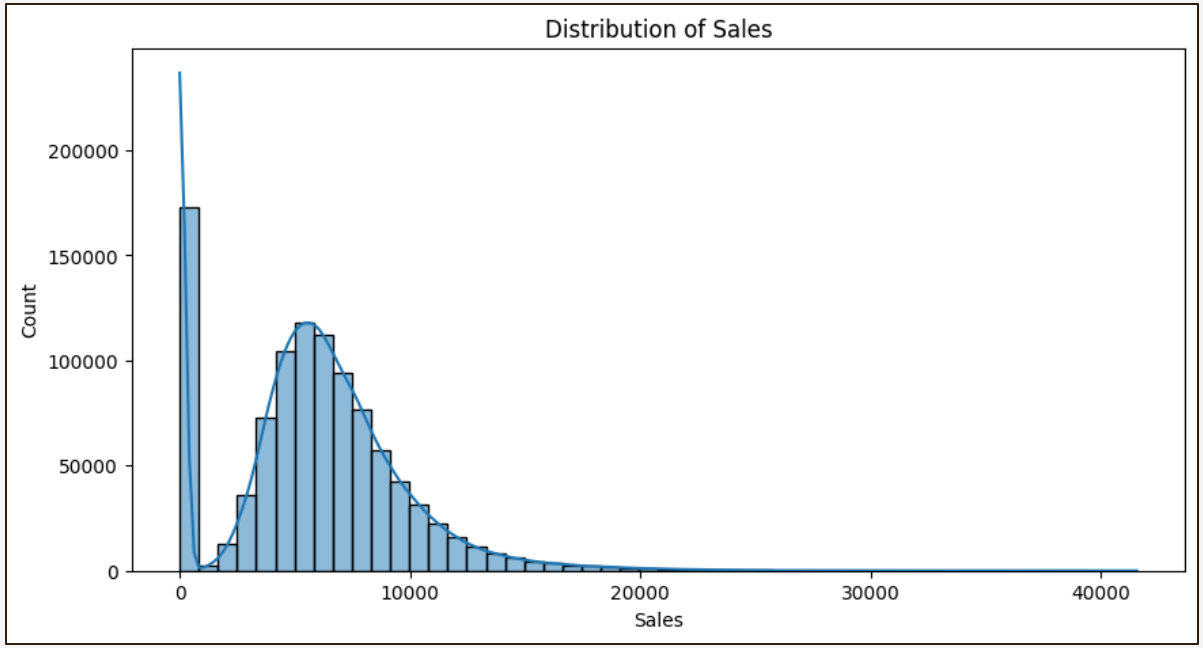
EDA

DUPLICATE DATA ANALYSIS

```
[ ]  print(train.duplicated().sum())
     print(test.duplicated().sum())
```

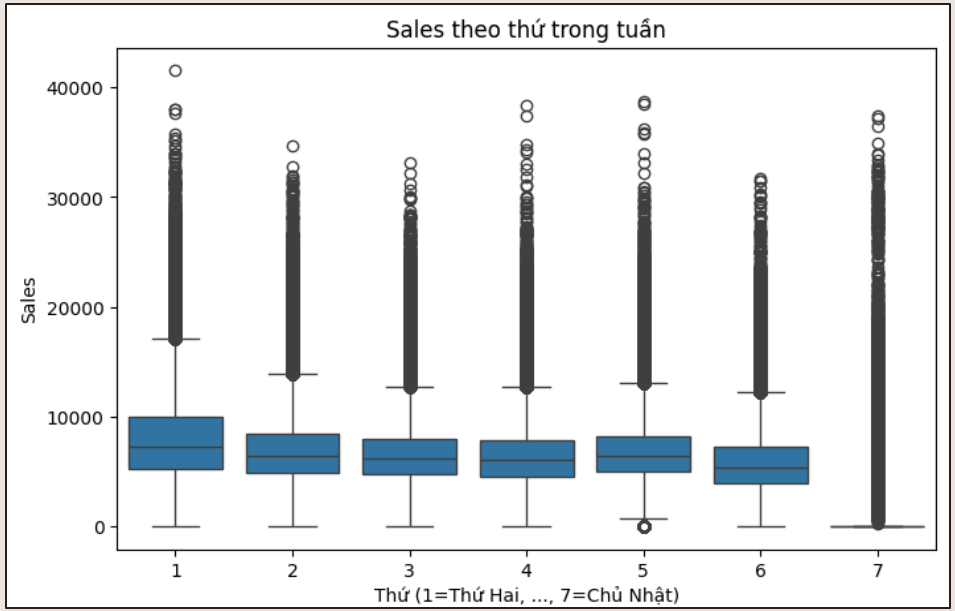
```
0
0
```

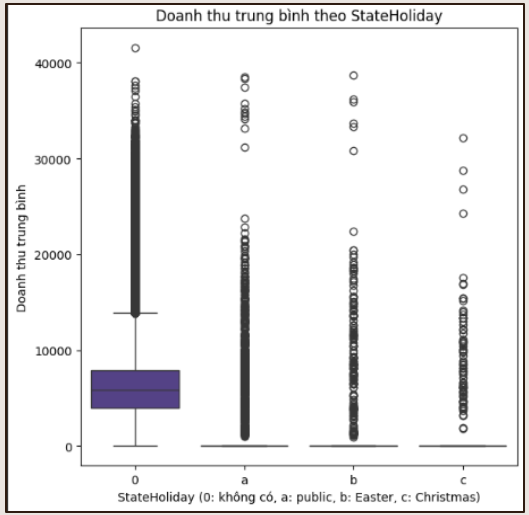
EDA

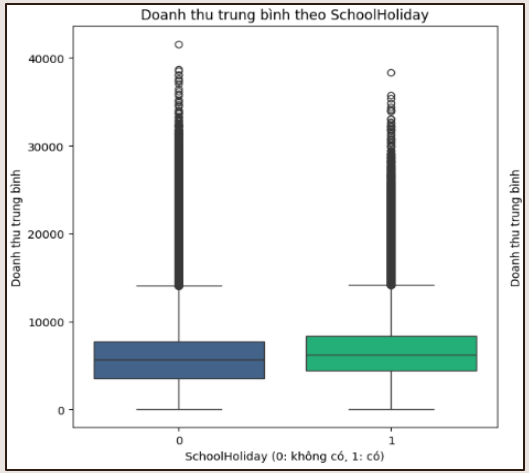


Phân phối doanh thu (Sales) có xu hướng lệch phải, tập trung chủ yếu ở mức thấp và xuất hiện nhiều giá trị tăng đột biến

