Predicting Toronto AirBnB Prices

Sixing Cao, Tian Gan, Yun Sum Wong, Tiffany Yeung

Objective

- Analyzing Airbnb data for Toronto to predict the price of an Airbnb rentals in different neighborhoods in Toronto using machine learning techniques discussed in class (Linear regression, Lasso regression, Random Forest regressor) .

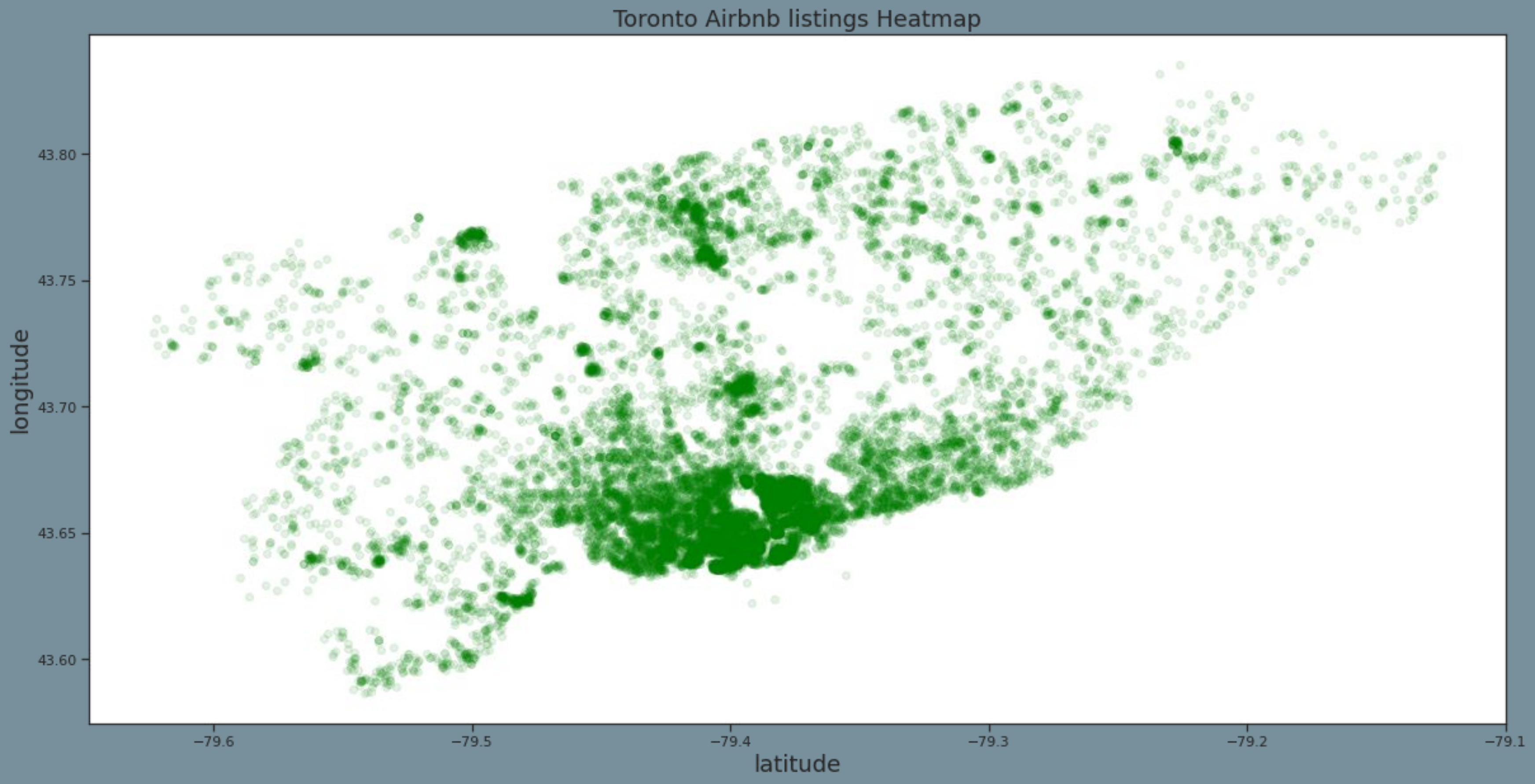
Assumptions and Methodologies I

- Assumptions:
  - Higher number of reviews mean that more people stayed at the Airbnb and more people saw the listing
- Data cleaning:
  - Dropped airbnbs listed higher than $700 as outliers (4.64% of the data)
  - Filled in nulls with character values (e.g. columns such as “last_review” or “reviews_per_month”)
- 10 features including:
  - Community Council
  - Room_type
  - Price
  - min # of nights
  - # of reviews, reviews/month
  - availability
  - Name length
  - Calculated host listings count

