PHÂN TÍCH VÀ DỰ ĐOÁN GIÁ NHÀ DỰA TRÊN TẬP DỮ LIỆU MỞ AIRBNB

Group 15

Class BDML434077_21_1_01

Instuctor Quach Dinh Hoang

December 26th 2021

Faculty of Information Technology

HCMC University of Teachnology and Education

ABSTRACT

Kể từ năm 2008, khách và chủ nhà đã sử dụng Airbnb để mở rộng khả năng đi du lịch và trình bày cách trải nghiệm thế giới cá nhân, độc đáo hơn. Tệp dữ liệu này thì bao gồm các thông tin cần thiết để tìm hiểu thêm về các máy chủ lưu trữ, địa lý sẵn có,... Từ đó có thể tiến đến xây dựng các mô hình để dự đoán cho tương lai. Đó thực sự là một tiềm năng to lớn để thúc đẩy các chủ nhà và cũng như người thuê nhà có thể nắm bắt được xu hướng của giá nhà trong tương lai của thị trường.

1. INTRODUCTION

Trước khi chọn một nơi để thuê nhà thì dùng sẽ luôn cân nhắc về các yếu tố như là chất lượng, vị trí và đặc biệt là giá cả cho thuê. Và với thời điểm hiện tại, việc đưa ra giá cả cho thuê nhà hợp lý cho người dùng là vấn đề vô cùng khó khăn cho người cho thuê. Với mục tiêu là giúp phát triển thành công một ứng dụng có thể dự đoán giá nhà dựa trên các thuộc tính còn lại (kinh độ, vĩ độ, khu vực,...). Sau khi phân tích và đánh giá trên tập dữ liệu chúng ta sẽ tiến hành sử dụng các thuật toán học máy (linear regression, decision trees, random forest,..) để đưa ra các giá trị dự đoán cho tương lai.

2.DATA

Bài toán sẽ thực hiện phân tích dựa trên tập dữ liệu mở Airbnb của thành phố NewYork. Tập dữ liệu có hơn 48.800 quan sát được ghi nhận và được chia sẻ trên diễn đàn kaggle https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data. Dữ liệu mà ta có được vẫn còn khá thô sơ, để có thể phục vụ cho việc phân tích, ta tiền xử lý dữ liệu.

2.1 Tranform

Id là mã số của nhà và không mang nhiều ý nghĩa lắm nên ta sẽ loại bỏ.

Name là tên của nhà và cũng không mang nhiều ý nghĩa nên sẽ được loai bỏ.

Host ID và **Host name** lần lượt tên mã máy chủ và tên máy chủ, 2 thuộc tính này cũng sẽ được loại bỏ.

neighbourhood_group và **neighbourhood** lần lượt là nhóm khu vực và khu vực.

latitude và **longitude** lần lượt là vĩ độ và kinh độ.

room_type là thuộc tính mô tả loại phòng cho thuê.

price là giá cả cho thuê.

ninium_night là số đêm tối thiểu cho thuê.

number_of_reviews là số lượt đánh giá.

last_review là ngày nhận đánh giá cuối cùng.

reviews_per_month là trung bình đánh giá trên tháng.

calculated_host_listings_count là số lượng danh sách máy chủ được tính toán.

availability_365 là số ngày khả dụng trong năm.

2.2 CLEAN DATA

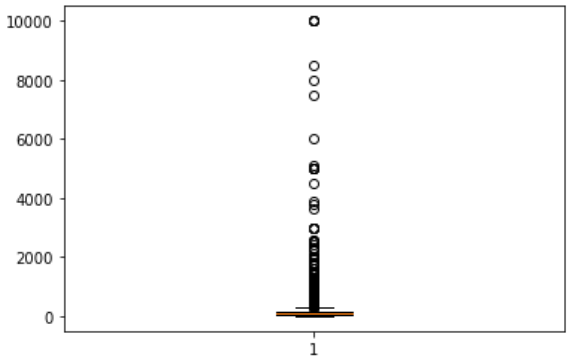
Đầu tiên chúng ta sẽ chuyển đổi cột "last_review" từ chuỗi thành ngày. Sau đó chúng ta tiến hành tìm kiếm ngày cuối cùng được đánh giá trên tập dữ liệu và lấy nó làm điểm zero. Từ đó chúng ta đếm được khoảng thời gian từ điểm zero đến ngày cuối cùng được review của mỗi nhà. Và cuối cùng là chuyển về dưới dạng biến phân loại là không có ngày đánh giá, cách đây 1 tuần, cách đây 1 tháng,...

Bước tiếp theo chúng ta chuyển các cột còn lại từ dạng chuỗi sang dạng số nguyên hoặc số thực.