EDA

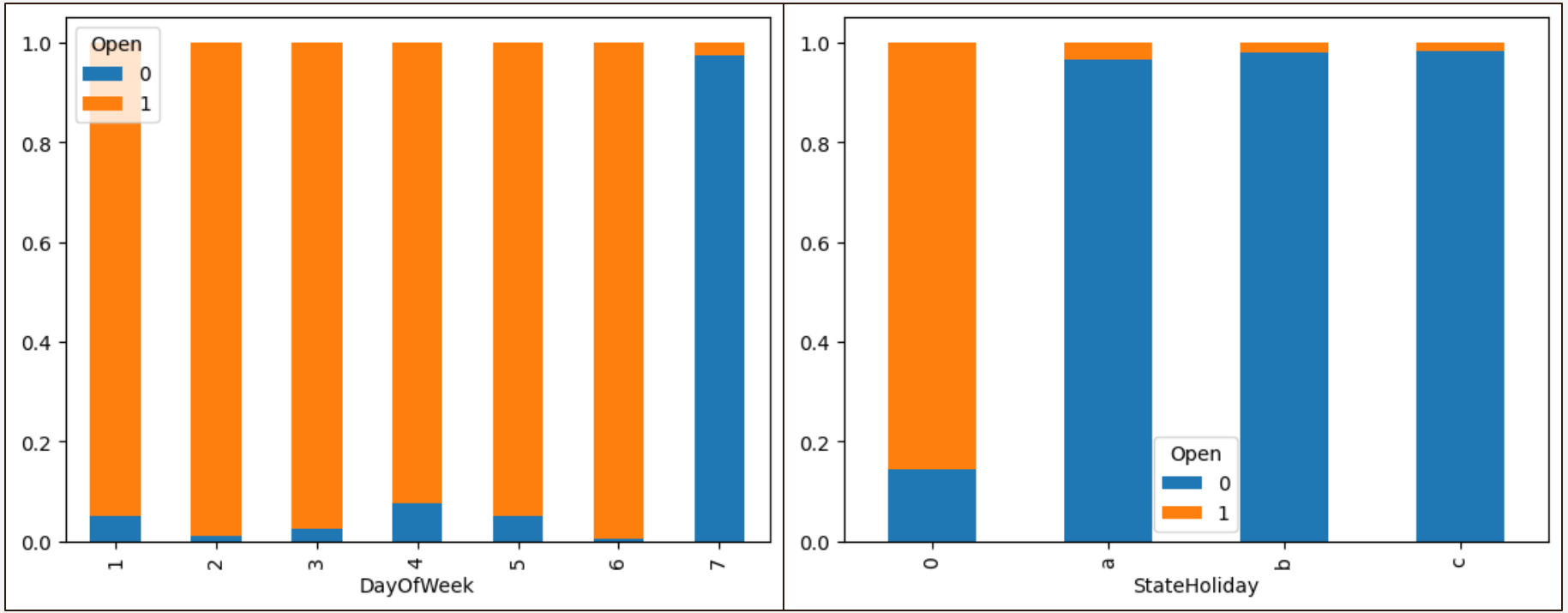






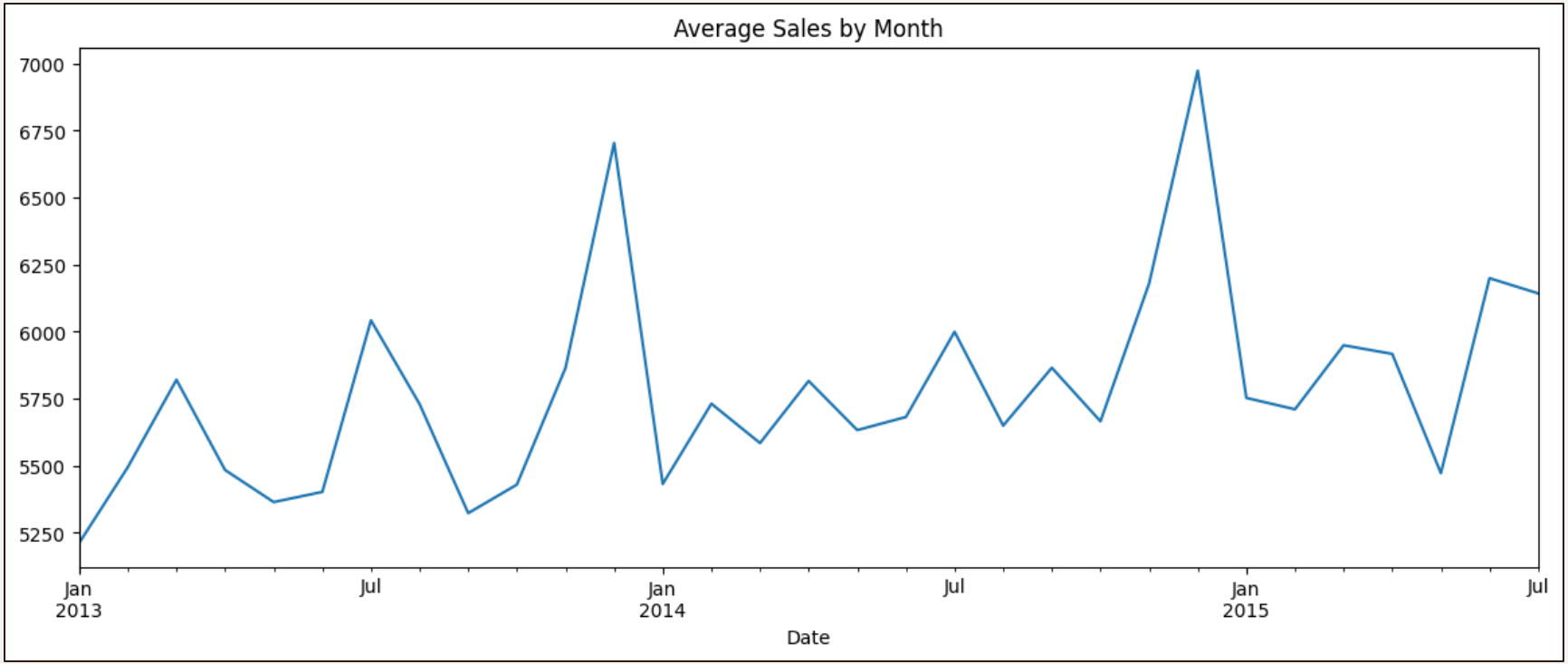
- Doanh số gần như bằng 0 vào Chủ nhật và ngày lễ
- Các ngày còn lại doanh số khá ổn định

