Supervised Learning

IRONHACK Project #6

CONTEXT & PROBLEM DEFINITION

I work for an apartment rental company which is Airbnb competitor and we would like to build a component to help our customers fix a price for their location.

I would like to predict the price per night for a location in Paris.

PROCESS

01

Data Cleaning

Large reduction of columns.

Handling missing values.

New columns and transformation of categorical data.

02

Exploratory

Exploration of data.

Top 5 most expensive neighborhoods.

Outliers.

03

Preprocessing

Scaling using MixMax Scale.

Feature Engineering using RFE and filter methods.

04

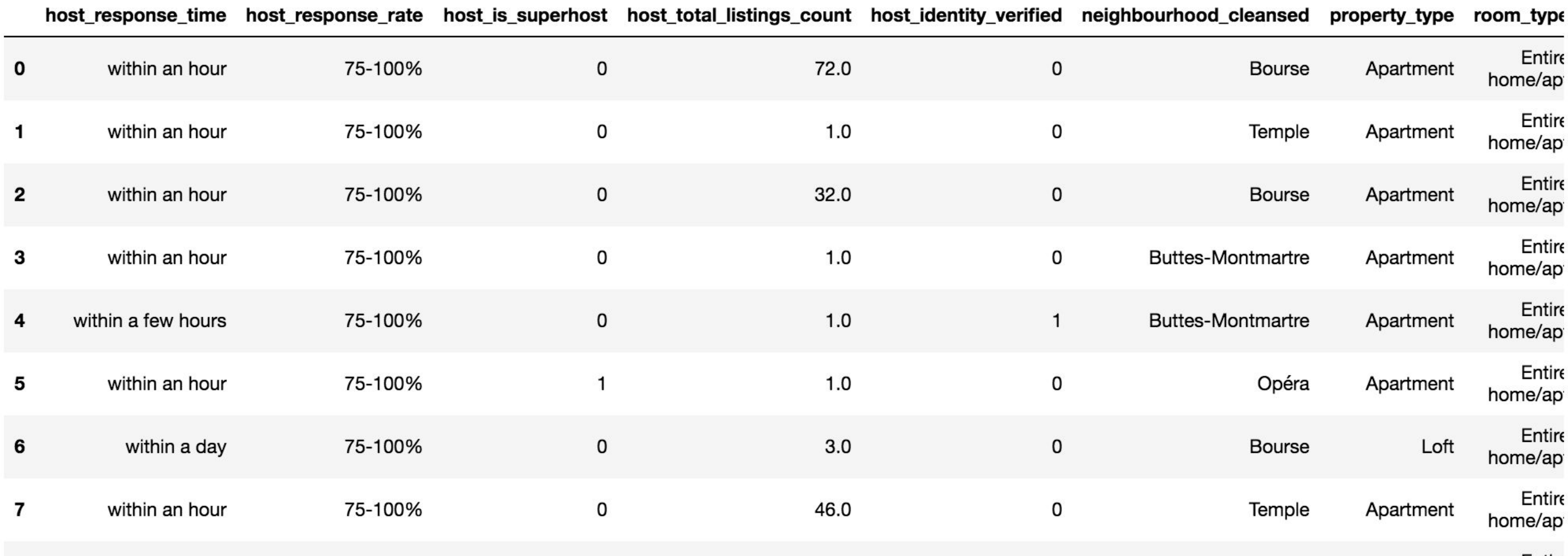
Modelling

Building Linear Regression, KNN and Random Forest models.

Improvement of model performance.

