Báo cáo cuối kì môn KHDL

...

Dự đoán giá nhà ở TP. HCM

Võ Nhật Vinh - 1612815
Nguyễn Ngọc Khải - 1612909

Câu hỏi đặt ra

- Nhiều người muốn mua nhà/bán nhà nhưng không biết giá cả thị trường thế nào ?

→ Cần có mô hình dự đoán giá nhà phù hợp.

→ Tránh bị lừa khi không biết về giá nhà.

Thu thập dữ liệu

- Thu thập dữ liệu nhà ở trên trang chợ tốt.
- Dữ liệu thu thập hợp pháp.
- Dữ liệu gồm 20 cột, và 17742 dòng.

```
import urllib.robotparser
rp = urllib.robotparser.RobotFileParser()
rp.set_url('https://nha.chotot.com/tp-ho-chi-minh/mua-ban-bat-dong-san')
rp.read()
rp.can_fetch('*', 'https://nha.chotot.com/tp-ho-chi-minh/mua-ban-nha-dat')
# Kết quả sẽ là True hoặc False
```

```
True
```

```
price             17742 non-null int64
type              17742 non-null object
time              17742 non-null object
link              17742 non-null object
address           17742 non-null object
title             17742 non-null object
area              17742 non-null int64
bedroom_num       17742 non-null object
house_type        17742 non-null object
toilet_num        17742 non-null object
direction         17742 non-null object
legcal_doc        17742 non-null object
block_name        17742 non-null object
total_floor       17742 non-null object
housing_feature   17742 non-null object
city              17734 non-null object
district          17736 non-null object
ward              17719 non-null object
street            16935 non-null object
description       17742 non-null object
```

Tiền xử lí dữ liệu

- Lọc trùng
- Tách thêm thuộc tính, tạo thuộc tính mới: Địa chỉ -> số nhà, đường, phường, quận.
- Phân bin dữ liệu và gán label.
- Xử lí dữ liệu thiếu.
- Loại bỏ những giá trị bị sai.
- Onehot encoding
- Chuẩn hóa

Tiền xử lí dữ liệu

- Phân bin dữ liệu:

```
# diện tích lớn hơn 260 cho vô 1 bin
train_all['area'] = train_all['area'].apply(lambda x: x if x < 260  else 260)
# lớn hơn 7 phòng ngủ cho là 7 phòng ngủ
train_all['bedroom_num'] = train_all['bedroom_num'].apply(lambda x: x if x < 7  else 7)
# lớn hơn 6 phòng vệ sinh cho vô 1 bin
train_all['toilet_num'] = train_all['toilet_num'].apply(lambda x: x if x < 6  else 6)
```

Tiền xử lí dữ liệu

- Xử lý dữ liệu thiếu:

```
class FillNA(BaseEstimator, TransformerMixin):
    def fit(self, X_df, y=None):
        return self

    def transform(self, X_df, y=None):
        res_df = X_df.copy()
        other_cols = ['housing_feature', 'house_type', 'street', 'district', 'ward']
        _0_cols = ['bedroom_num', 'toilet_num']
        for i in other_cols:
            res_df[i].fillna('Khác',inplace=True)
        for i in _0_cols:
            res_df[i].fillna(0.0,inplace=True)
        res_df['legcal_doc'].fillna('Không có', inplace=True)
        return res_df
```

Tiền xử lí dữ liệu

- One hot encoding

```
onehot = OneHotEncoder(handle_unknown='ignore')
```

- Chuẩn hóa

```
scaler = StandardScaler()
```

Huấn luyện mô hình

Sử dụng các mô hình phân lớp sau:

- Random Forest
- Logistic Regression
- MLP
- KNN

Huấn luyện mô hình

Kết quả trên tập train và tập val:

- Random Forest
  Train score: 97.26%
  Val score: 0.7576857386848848

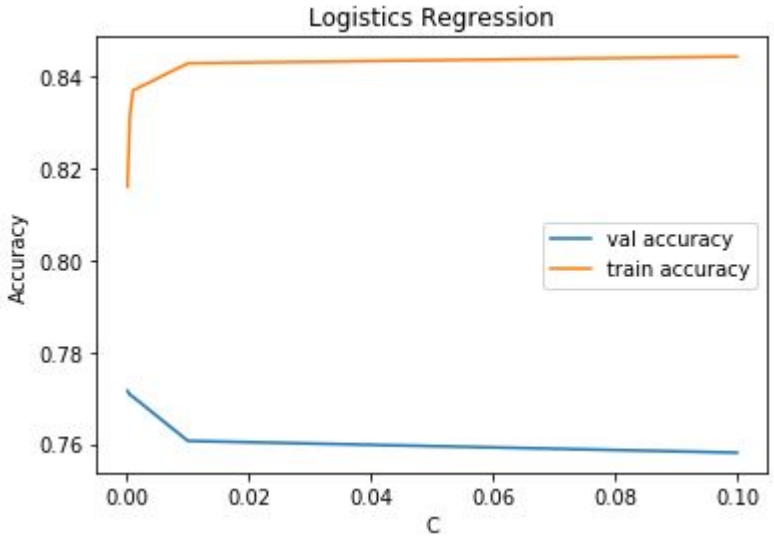
- Logistic Regression
  Train score: 0.843819973130318
  Val score: 0.7549103330486764

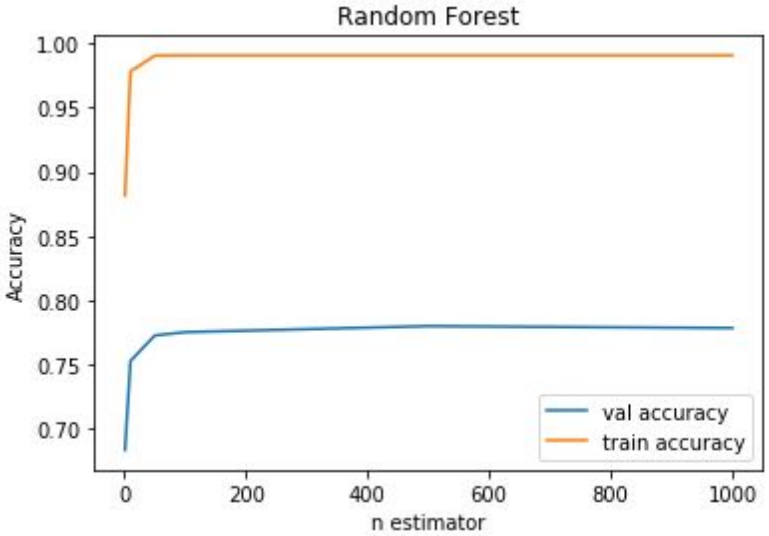
- MLP:
  Train score: 0.9296909986565159
  Val score: 0.7224594363791631

- KNN
  Train score: 0.7875055978504254
  Val score: 0.6833902647309992

Kết quả

Chạy các tham số với mô hình logistic regression, random forest
Logistics Regression: C_params = [0.0001, 0.0005, 0.001, 0.01, 0.1]
Random Forest: n_estimators_params = [1, 10, 50, 100, 500, 1000]





Kết quả

Chọn được mô hình Random Forest với tham số n estimator =500.

Chạy trên tập test đạt độ chính xác 79.9%