EDA



→ Phần lớn cửa hàng đóng cửa vào ngày cuối tuần và ngày lễ

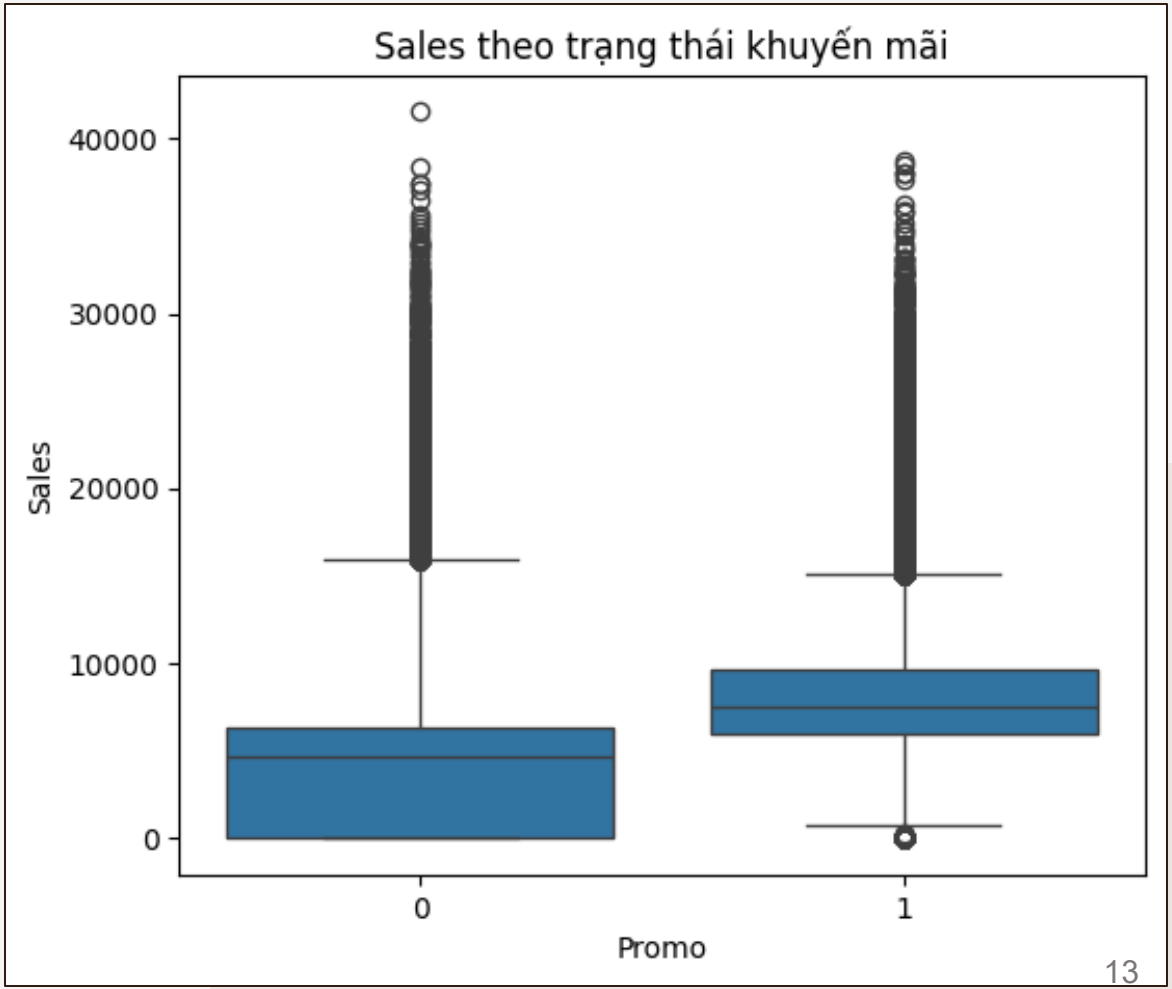
EDA



→ Doanh thu tăng mạnh vào tháng 3, kỳ nghỉ hè (tháng 6,7) và dịp cuối năm (tháng 12)

EDA

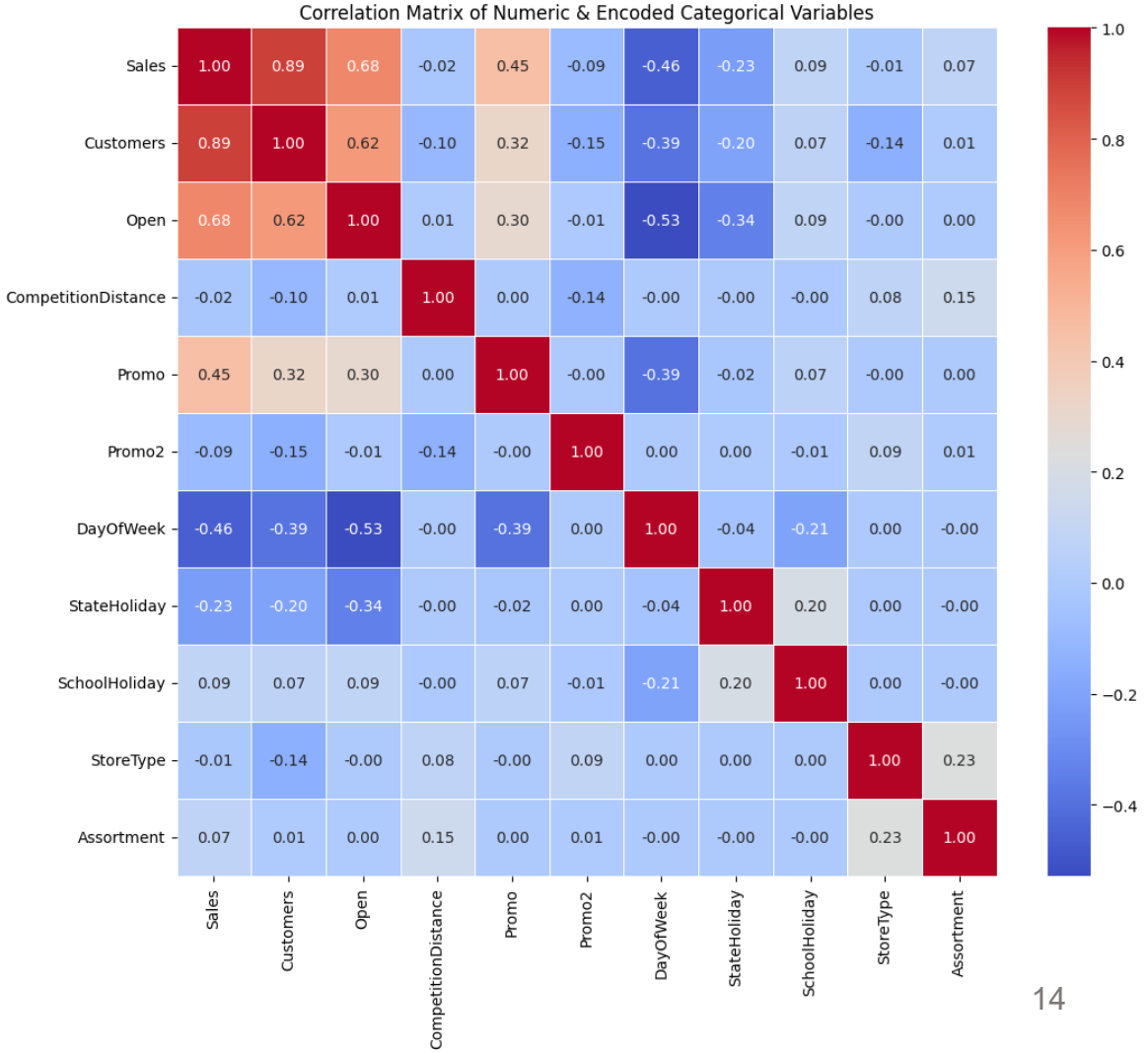
- Cửa hàng nào có Promo=1 thường có doanh thu cao hơn