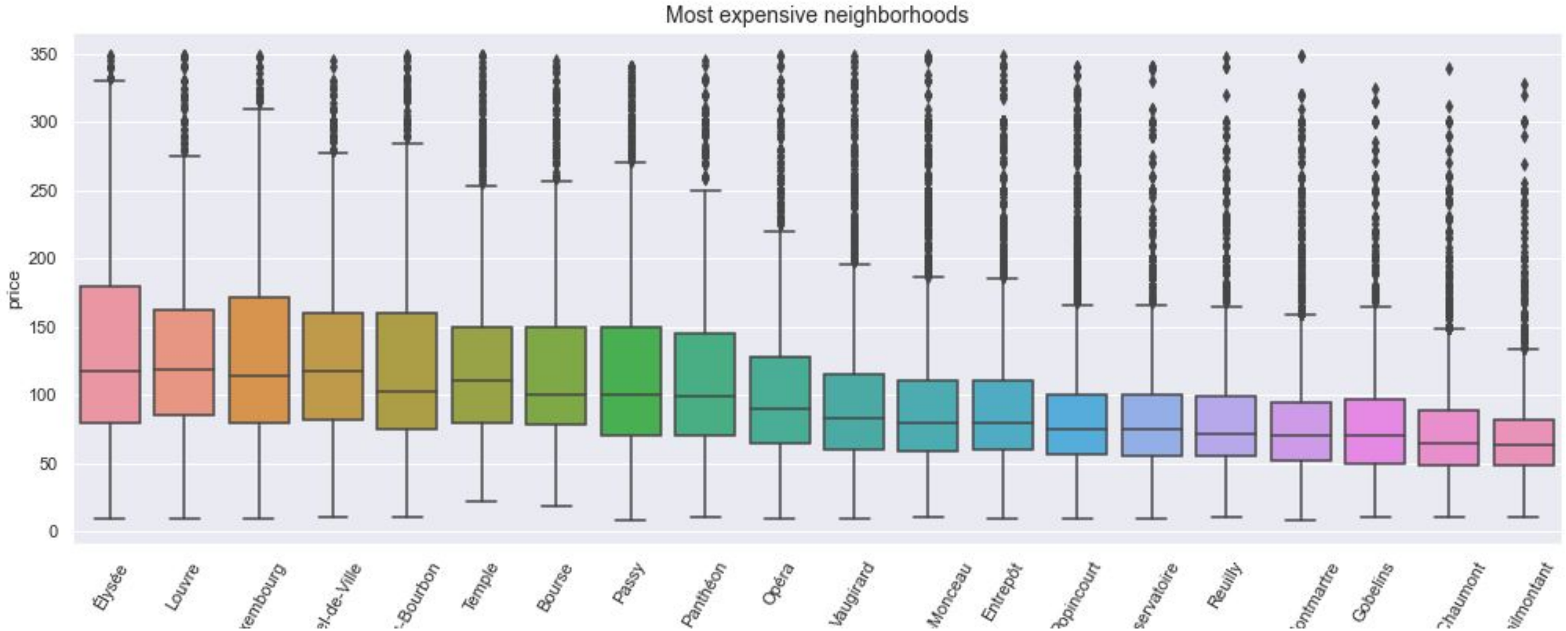
AIRBNB DATASET

- Data source: insideairbnb.com
- Use of data scrapped last November to avoid bias on prices
- Raw data shape: (65493, 106) ⇒ Clean data shape without dummies (62837, 21)



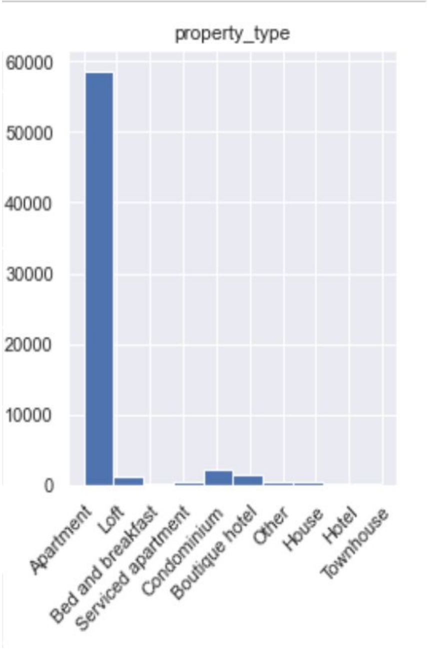
	host_response_time	host_response_rate	host_is_superhost	host_total_listings_count	host_identity_verified	neighbourhood_cleansed	property_type	room_type
0	within an hour	75-100%	0	72.0	0	Bourse	Apartment	Entire home/ap
1	within an hour	75-100%	0	1.0	0	Temple	Apartment	Entire home/ap
2	within an hour	75-100%	0	32.0	0	Bourse	Apartment	Entire home/ap
3	within an hour	75-100%	0	1.0	0	Buttes-Montmartre	Apartment	Entire home/ap
4	within a few hours	75-100%	0	1.0	1	Buttes-Montmartre	Apartment	Entire home/ap
5	within an hour	75-100%	1	1.0	0	Opéra	Apartment	Entire home/ap
6	within a day	75-100%	0	3.0	0	Bourse	Loft	Entire home/ap
7	within an hour	75-100%	0	46.0	0	Temple	Apartment	Entire home/ap

