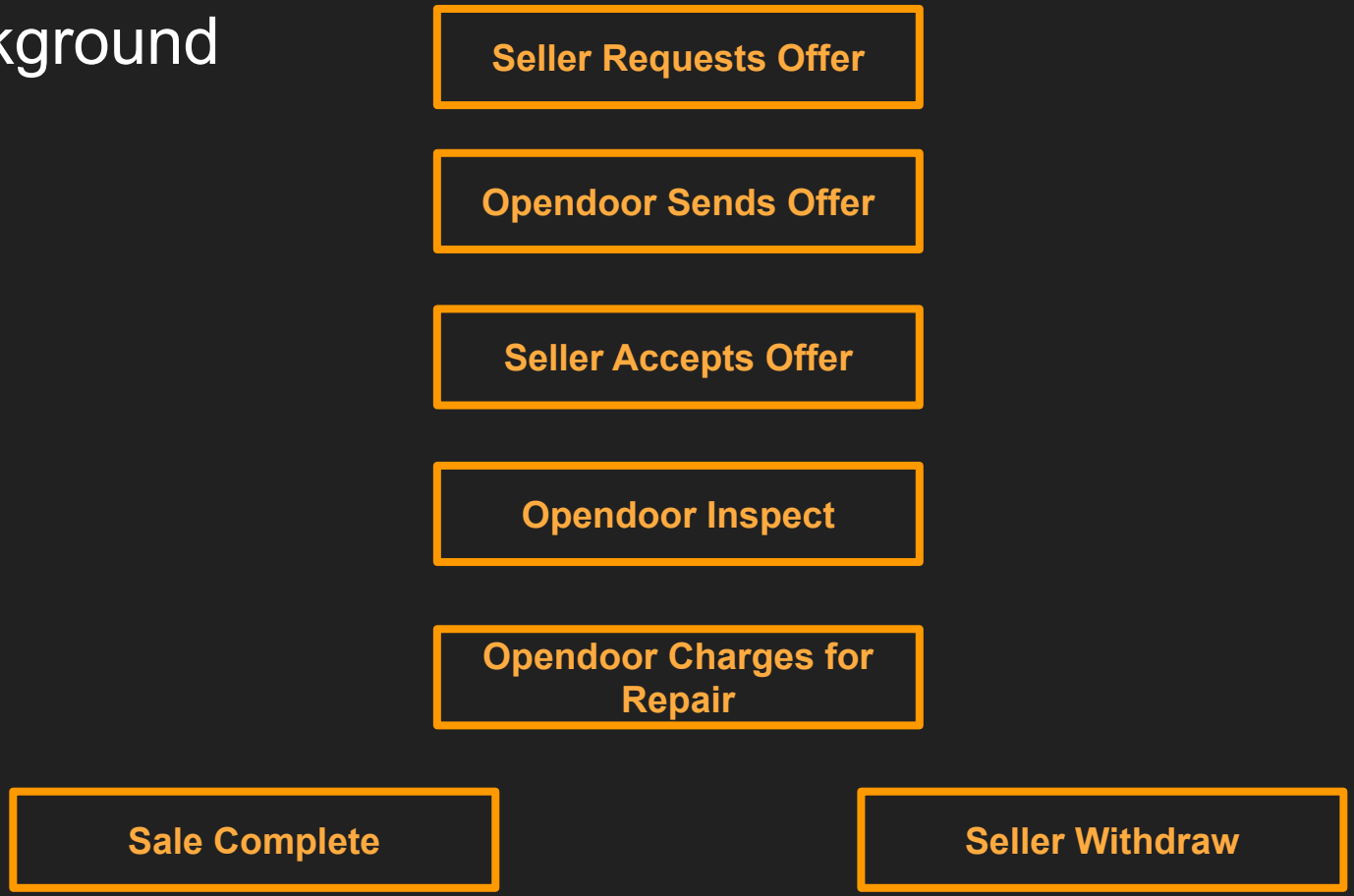
Predicting Common Scopes

Bijan Seyednasrollah

April 1st, 2021

Background


Seller Requests Offer
Opendoor Sends Offer
Seller Accepts Offer
Opendoor Inspect
Opendoor Charges for Repair
Sale Complete
Seller Withdraw

Problem Statement

Traditional Method: Inspectors estimate the repair cost.

New Method: Can we use data science to eliminate inspection?

What we know:

- Every sale is equivalent of $5000 in profit
- Every $150 in repair charge increase the withdrawal by 1%
- Eliminating inspector’s visit can lead to 5% decline in withdrawal

Data

Features:

House characteristics such as: number of bedrooms, age, location, area, etc.