CORRELATION MATRIX

- Mối tương quan giữa Sales và Customers là mạnh mẽ nhất
  → Open-Sales và Customers-Sales

- Promo cũng ảnh hưởng đến Sales và Customers



03

PREPROCESSING

PREPROCESSING – HANDLING MISSING VALUES

```
[ ]  train.isnull().sum()
```

CompetitionDistance	2642
CompetitionOpenSinceMonth	323348
CompetitionOpenSinceYear	323348
Promo2SinceWeek	508031
Promo2SinceYear	508031
PromoInterval	508031

- CompetitionDistance → Điền 0
- CompetitionSinceMonth/Year → Điền 0
- Các cột Promo2SinceYear, Promo2SinceWeek, PromoInterval chỉ thiếu khi Promo2=0
  → Điền Promo2SinceYear/Week=0, PromoInterval="None"

PREPROCESSING – HANDLING OUTLIERS

```
Số dòng có Open = 1 và Sales = 0: 54
```

→ Drop

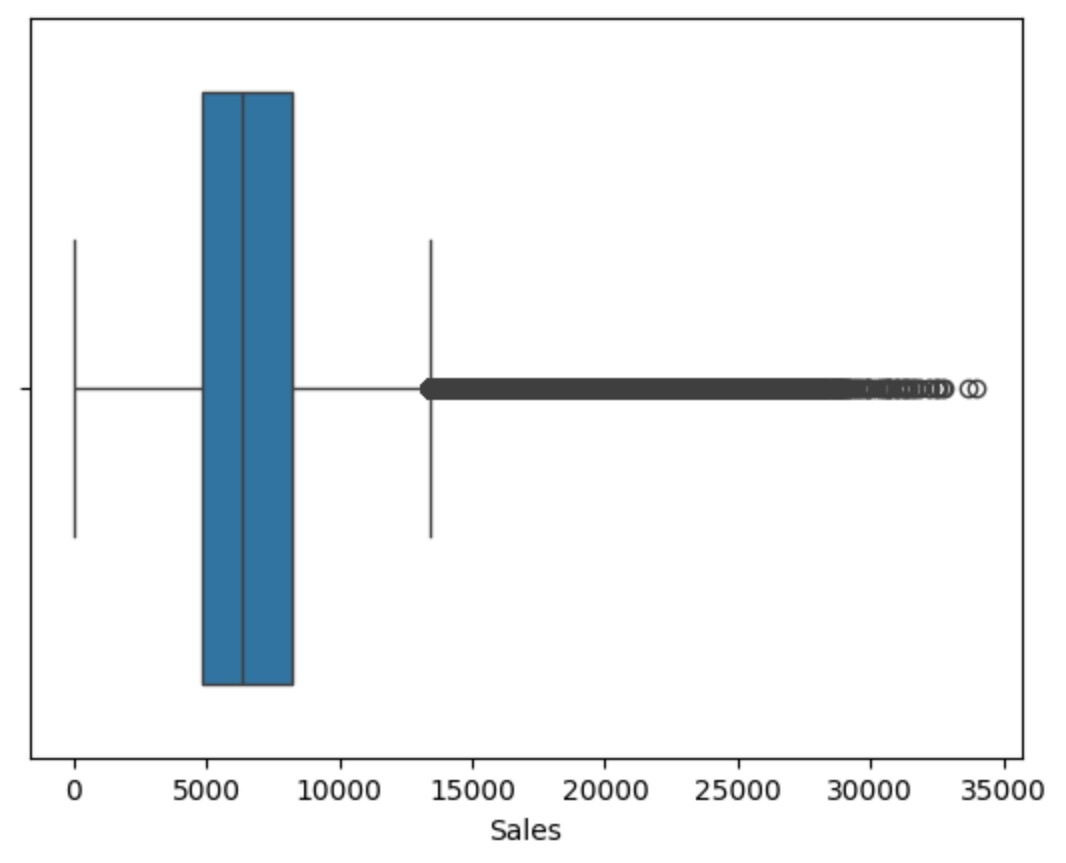
PREPROCESSING – MAD-BASED OUTLIER DETECTION

Median Absolute Deviation (MAD) = $\mathbf{median}\ (|x_i\ - \mathbf{median}(x)|)$

$$\mathbf{Modified\ Z-score} = \frac{0.6745\ \times\ |x_i\ - \mathbf{median}|}{\mathbf{MAD}}$$