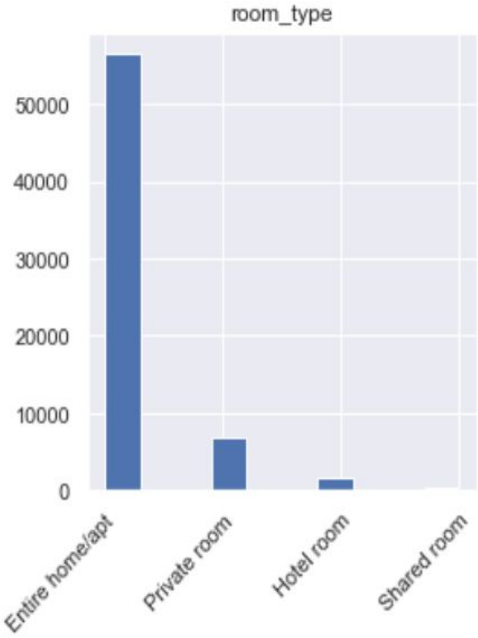
02 EXPLORATORY

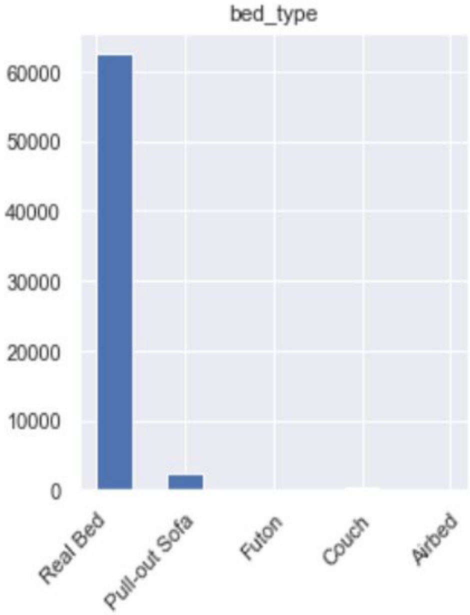


⇒ Top 5 most expensive neighborhoods: Elysée, Louvre, Luxembourg, Hotel de Ville, Palais Bourbon