Target:

Common Scope estimates

Approach

Several possible approaches:

- Regression Analysis:
  - Linear Regression
  - Lasso Regression
  - Ridge Regression
- Neural Network
- AutoML

Approach

Several possible approaches:

- Regression Analysis:
  - Linear Regression
  - Lasso Regression
  - Ridge Regression
- Neural Network
- AutoML

Implemented as customized class:
CommonScope(model, params, ...)

Approach

Several possible approaches:

- Regression Analysis:
  - Linear Regression
  - Lasso Regression
  - Ridge Regression
- Neural Network — Deep Learning model built in TensorFlow
- AutoML

Approach

Implemented approaches:

- Regression Analysis:
  - Linear Regression
  - Lasso Regression
  - Ridge Regression
- Neural Network
- AutoML

Google Cloud Platform

Data / Modeling Challenges

1. **Missing Data**
   Imputation per data type

2. **Multifarious data (various types)**
   boolean, integer, float, categorical, text

3. **Multicollinearity**
   Regularization and Variance Inflation Factor

4. **Positive-Only Target**
   Transformations

```
summarize(cs.train_data)
```

	column_name	data_type	unique_values	percent_missing
has_renovation	has_renovation	bool	2	0.000000
in_gated_community	in_gated_community	bool	2	0.000000
renovation_amount	renovation_amount	int64	40	0.000000
pool	pool	float64	2	0.000000
total_finished_sq_ft	total_finished_sq_ft	float64	2520	0.000000
above_grade_sq_ft	above_grade_sq_ft	float64	2454	0.000000
age	age	float64	61	0.000000
f_days_since_prev_close	f_days_since_prev_close	float64	3910	0.000000
common_scope	common_scope	float64	9032	0.000000
bedrooms	bedrooms	float64	7	0.009158
bathrooms_half	bathrooms_half	float64	6	0.018317
garage_spaces	garage_spaces	float64	5	0.018317
sq_ft	sq_ft	float64	2486	0.054950
exterior_stories	exterior_stories	float64	3	0.082425
basement_finished_sq_ft	basement_finished_sq_ft	float64	247	0.201484
bathrooms_full	bathrooms_full	float64	5	1.053210
bathrooms	bathrooms	float64	21	44.399670
pool_above_ground	pool_above_ground	float64	2	44.399670
basement_unfinished_sq_ft	basement_unfinished_sq_ft	float64	201	64.428977
flip_token	flip_token	object	10919	0.000000
front_yard_condition	front_yard_condition	object	1	0.000000
valuation_date	valuation_date	object	10919	0.000000

Modeling Framework

I developed a Python module featuring the ***CommonScope*** class. The ***CommonScope*** class contains all the necessary methods to load and preprocess the data, train the model, and predict for new dataset. The class takes care of customized missing data handling and user-defined arguments to control the model.

Inside the class, three regression model are presented: Linear Regression, Lasso Regression and Ridge Regression.

I compared the outcome with results from GCP's AutoML as well as a customized Neural Network model built in TensorFlow.

CommonScope Initialization

Method I: The basic mehtod to instantiate the class is very straight-forward by passing the train and hold out files.

The data files get processed and loaded in the class. If any exception occures, it will be caught and messaged to the user.

```
# loading the class at the time of creating the instance

cs = CommonScope(model = 'Lasso',
                 drop_colinears  = False,
                 train_file = 'data/development_df.csv',
                 holdout_file = 'data/holdout_candidate_df.csv')
```

```
"data/development_df.csv" was loaded as the training data set: 10919 x 40
"data/holdout_candidate_df.csv" was loaded as the hold-out data set: 1580 x 39
```

CommonScope Initialization

Method II: Alternatively, these can be done in three steps for more transparency, as follows:

```
# class instantiation
cs = CommonScope(model = 'Ridge')

# loading the training file
cs.load_data(file_path='data/development_df.csv')

# loading the holdout file
cs.load_data(file_path='data/holdout_candidate_df.csv', holdout = True)
```

```
"data/development_df.csv" was loaded as the training data set: 10919 x 40
"data/holdout_candidate_df.csv" was loaded as the hold-out data set: 1580 x 39
```