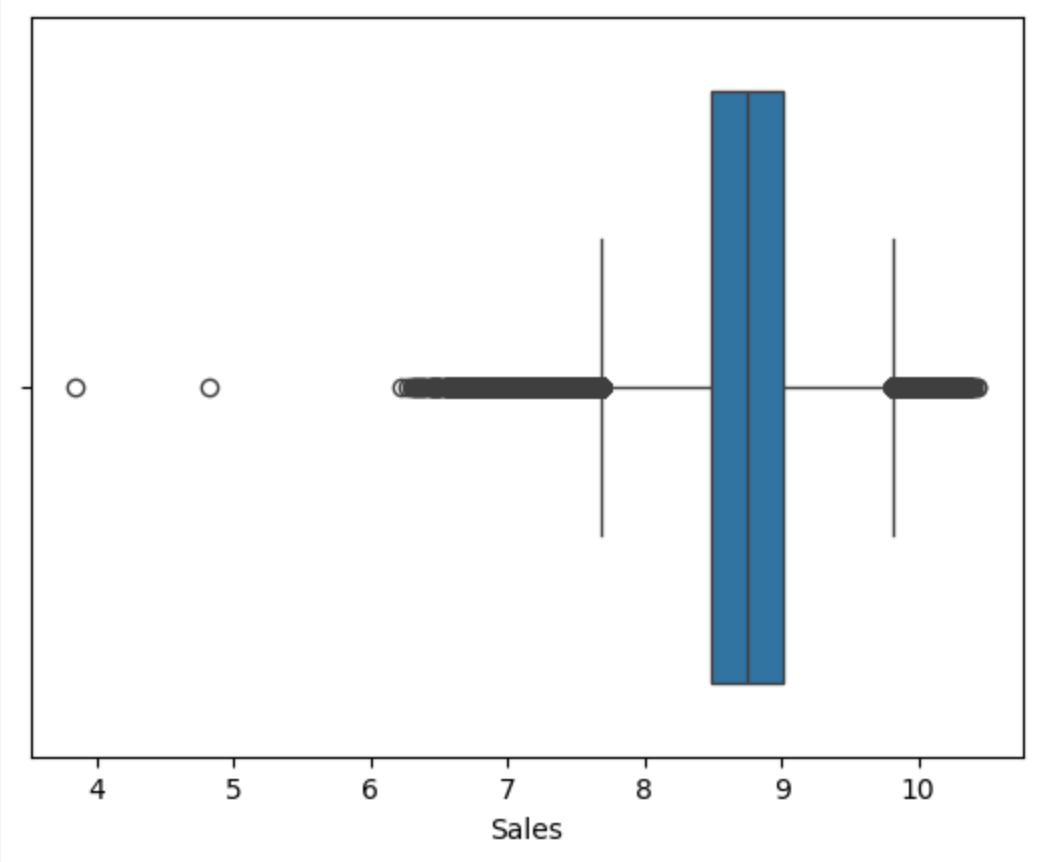
If Z-score > threshold ⇨ Outliers

PREPROCESSING – HANDLING OUTLIERS

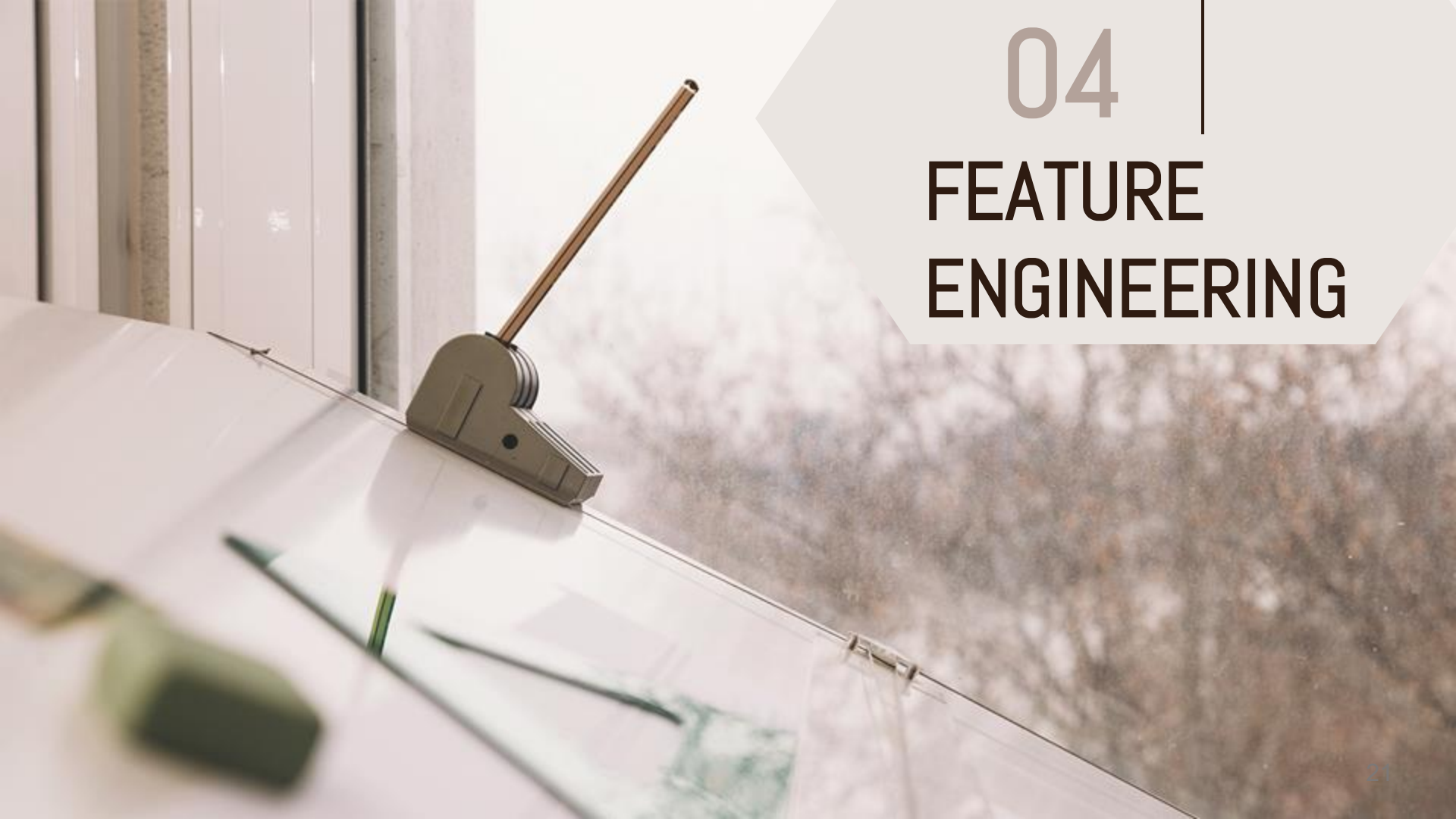


→ Dùng log-transform

PREPROCESSING – HANDLING OUTLIERS



→ Kết quả sau khi áp dụng log-transform



04

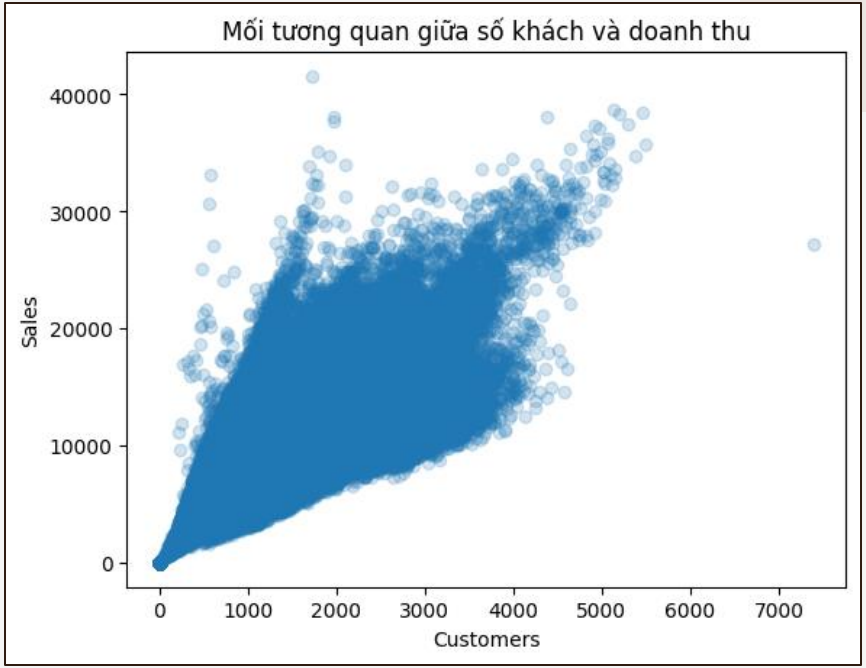
FEATURE ENGINEERING

FEATURE ENGINEERING

Customers → ảnh hưởng đến Sales

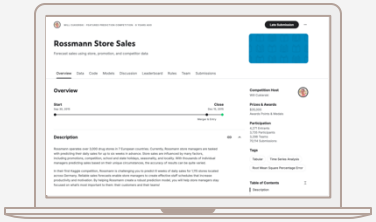
Tạo ra các cột mới
- Store_sales_per_day
- Store_customer_per_day
- Store_sales_per_customer_per_day

```
train = train.drop('Customers', axis=1)
```