02 EXPLORATORY



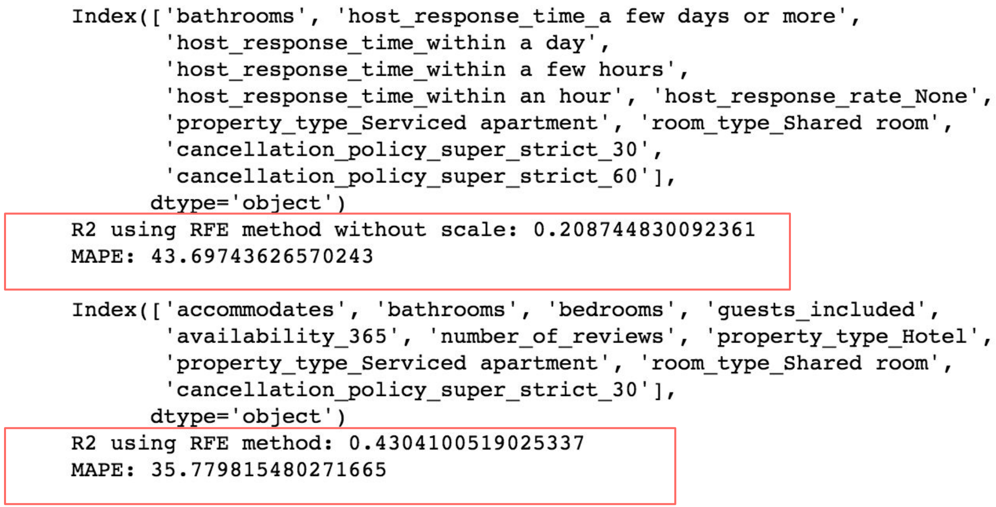




⇒ Most common locations are Apartment, to rent the Entire Apt/Room, that can accommodates 2-3 people, with 1 bathroom and 1 bedroom.

03 PREPROCESSING

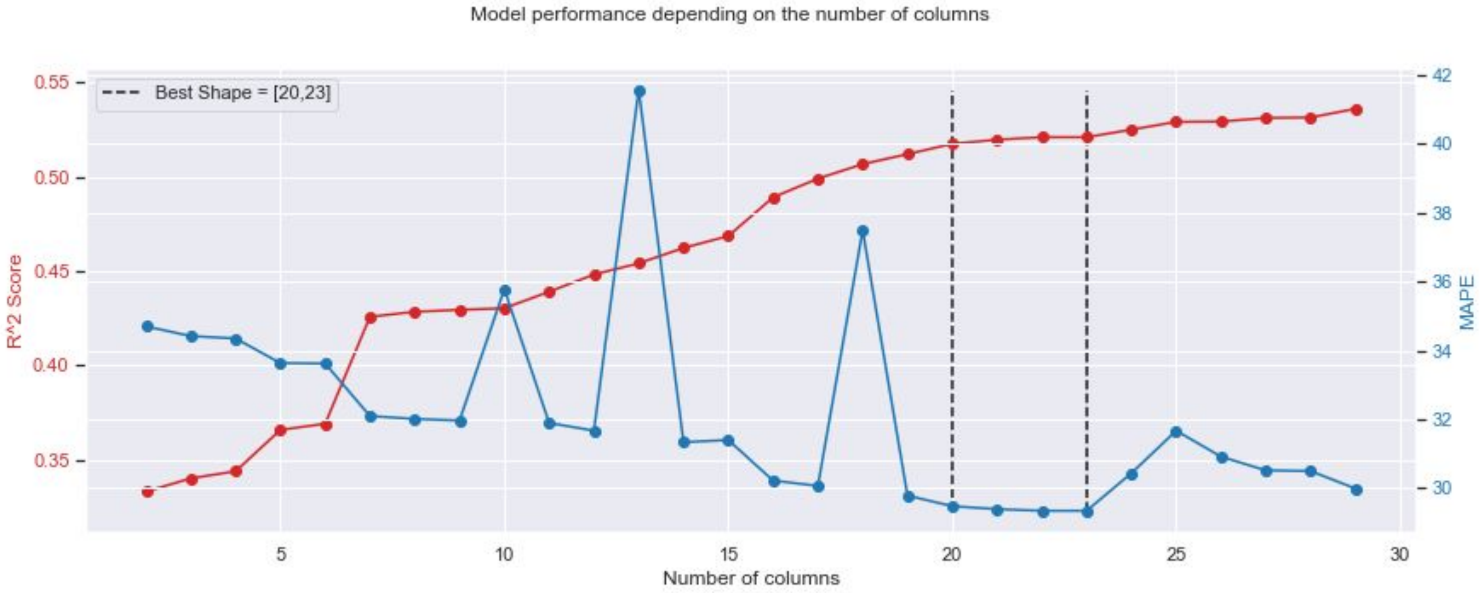
- I scaled the data using MixMax Scale because of many dummies and boolean columns.



```
Index(['bathrooms', 'host_response_time_a few days or more',
       'host_response_time_within a day',
       'host_response_time_within a few hours',
       'host_response_time_within an hour', 'host_response_rate_None',
       'property_type_Serviced apartment', 'room_type_Shared room',
       'cancellation_policy_super_strict_30',
       'cancellation_policy_super_strict_60'],
      dtype='object')
R2 using RFE method without scale: 0.208744830092361
MAPE: 43.69743626570243

Index(['accommodates', 'bathrooms', 'bedrooms', 'guests_included',
       'availability_365', 'number_of_reviews', 'property_type_Hotel',
       'property_type_Serviced apartment', 'room_type_Shared room',
       'cancellation_policy_super_strict_30'],
      dtype='object')
R2 using RFE method: 0.4304100519025337
MAPE: 35.779815480271665
```

