Link Google Colab: Link

Họ tên	MSSV
Nguyễn Võ Ngọc Bảo	23520131
Vũ Việt Cương	23520213
Ngô Phương Nam	23520974

BÁO CÁO BƯỚC 5 - HUẤN LUYỆN, ĐÁNH GIÁ VÀ TINH CHỈNH MÔ HÌNH

Môn học: Lập trình Python cho Máy học

1. CatBoost

1.1 Giới thiệu

CatBoost là một thuật toán tăng cường gradient do Yandex phát triển, nổi bật nhờ khả năng xử lý tốt dữ liệu dạng phân loại (categorical) và không yêu cầu xử lý tiền xử lý đặc biệt như one-hot encoding. Trong bài toán dự đoán này, CatBoost được sử dụng như một baseline mạnh mẽ nhờ tính chính xác và khả năng tránh overfitting.CatBoost là một thuật toán tăng cường gradient do Yandex phát triển, nổi bật nhờ khả năng xử lý tốt dữ liệu dạng phân loại (categorical) và không yêu cầu xử lý tiền xử lý đặc biệt như one-hot encoding. Trong bài toán dự đoán này, CatBoost được sử dụng như một baseline mạnh mẽ nhờ tính chính xác và khả năng tránh overfitting.

1.2 Các bước thực hiện

- Dữ liệu được chia làm tập huấn luyện và kiểm tra theo tỉ lệ 80-20 với random_state=42 để tái lập kết quả.
- Các cột dạng phân loại được nhận diện và khai báo trong Pool (CatBoost format).
- Cấu hình mô hình:

```
CatBoostRegressor(
    iterations=1000,
    learning_rate=0.03,
    depth=12,
    leaf_reg=15,
    bagging_temperature=1.0,
    subsample=0.8,
    random_strength=1,
    eval_metric='RMSE',
    early_stopping_rounds=100,
    random_seed=42,
    verbose=100
)
```

- Quá trình huấn luyện hiển thị kết quả từng vòng, theo dõi RMSE để dừng sớm.

1.3 Đánh giá kết quả

```
[ ] y_pred_cat = cat_model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred_cat))
    print(f"CatBoost RMSE: {rmse:.4f}")

    CatBoost RMSE: 3238.9519
```

```
[ ] from sklearn.metrics import mean_absolute_error

    mae_cat = mean_absolute_error(y_test, y_pred_cat)
    print(f"CatBoost MAE on validation set: {mae:.4f}")

    CatBoost MAE on validation set: 1920.6784
```

- RMSE trên tập kiểm tra: 3236.9519
- MAE (Mean Absolute Error): 1920.6784

1.4 Phân tích đặc trưng

- Biểu đồ feature importance cho thấy các đặc trưng quan trọng nhất là:
  + model (chiếm ~60%)
  + running_km
  + status
  + run_per_year

⇒ Các biến liên quan đến loại xe và mức độ sử dụng ảnh hưởng nhiều nhất đến kết quả dự đoán.

2. XGBoost

2.1 Giới thiệu

XGBoost là một thuật toán boosting tối ưu hóa hiệu năng, được sử dụng rộng rãi nhờ tốc độ huấn luyện nhanh và khả năng tùy chỉnh cao.

2.2 Quy trình thực hiện

- Sử dụng thư viện **Optuna** để tối ưu siêu tham số cho XGBoost.
- Các tham số được tối ưu bao gồm: max_depth, learning_rate, n_estimators, subsample, colsample_bytree, reg_alpha, reg_lambda, v.v.
- Sử dụng cross_val_score để đánh giá độ chính xác trong quá trình tìm kiếm.

2.3 **Tối ưu tham số bằng Optuna**

- Sử dụng thư viện **Optuna** để tối ưu hóa tự động các siêu tham số.
- Hàm mục tiêu dùng cross_val_score với scoring là neg_mean_squared_error.

2.4 Các siêu tham số được tìm ra gồm:

```
Best_parms {
 'max_depth': 3,
 'max_leaves': 709,
 'learning_rate': 0.0435,
 'n_estimators': 1171,
 'min_child_weight': 12,
 'subsample': 0.6627,
 'reg_alpha': 0.2381,
 'reg_lambda': 0.6244,
```

```
    'colsample_bytree': 0.9043,
    'colsample_bynode': 0.6941,
    'objective': 'reg:absoluteerror',
    'n_jobs': -1
}
```

2.5 Huấn luyện và đánh giá

- Mô hình được train với XGBRegressor(**params_xgb)
- **MAE trên tập test**: **1918.0765**

⇒ Hiệu suất tốt hơn CatBoost một chút, nhờ tinh chỉnh tham số kỹ lưỡng.