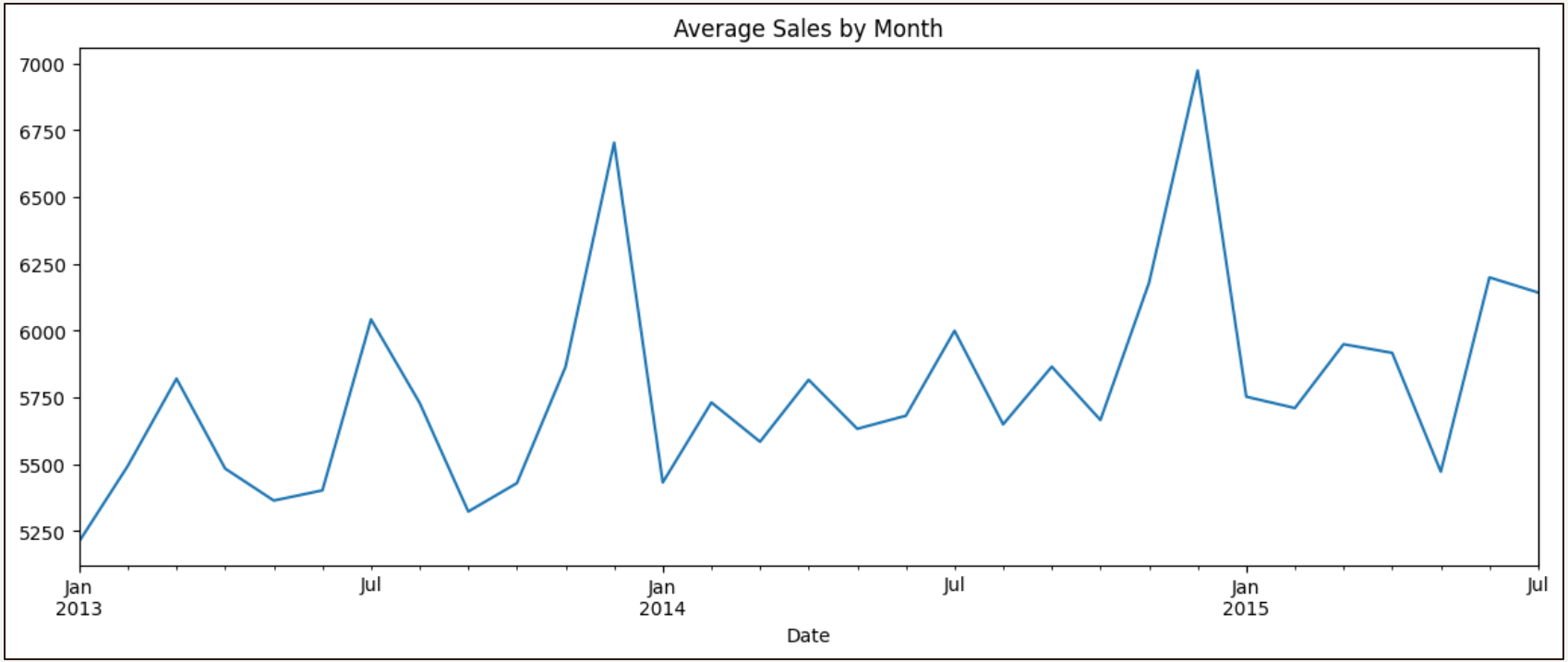


FEATURE ENGINEERING

- Promo2:
  - Gộp 2 cột **Promo2SinceYear** và **Promo2SinceWeek** → **Promo2StartDate**
  - Tính toán số ngày kể từ khi Promo2 bắt đầu → **TimeSincePromo2Start**
- Tương tự, ở **Competition** xử lý tương tự → **CompetitionAge**
- Vì các đặc trưng thời gian (**Month,DayofWeek, Week**) có tính chất tuần hoàn → Dùng 2 hàm số **sin** và **cos** để encoding

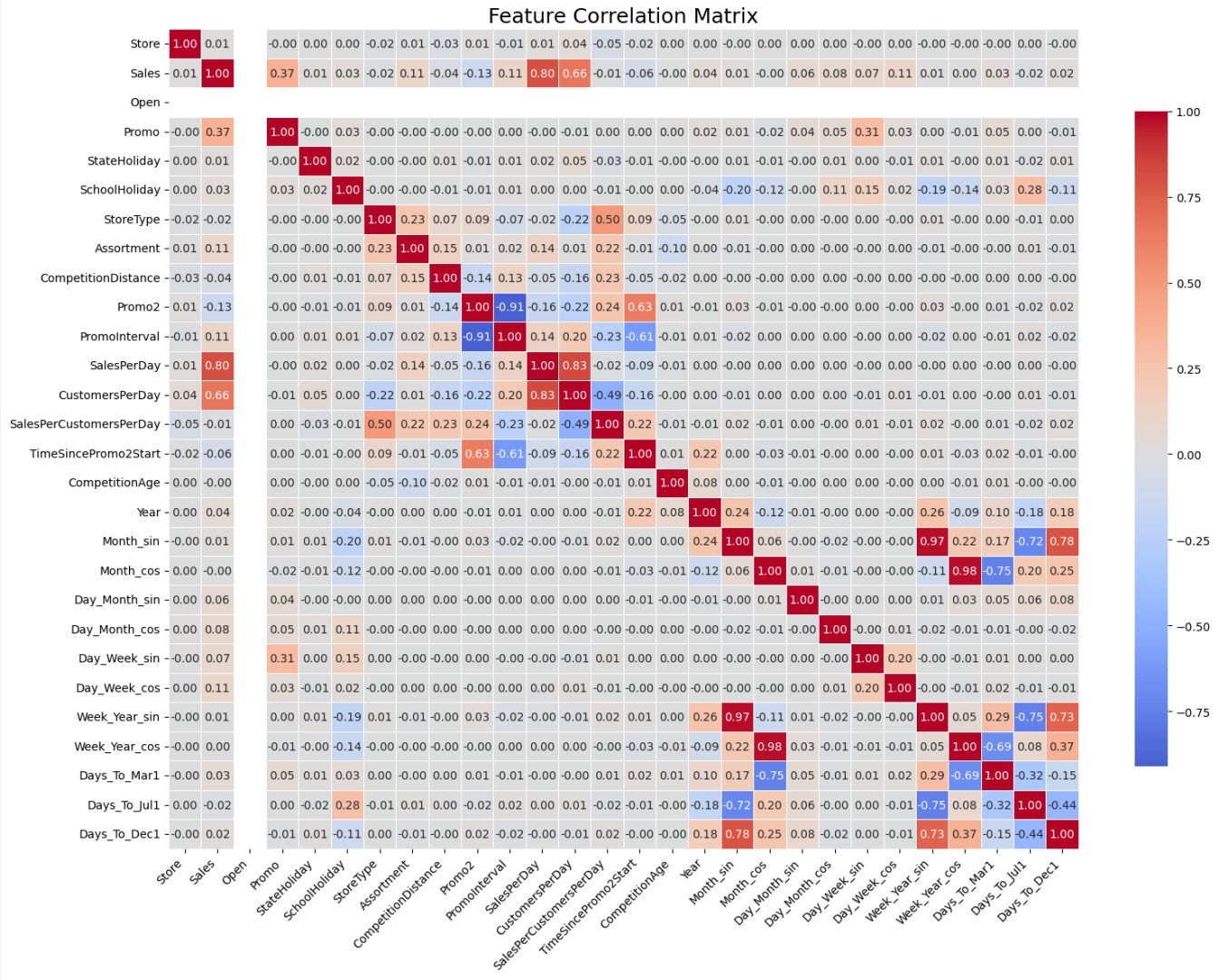


FEATURE ENGINEERING



Doanh thu tăng mạnh ở khoảng thời gian tháng 3, kỳ nghỉ hè (tháng 6,7), trước Giáng sinh và năm mới (Tháng 11,12)