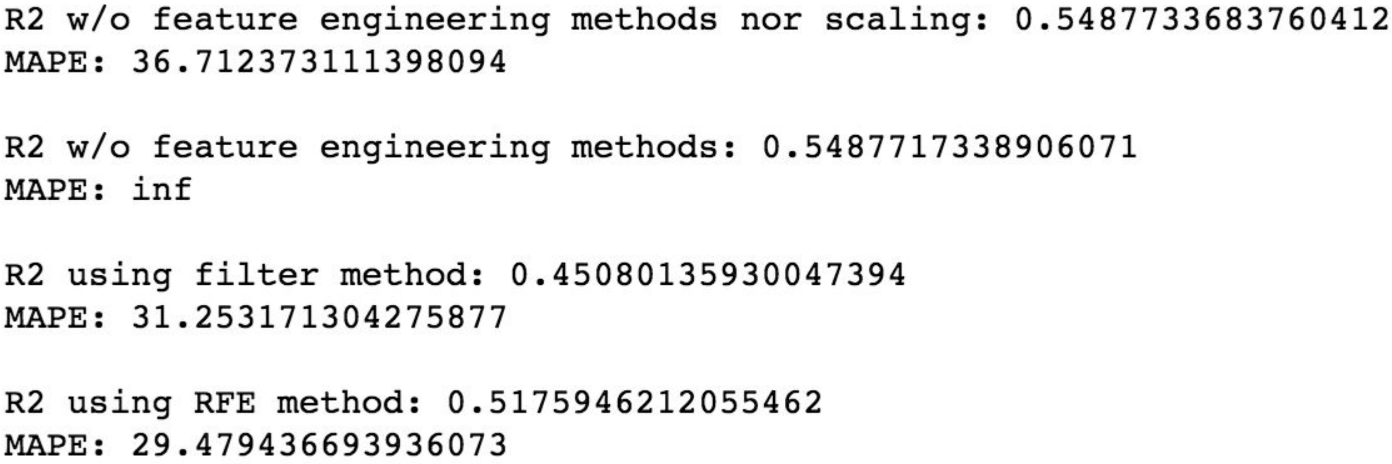
⇒ There is a much higher performance with scaled data

03 PREPROCESSING



⇒ By making tests on RFE method, I managed to have a better performances with this preprocessing method.

03 PREPROCESSING

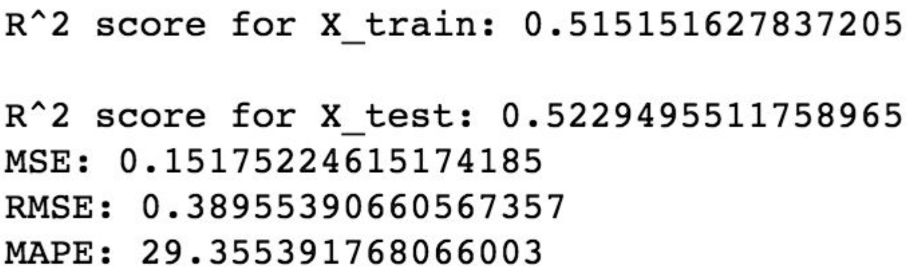


```
R2 w/o feature engineering methods nor scaling: 0.5487733683760412
MAPE: 36.712373111398094

R2 w/o feature engineering methods: 0.5487717338906071
MAPE: inf

R2 using filter method: 0.45080135930047394
MAPE: 31.253171304275877

R2 using RFE method: 0.5175946212055462
MAPE: 29.479436693936073
```