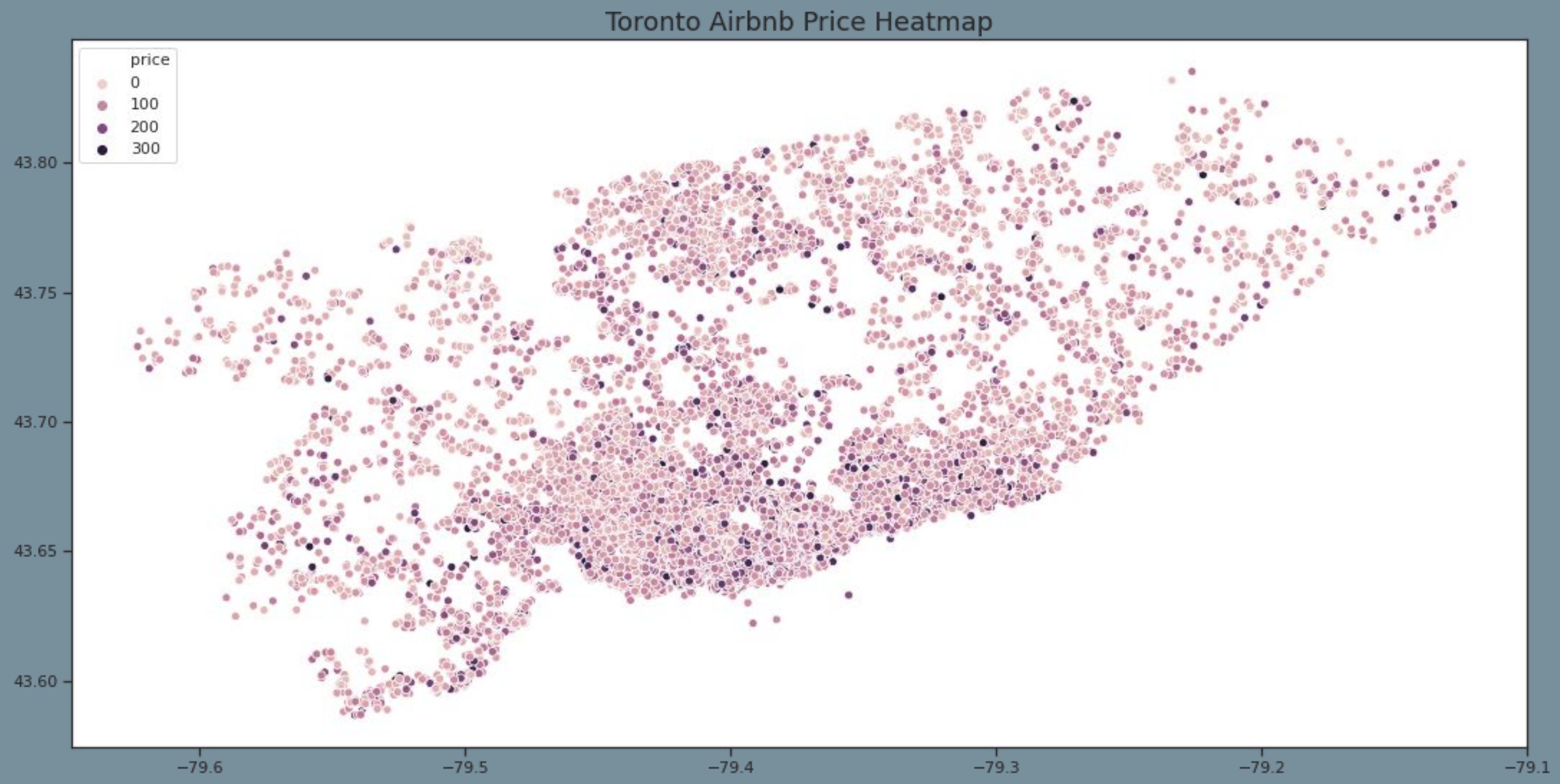
Assumptions and Methodologies II

- Used one hot encoding to convert categorical features into binary dummy variables (features: room_type, community council)
- Apply linear regression model to have a baseline to compare other models (lasso generator and random forest generator)