CORRELATION MATRIX



CORRELATION MATRIX

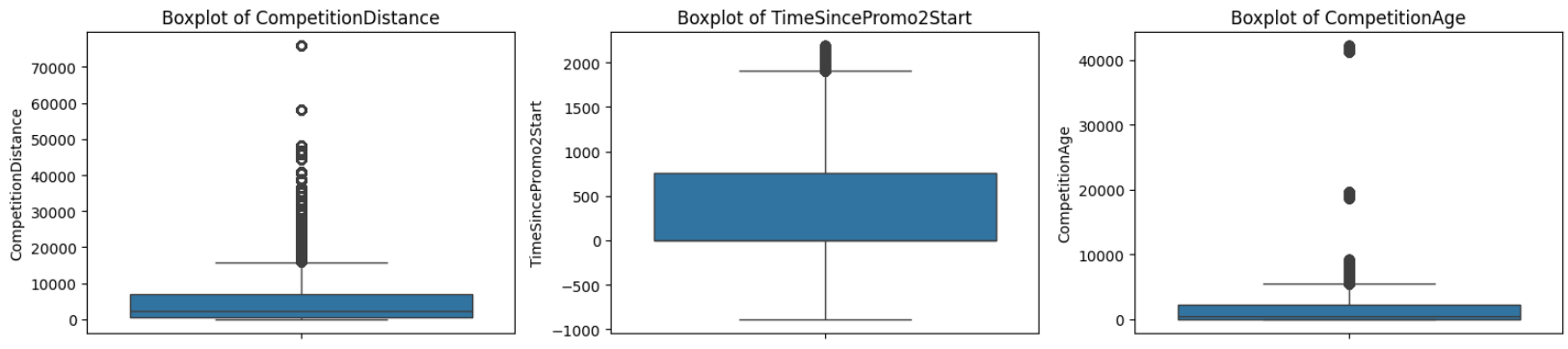
FEATURE ENCODING

Dùng **Label Encoding** để chuyển các cột phân loại thành dạng số

```
[ ]  for col in cat_cols:
         print(col, train[col].unique(), test[col].unique())
```

```
StateHoliday [0 1 2 3] [0 1]
Assortment [0 2 1] [0 2 1]
PromoInterval [3 1 0 2] [3 1 0 2]
```

FEATURE TRANSFORM



RobustScaler

DATASET SUMMARY

```
train.shape
```

```
(1017209, 18)
```

TRƯỚC KHI XỬ LÝ

Rows: 1,017,209
Columns: 18

SAU KHI XỬ LÝ

Rows: 831,359
Columns: 26

```
train.shape
```

```
(831359, 26)
```

05

MODEL TRAINING

MODEL SELECTION

- LINEAR REGRESSION
- RIDGE REGRESSION
- LASSO REGRESSION
- CATBOOST
- XGBOOST
- LIGHTGBM

METRICS

- MAE (Mean Absolute Error)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

- RMSPE (Root Mean Squared Percentage Error)

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

TRAINING AND TUNING MODEL

```
[ ]  X_train,X_val,y_train,y_val=train_test_split(X,y,test_size=0.25,shuffle=False)
```

Tối ưu hóa tham số mô hình

- GridSearchCV: Ridge Regression, Lasso Regression

```
ridge_grid = {
    "alpha": [0.001 ,0.01, 0.1, 1, 10]
}
```

```
lasso_grid = {
    "alpha": [0.0001, 0.001, 0.01, 0.1, 1, 10],
    'max_iter': [1000, 5000, 10000, 10000]
}
```

TRAINING AND TUNING MODEL

Tối ưu hóa tham số mô hình

- **Optuna**: XGBoost, CatBoost, LightGBM

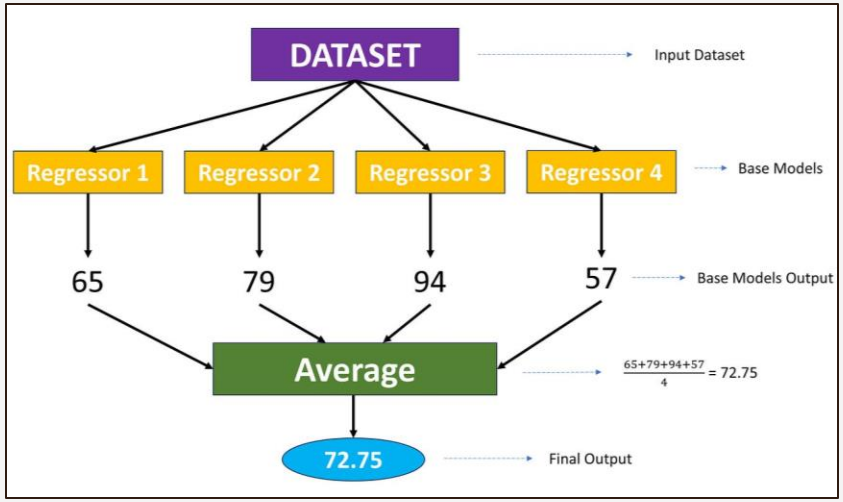

```python
# ==== Tuning XGBoost (GPU) với Optuna ====
def objective_xgb(trial):
    params = {
        'n_estimators': trial.suggest_int('n_estimators', 100, 1000),
        'max_depth': trial.suggest_int('max_depth', 3, 10),
        'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.3, log=True),
        'subsample': trial.suggest_float('subsample', 0.5, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.5, 1.0),
        'gamma': trial.suggest_float('gamma', 0, 5),
        'reg_alpha': trial.suggest_float('reg_alpha', 0, 10),
        'reg_lambda': trial.suggest_float('reg_lambda', 0, 10),
        "device": "cuda",
        "random_state": 42
    }
```

```python
# ==== Tuning LightGBM (GPU) với Optuna ====
def objective_lgb(trial):
    params = {
        'n_estimators': trial.suggest_int('n_estimators', 100, 1000),
        'max_depth': trial.suggest_int('max_depth', 3, 10),
        'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.3, log=True),
        'num_leaves': trial.suggest_int('num_leaves', 20, 200),
        'min_data_in_leaf': trial.suggest_int('min_data_in_leaf', 10, 100),
        'feature_fraction': trial.suggest_float('feature_fraction', 0.5, 1.0),
        'bagging_fraction': trial.suggest_float('bagging_fraction', 0.5, 1.0),
        'lambda_l1': trial.suggest_float('lambda_l1', 0.0, 5.0),
        'lambda_l2': trial.suggest_float('lambda_l2', 0.0, 5.0),
        "device": "gpu",
        'boosting_type':'gbdt',
        "random_state": 42,
        'boosting_type': 'gbdt',
    }
```