⇒ RFE method seems to perform better now

04 MODELLING

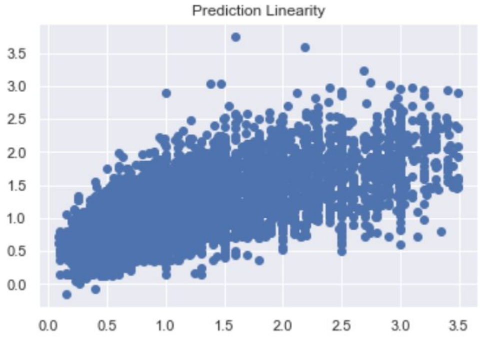
Linear Regression

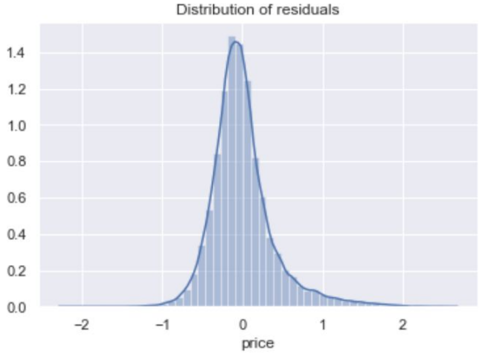


```
R^2 score for X_train: 0.515151627837205

R^2 score for X_test: 0.5229495511758965
MSE: 0.15175224615174185
RMSE: 0.38955390660567357
MAPE: 29.355391768066003
```

⇒ Model performance doesn't seem so good but remember it improved with RFE optimization

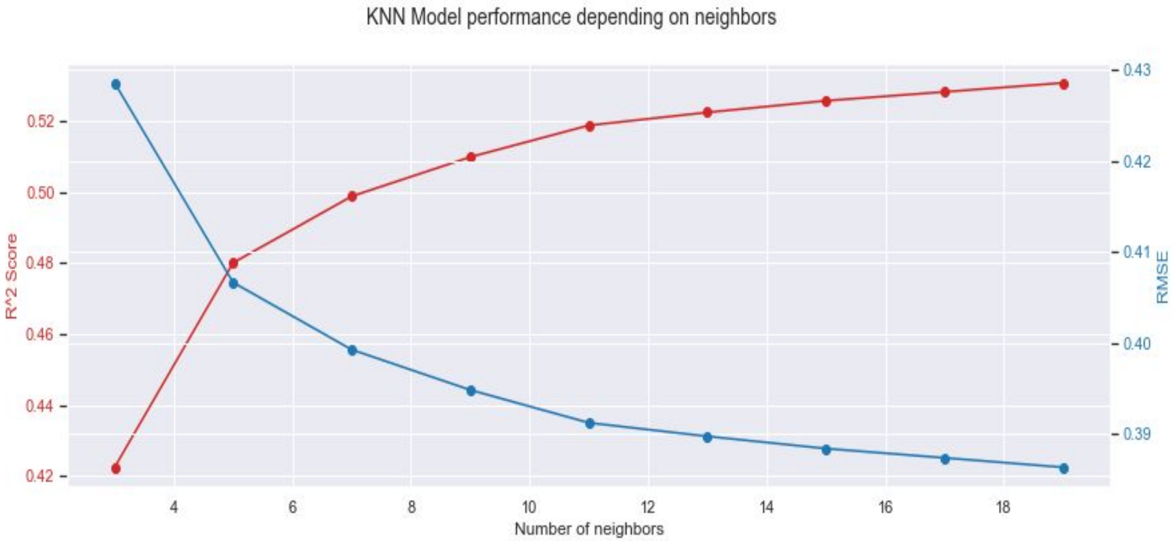
⇒ Having negative values





04 MODELLING

KNeighborsRegressor

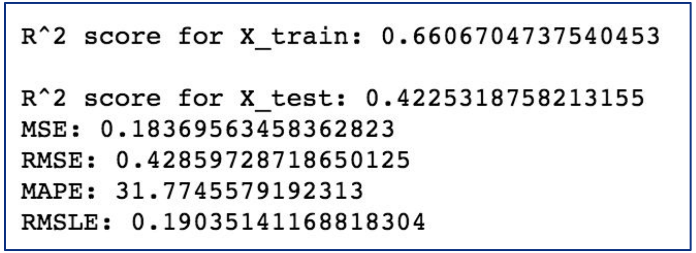


⇒ Improved the model performance and reduced errors by increasing the number of neighbors.