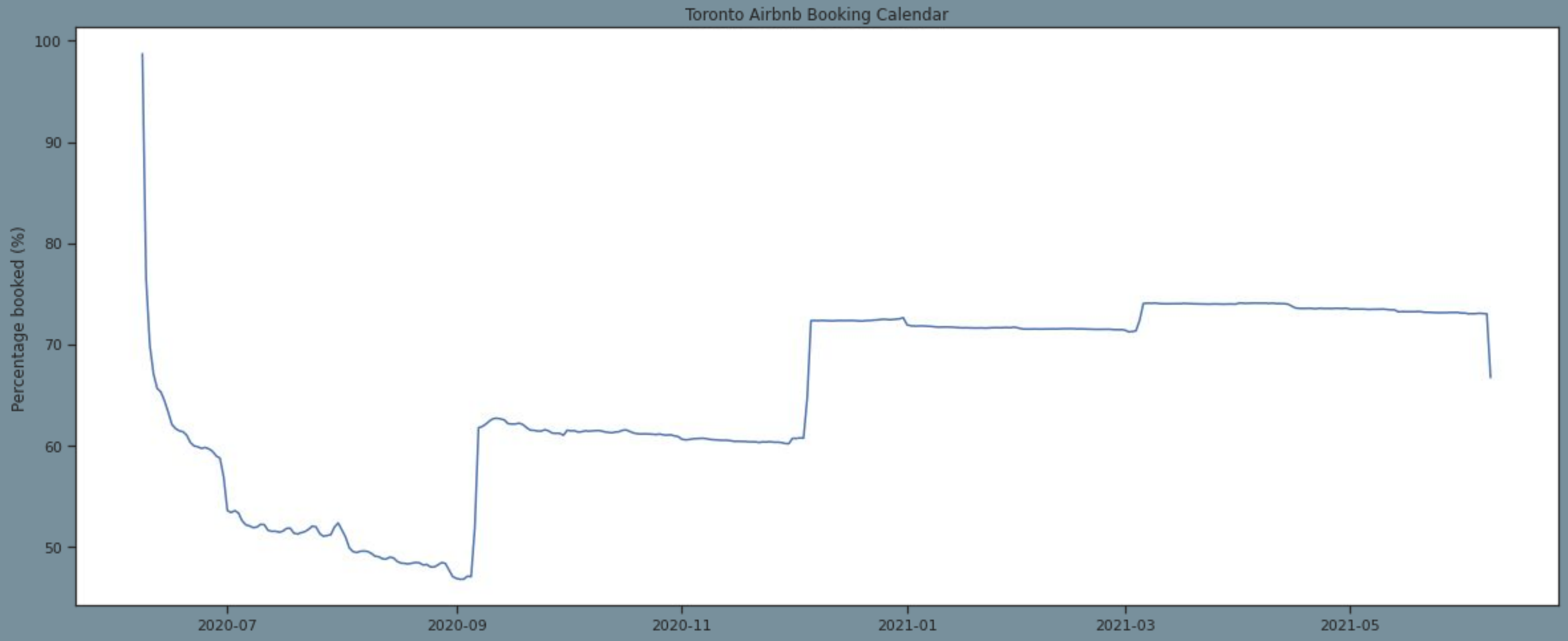
Results and Discussion - Listings Heatmap