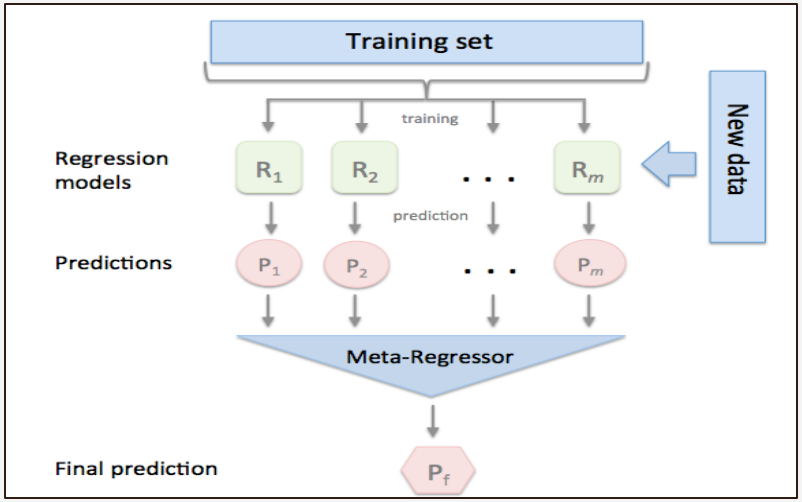
```python
def objective_catboost(trial):
    params = {
        'iterations': trial.suggest_int('iterations', 300, 1000),
        'depth': trial.suggest_int('depth', 4, 10),
        'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.3, log=True),
        'l2_leaf_reg': trial.suggest_float('l2_leaf_reg', 1, 10),
        'random_strength': trial.suggest_float('random_strength', 1e-9, 10.0, log=True),
        'bootstrap_type': trial.suggest_categorical('bootstrap_type', ['Bayesian', 'Bernoulli', 'MVS']),
        'eval_metric': 'RMSE',
        'task_type': 'GPU',
        'random_seed': 42,
        'verbose': False
    }
```

ENSEMBLE

- VOTING REGRESSOR



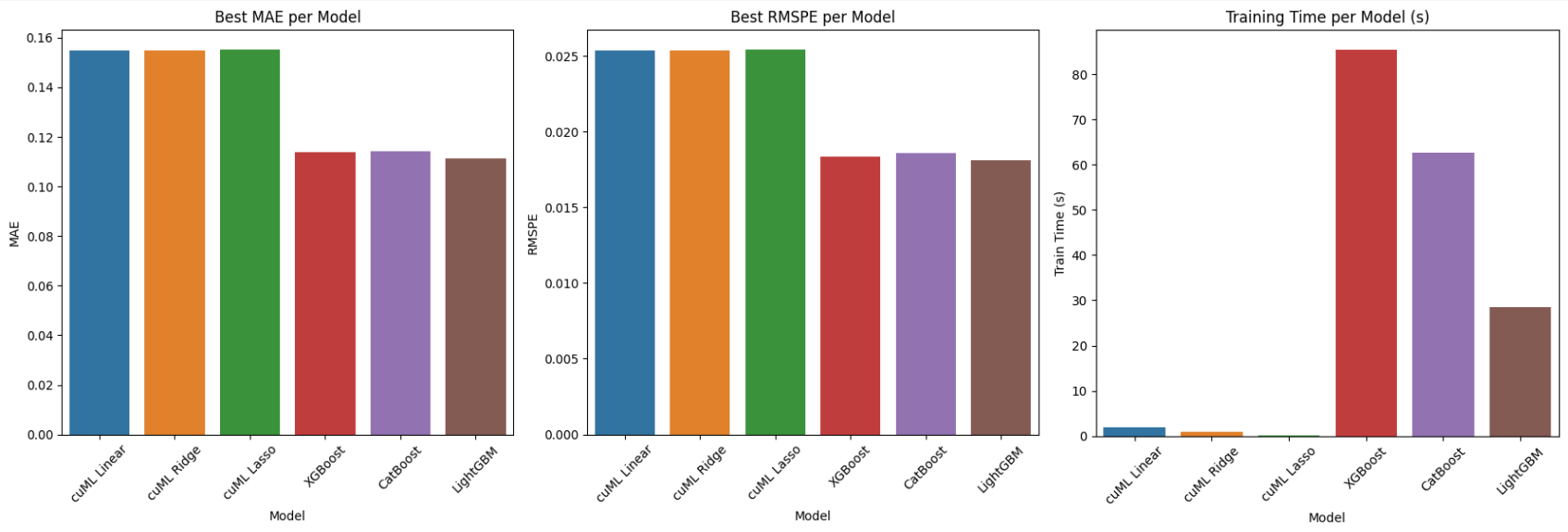
- STACKING REGRESSOR



06

MODEL EVALUATION

MODEL EVALUATION



PERFORMANCE COMPARISON OF REGRESSION MODELS

MODEL EVALUATION

MODEL	LINEAR REGRESSION	RIDGE REGRESSION	LASSO REGRESSION	XGBOOST	CATBOOST	LIGHTGBM
MAE	0.1549	0.1549	0.1553	0.1137	0.1143	**0.1114**
RMSPE	0.0254	0.0253	0.0253	0.0185	0.0183	**0.0180**
TRAINING TIME (s)	1.9923	0.9202	**0.1595**	85.501	62.571	58.584

BENCHMARKING OF MODELS

MODEL EVALUATION

MODEL	MAE	RMSPE	TRAINING TIME (s)
VOTING REGRESSOR	0.1100	0.0179	**63.11**
STACKING REGRESSOR	**0.1093**	**0.0177**	426.4

ENSEMBLE MODELS

MODEL EVALUATION

Submission and Description	Private Score	Public Score	Selected
catboost_submission (17).csv Complete (after deadline) · 2d ago	0.11486	0.10548	☐
xgb_submission (18).csv Complete (after deadline) · 1d ago	0.12188	0.11739	☐
lgbm_submission (18).csv Complete (after deadline) · 2d ago	0.11444	0.10711	☐
vote_submission (18).csv Complete (after deadline) · 2d ago	0.11368	0.10862	☐
stack_submission (18).csv Complete (after deadline) · 2d ago	0.11286	0.10763	☐

MODEL EVALUATION

Rank: 79/3300

74	▲ 44	Jan		0.11270	49	9y
75	▼ 35	Konrad Kamiński		0.11277	56	9y
76	▲ 265	HojinYoo		0.11277	44	9y
77	▲ 33	nhlx5haze		0.11278	3	10y
78	▲ 107	grandprix		0.11284	56	9y
79	▲ 13	pxk		0.11291	144	9y
80	▼ 26	utah777		0.11308	108	9y
81	▲ 26	Shim vui ann		0.11310	57	9y

THANKS FOR WATCHING

Does anyone have any questions?

23521745@gm.uit.edu.vn
University of Information Technology - VNUHCM