04 MODELLING

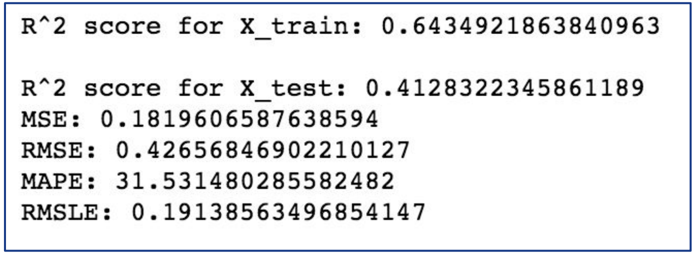
KNeighborsRegressor

Sample 1, random_state = 8

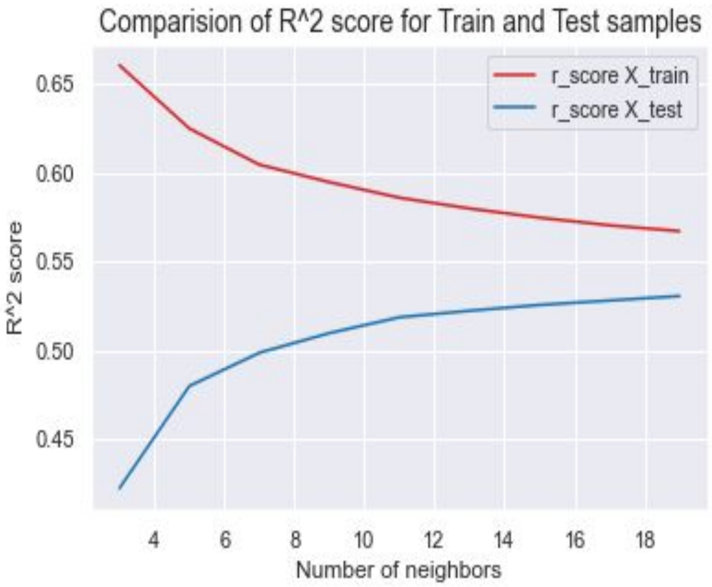


```
R^2 score for X_train: 0.6606704737540453

R^2 score for X_test: 0.4225318758213155
MSE: 0.18369563458362823
RMSE: 0.42859728718650125
MAPE: 31.7745579192313
RMSLE: 0.19035141168818304
```

Sample 2, random_state = 42



```
R^2 score for X_train: 0.6434921863840963

R^2 score for X_test: 0.4128322345861189
MSE: 0.1819606587638594
RMSE: 0.42656846902210127
MAPE: 31.531480285582482
RMSLE: 0.19138563496854147
```

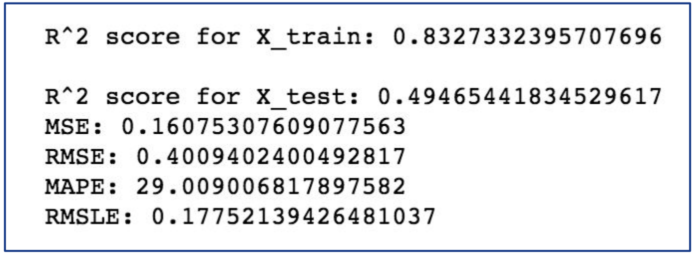


⇒ Overfitting which has been fixed by increasing the number of neighbors as well.