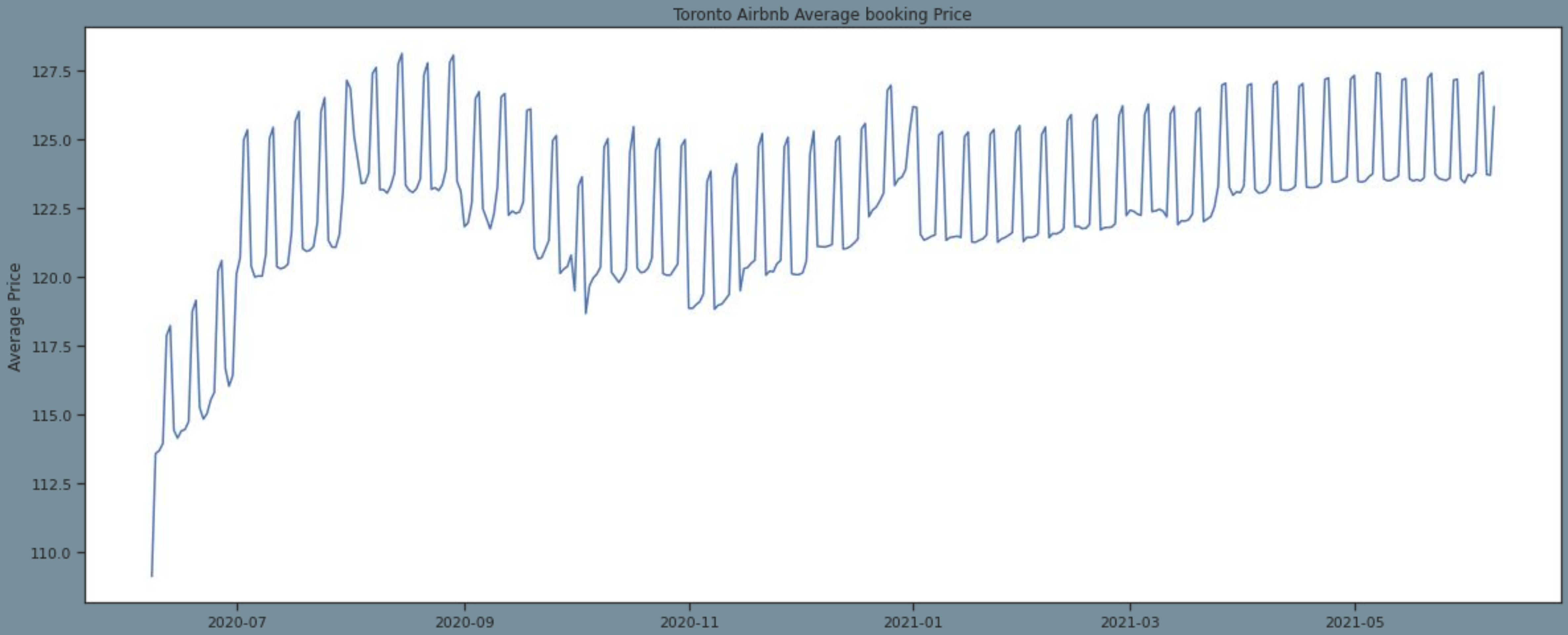
Results and Discussion - Prices Heatmap



Results and Discussion - Predicted Booking Calendar



Results and Discussion - Predicted Avg Price



Results and Discussion

- Linear regression model:
  - RMSE: 49.21, r-squared of 0.339
  - Seems to be an issue with multicollinearity between features (cond. #: 3.34e+03)
  - Data seems to be skewed (1.03) and deviation from normal distribution (kurtosis @4.189 vs of normal distribution would be@3)
  - Condition number @3.34e+03

- Due to strong multicollinearity, we tried Lasso Regression to select a subset of parameters:
  - RMSE: 47.78, r-squared 0.377
  - This regularised model did way better than normal linear regression