Bước cuối cùng là thay thế các giá trị null thành 0.

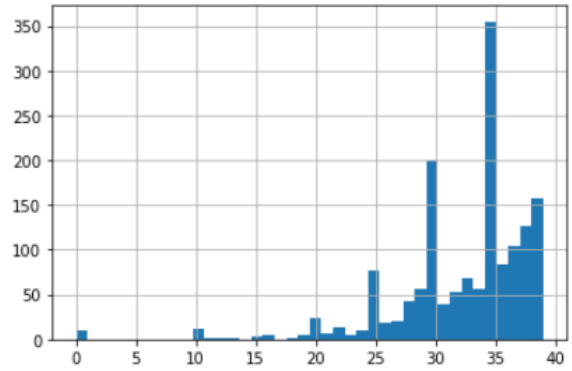
3. DATA VISULIZATION

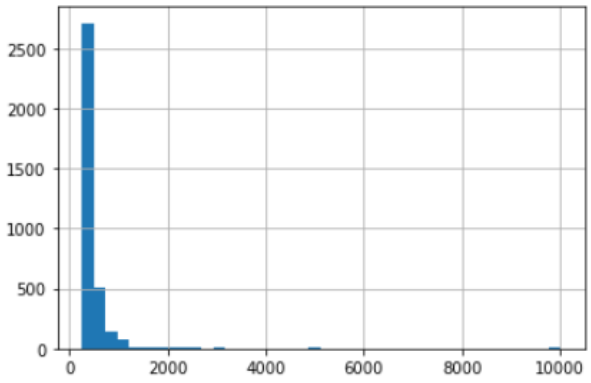
- Price



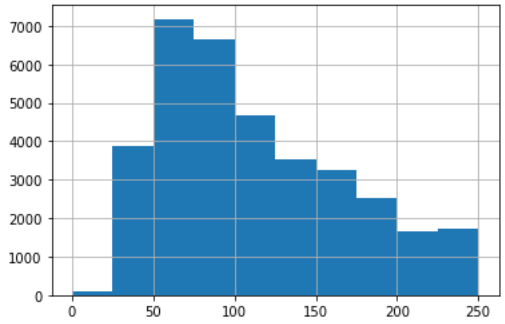
Từ biểu đồ chúng ta có thể nhận thấy rằng giá cả cho thuê thì tập trung dày đặc ở khoảng từ 0 – 250 đô. Từ khoảng 250 – 2000 đô thì tập trung vừa. Và trên 2000 thì chỉ xuất hiện rất ít.

Tiếp đến chúng ta sẽ xem sự phân bố về giá của price ở tầm dưới 40 đô và trên 250 đô thì như thế nào.



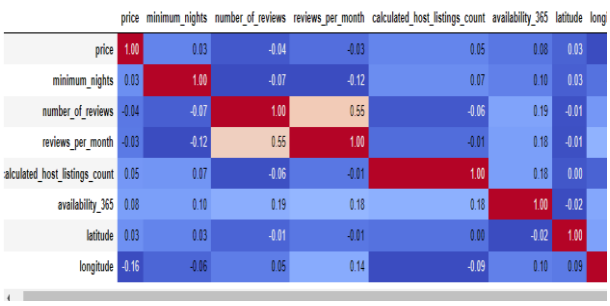


Bây giờ thì chúng ta sẽ xem dữ liệu ở mức dưới 250 đô.



Có thể thấy là dao động khá là đều ở khoảng giá 25 – 250 đô. Vậy nên chúng ta sẽ loại bớt dữ liệu quá chênh lệch để tránh gây hại cho mô hình của chúng ta.

Và cuối cùng chúng ta sẽ xem qua ma trận tương quan giữa các biến trước khi bắt tay vào xây dựng mô hình.



4. BUILD MODEL

Để xây dựng mô hình, ta sẽ phân vùng tập dữ liệu thành tập training (80%) và testing(20%). Sau khi phân vùng dữ liệu ta sẽ xây dựng mô hình trên *training set* và sau đó chúng ta sẽ tiến hành đánh giá trên tập test set.