2.6 Kết quả

```
y_pred_xgb = xgb_model.predict(X_test)
mae_xgb    = mean_absolute_error(y_test, y_pred_xgb)
print(f"XGBoost MAE: {mae_xgb:.4f}")
```

```
XGBoost MAE: 1918.0705
```

- MAE trên tập test: **1918.0705**
- Kết quả này tốt hơn một chút so với CatBoost.

3. Kết hợp CatBoost + XGBoost

3.1 Mục tiêu

Sau khi đánh giá riêng lẻ hai mô hình **XGBoost** và **CatBoost**, ta tiến hành kết hợp chúng để cải thiện hiệu suất dự đoán bằng cách tận dụng ưu điểm của từng mô hình.

3.2 Lý do chọn Ensemble

- **XGBoost** hoạt động tốt trên dữ liệu đã qua xử lý và tối ưu siêu tham số.
- **CatBoost** rất mạnh khi làm việc với dữ liệu chứa nhiều đặc trưng phân loại (categorical features) và không cần mã hóa.
- Hai mô hình có cấu trúc và cơ chế học khác nhau ⇒ sai số dự đoán có thể mang tính **bù trừ**.

→ Kết hợp dự đoán của cả hai mô hình thường dẫn đến kết quả ổn định và chính xác hơn.

3.3 Phương pháp kết hợp

Sử dụng phương pháp **trung bình đơn giản (simple averaging)** của hai mô hình:

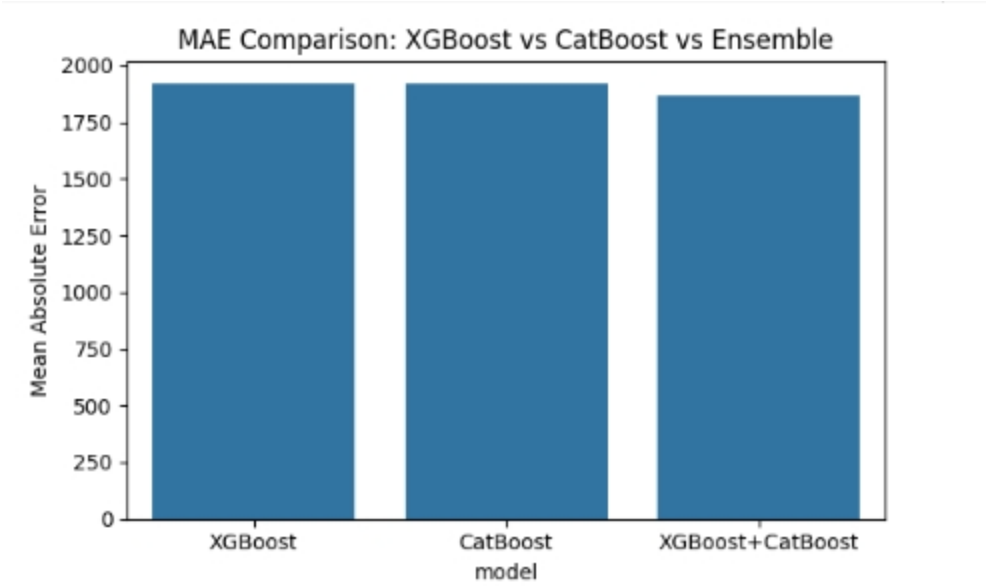
```
y_pred_ens = (y_pred_xgb + y_pred_cat) / 2
```

3.4 Đánh giá kết quả

```
[ ] y_pred_ens = (y_pred_xgb + y_pred_cat) / 2
    mae_ens = mean_absolute_error(y_test, y_pred_ens)
    print(f"Ensemble MAE (XGBoost+CatBoost): {mae_ens:.4f}")
```

```
Ensemble MAE (XGBoost+CatBoost): 1865.8889
```

- **MAE trên tập kiểm tra**: **1865.8889**
- Biểu đồ thể hiện MAE các mô hình:



Mô hình	**MAE**
XGBoost	1918.0705
CatBoost	1920.6784
CatBoost + XGBoost	**1865.8889**

⇒ Mô hình tổ hợp cho kết quả **tốt nhất**, giảm sai số khoảng 3% so với từng mô hình riêng biệt.