Results and Discussion

Linear Regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.339
Model:                            OLS   Adj. R-squared:                  0.338
Method:                 Least Squares   F-statistic:                     673.5
Date:                Tue, 18 Aug 2020   Prob (F-statistic):               0.00
Time:                        02:01:30   Log-Likelihood:                -84011.
No. Observations:               15807   AIC:                         1.680e+05
Df Residuals:                   15794   BIC:                         1.681e+05
Df Model:                          12
Covariance Type:            nonrobust
==================================================================================================
                                    coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------
const                           111.7353      1.939     57.633      0.000     107.935     115.536
minimum_nights                   -0.0408      0.012     -3.505      0.000      -0.064      -0.018
number_of_reviews                -0.0828      0.012     -6.807      0.000      -0.107      -0.059
reviews_per_month                 0.7354      0.405      1.818      0.069      -0.058       1.529
calculated_host_listings_count    0.1430      0.042      3.429      0.001       0.061       0.225
availability_365                  0.0269      0.003      8.744      0.000       0.021       0.033
name_length                       0.1321      0.036      3.619      0.000       0.061       0.204
ng_North York                    -0.7469      1.620     -0.461      0.645      -3.922       2.428
ng_Scarborough                   -6.8203      1.922     -3.548      0.000     -10.588      -3.053
ng_Toronto and East York         21.0956      1.360     15.517      0.000      18.431      23.760
rt_Hotel room                   -29.8790      7.070     -4.226      0.000     -43.736     -16.022
rt_Private room                 -61.9298      0.866    -71.485      0.000     -63.628     -60.232
rt_Shared room                  -87.4276      2.943    -29.707      0.000     -93.196     -81.659
==============================================================================
Omnibus:                     2344.827   Durbin-Watson:                   1.987
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3725.853
Skew:                           1.030   Prob(JB):                         0.00
Kurtosis:                       4.189   Cond. No.                     3.34e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.34e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Lasso Regression

	Variable	Coefficients
43	nh_Financial District	39.071991
42	nh_Fashion District	28.699260
38	nh_Entertainment District	28.213108
51	nh_Harbourfront	27.038636
97	nh_Port Union	24.658715
...	...	...
26	nh_Crescent Town	-17.378421
63	nh_Keelesdale	-23.339250
147	rt_Hotel room	-24.915373
148	rt_Private room	-59.072624
149	rt_Shared room	-81.222715

150 rows × 2 columns

Results and Discussion

- Random forest generator:
    - Overall bias reduced because each tree is trained on a subset of data
    - Functions well with categorical data
    - Applying feature importance, we were able to determine the features that have the most weight
    - Found that private room, reviews/month, name length, and 365 availability had the most influence on the price

```
RandomForestRegressor(n_estimators=300)
```

```
### We get R squared value at 91.2%! There is obviously a problem of overfitting:(

print(regrRM.score(x_trainL11, y_trainL11))
y_predL1= regrRM.predict(x_testL11)
print(np.sqrt(metrics.mean_squared_error(y_testL11,y_predL1)))
```

```
0.9123481639858101
47.00893355354233
```

	Variable	FeatureImportance
148	rt_Private room	0.263532
2	reviews_per_month	0.106922
5	name_length	0.105750
4	availability_365	0.088408
1	number_of_reviews	0.086288
...	...	...
98	nh_Princess	0.000058
111	nh_Scarborough Village	0.000047
85	nh_North Park	0.000044
81	nh_Mount Olive	0.000043
75	nh_Markland Woods	0.000038

```
RandomForestRegressor(n_estimators=200, max_depth =
50, min_samples_split = 5,min_samples_leaf =4)
```

```
### We get a smaller value for R squared
print(regrRM2.score(x_trainL11, y_trainL11))
y_predL1= regrRM2.predict(x_testL11)
print(np.sqrt(metrics.mean_squared_error(y_testL11,y_predL1)))
```

```
0.7153395450261368
46.833505445861896
```

	Variable	FeatureImportance
148	rt_Private room	0.354005
2	reviews_per_month	0.101097
5	name_length	0.092268
4	availability_365	0.086472
1	number_of_reviews	0.078301
...	...	...
81	nh_Mount Olive	0.000000
73	nh_Malvern West	0.000000
58	nh_Humberlea	0.000000
25	nh_Corktown	0.000000
75	nh_Markland Woods	0.000000

Challenges!

- Too many variables of neighbourhood values pointing to the same neighbourhood, e.g. Toronto

```
#df.loc[df['smart_location'].isin(['Toronto, Canada','Toronto , Canada','toronto, Canada', '토론토, Canada','多伦多,
```

- Data aggregation for neighbourhood values, need different datasource to categorize small location to higher level (community council)

- Dealing with rooms that have a high minimum number of nights

Lessons Learnt

- A lot of more time than expected is needed to clean and make sense of real data
- There is no one correct way of analyzing the data, and finding the correct model is based on experience and trial and error
- Feature importance is essential to ensuring your model is not overfit