Đầu tiên chúng ta sử dụng mô hình linear regression cho bài toán này.

```
lr_evaluator_raw = RegressionEvaluator(predictionCol="predic
print("R Squared (R2) on test data = %g" % lr_evaluator_raw.
lr_evaluator_raw = RegressionEvaluator(predictionCol="predic
print("RMSE (R2) on test data = %g" % lr_evaluator_raw.evalu
```

```
R Squared (R2) on test data = 0.13579
RMSE (R2) on test data = 184.575
```

Ở mô hình này thì chúng ta thấy được hệ số R-squared cực thấp. Để cải thiện mô hình chúng ta sẽ thử thay đổi một số siêu tham số trên mô hình của chúng ta.

```
lr = LinearRegression(maxIter=10)

# We use a ParamGridBuilder to construct a grid of param
# TrainValidationSplit will try all combinations of valu
# the evaluator.
paramGrid = ParamGridBuilder()\
    .addGrid(lr.regParam, [0.1, 0.01]) \
    .addGrid(lr.fitIntercept, [False, True])\
    .addGrid(lr.elasticNetParam, [0.0, 0.5, 1.0])\
    .build()

# In this case the estimator is simply the linear regres
# A TrainValidationSplit requires an Estimator, a set of
tvs = TrainValidationSplit(estimator=lr,
                           estimatorParamMaps=paramGrid,
                           evaluator=RegressionEvaluator
                           # 70% of the data will be use
                           trainRatio=0.7)

# Run TrainValidationSplit, and choose the best set of p
model = tvs.fit(train)
```

Và kết quả mang lại là hệ số R-squared có tăng lên như vẫn còn ở mức thấp.

```
R Squared (R2) on test data = 0.559639
RMSE (R2) on test data = 37.2255
```

Vậy nên chúng ta sẽ chuyển qua 2 thuật toán tiếp theo để xây dựng mô hình là decison tree và random forest. Trên cả 2 thuật toán này chúng ta cũng điều sẽ sử dụng siêu tham số để tăng độ chính xác cho mô hình.

Tuning hyparameters

```
[33]: rf = RandomForestRegressor(featuresCol = 'features', labelCol='label', maxDepth = 14)
rf_model = rf.fit(train_df)
rfPredictions = rf_model.transform(test_df)

rf_evaluator = RegressionEvaluator(predictionCol="prediction", labelCol="label",metricN
print("R Squared (R2) on test data = %g" % rf_evaluator.evaluate(rfPredictions))
rf_evaluator = RegressionEvaluator(predictionCol="prediction", labelCol="label",metricN
print("RMSE on test data = %g" % rf_evaluator.evaluate(rfPredictions))
```

```
R Squared (R2) on test data = 0.612062
RMSE on test data = 35.1
```

Sau khi sử dụng mô hình thì kết quả tốt nhất là nhóm nhận được là khi sử dụng maxDepth = 14 trên mô hình Random Forest.

Vì đây là kết quả tốt nhất mà nhóm đã thực hiện được vậy nên nhóm quyết định thử sử dụng mô hình trên dự đoán để xem độ chênh lệch giữa kết quả với thực tế.

```
rfPredictions.show(5)
```

```
+--------------------+-----+------------------+
|            features|label|        prediction|
+--------------------+-----+------------------+
|(236,[0,1,2,3,4,5...|   51|  68.6468192429817|
|(236,[0,1,2,3,4,5...|   60| 64.77113899405927|
|(236,[0,1,2,3,4,5...|   46| 67.20110719140075|
|(236,[0,1,2,3,4,5...|   75| 71.91950773443997|
|(236,[0,1,2,3,4,5...|   95|60.874555377365404|
+--------------------+-----+------------------+
```

Như chúng ta có thể thấy độ chính xác mà mô hình đem đến là chưa cao. Vậy nên mô hình này chỉ dừng lại ở mức tham khảo chứ không hoàn toàn đưa vào áp dụng được.

5 CONCLUSION

5.1 Hạn chế.

Qua quá trình phân tích và xây dựng mô hình thì kết quả đem lại cho nhóm còn ở mức quá thấp. Không thể áp dụng vào thực tiễn. Ngoài ra nhóm chỉ mức chú trọng vào việc tìm mô hình hợp với bộ dữ liệu mà chưa xem đến quá trình làm sạch và thay đổi dữ liệu có ảnh hưởng như thế nào đến với mô hình.

5.2 Hướng phát triển.

Nếu có cơ hội thì nhóm sẽ tìm hiểu và thay đổi dữ liệu đầu vào cũng như áp dụng các thuật toán khác để đem đến mô hình tốt hơn.

6. CONTRIBUTIONS

MSSV	STT	NAME	PROGRESS
18133048	**36**	**Nguyễn Hoàn Thai**	**Data cleaing and linear regression**
18133024	**18**	**Ngô Phi Lít**	**Decision tree and random forest**

REFERENCES

[1] Nguyen, Van Tuan. *Mô Hình Hồi Quy Và Khám Phá Khoa Học*. Nhà xuất bản tổng hợp thành phố Hồ Chí Minh, 2020. Printed book

[2]https://spark.apache.org/docs/1.5.2/ml-decision-tree.html

[3]https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.mllib.tree.RandomForest.html

[4] Quach, Dinh Hoang. Lecture, 2021. Slide and video