04 MODELLING

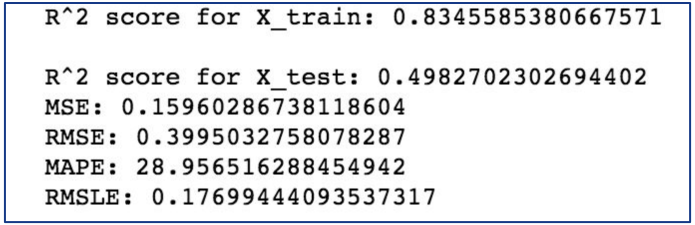
Random Forest Regressor

Sample 1, estimators=100 (default)



```
R^2 score for X_train: 0.8327332395707696

R^2 score for X_test: 0.49465441834529617
MSE: 0.16075307609077563
RMSE: 0.4009402400492817
MAPE: 29.009006817897582
RMSLE: 0.17752139426481037
```

Sample 2, estimators=1000



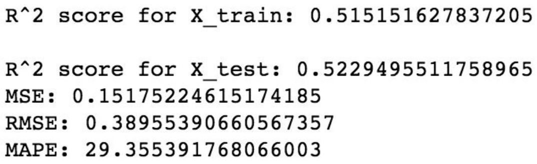
```
R^2 score for X_train: 0.8345585380667571

R^2 score for X_test: 0.4982702302694402
MSE: 0.15960286738118604
RMSE: 0.3995032758078287
MAPE: 28.956516288454942
RMSLE: 0.17699444093537317
```

⇒ Overfitting problem not solved

⇒ Couldn’t manage to test with categorical data (without dummies)

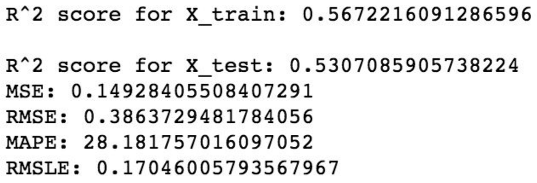
CONCLUSION

Linear Regression



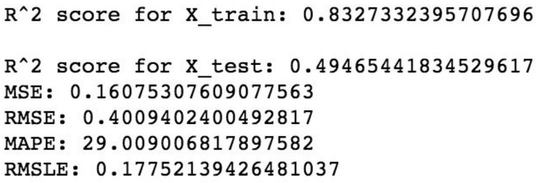
```
R^2 score for X_train: 0.515151627837205

R^2 score for X_test: 0.5229495511758965
MSE: 0.15175224615174185
RMSE: 0.38955390660567357
MAPE: 29.355391768066003
```

KNeighborsRegressor



```
R^2 score for X_train: 0.5672216091286596

R^2 score for X_test: 0.5307085905738224
MSE: 0.14928405508407291
RMSE: 0.3863729481784056
MAPE: 28.181757016097052
RMSLE: 0.17046005793567967
```

Random Forest Regressor



```
R^2 score for X_train: 0.8327332395707696

R^2 score for X_test: 0.49465441834529617
MSE: 0.16075307609077563
RMSE: 0.4009402400492817
MAPE: 29.009006817897582
RMSLE: 0.17752139426481037
```

⇒ Linear Regression and KNN performs better although results are not so good

⇒ Linear Regression model assumptions are not met (Normality and still 0.23% of outliers

OBSTACLES

Poor results from the beginning

From the very beginning the R^2 of models started around 0.40 so it’s hard to make it really good.

⇒ Continuous iteration on preprocessing and models parameters.

⇒ Maybe try to keep more columns from row dataset and interpolate more missing values.

Overfitting of models

KNN and Random Forest were overfitted.

⇒ Need to test with different sample to confirm the overfitting.

⇒ Playing on models parameters may allow to reduce the overfitting effect.

⇒ Cross validation may help.

IMPROVEMENTS

Better data cleaning

Use z-score to get a more precise cleaning of outliers. Try to interpolate more missing values to keep other kind of feature.

Try other preprocessing methods

Try Sequential Selection or PCA to see if they impact the performance of models.

Normalization of dataset

Transform the data to make them normally distributed in order to meet Linear Regression assumptions

Try other models

Try Lasso, XGboost or Nayes Bayes models to see if we can get better performance.