CommonScope Data Exploration

```
# summarizing the training data

summarize(cs.train_data)
```

	column_name	data_type	unique_values	percent_missing
has_renovation	has_renovation	bool	2	0.000000
in_gated_community	in_gated_community	bool	2	0.000000
renovation_amount	renovation_amount	int64	40	0.000000
pool	pool	float64	2	0.000000
total_finished_sq_ft	total_finished_sq_ft	float64	2520	0.000000
above_grade_sq_ft	above_grade_sq_ft	float64	2454	0.000000
age	age	float64	61	0.000000
f_days_since_prev_close	f_days_since_prev_close	float64	3910	0.000000
common_scope	common_scope	float64	9032	0.000000
bedrooms	bedrooms	float64	7	0.009158
bathrooms_half	bathrooms_half	float64	6	0.018317

CommonScope Preprocessing

Boolean ⇒ Binary vector, Missing data imputed as False

Integer ⇒ Number, Missing data imputed with **median of the training dataset**

Float ⇒ Number, Missing data imputed with **mean of the training dataset**

Categorical ⇒ One-Hot Encoding, Missing data imputed with most-frequent of the training dataset

Text ⇒ Parsed, then transformed to binary matrices, no imputation needed.

- Columns with majority as missing were dropped
- Non-linear terms were added to improve the performance of the model

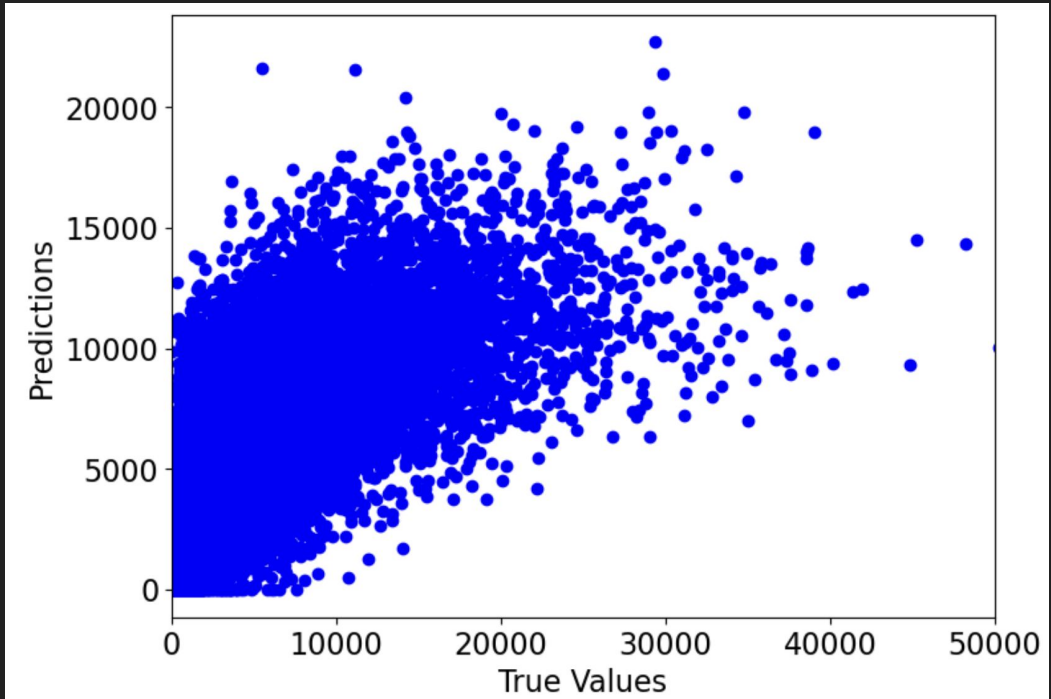
CommonScope Training

```
cs.fit(verbose = True,
       model_pars = {'positive': True,
                     'alpha': .2},
       model = 'Lasso')
```

```
{'train': {'R²': 0.34431613749556433, 'MAE': 3633.5876029172587, 'rMSE': 5046.289487018266},
'test': {'R²': 0.3518650354946905, 'MAE': 3545.394468202433, 'rMSE': 4846.795866542509}}
```

```
{'model': Lasso(alpha=0.2, normalize=True, positive=True),
 'model_summary': {'train': {'R²': 0.34431613749556433,
   'MAE': 3633.5876029172587,
   'rMSE': 5046.289487018266},
  'test': {'R²': 0.3518650354946905,
   'MAE': 3545.394468202433,
   'rMSE': 4846.795866542509}}}
```

CommonScope Evaluation



CommonScope Predictions

Predictions for the Hold-out Data

```
holdout_preds = cs.predict(df= cs.holdout_data)
```

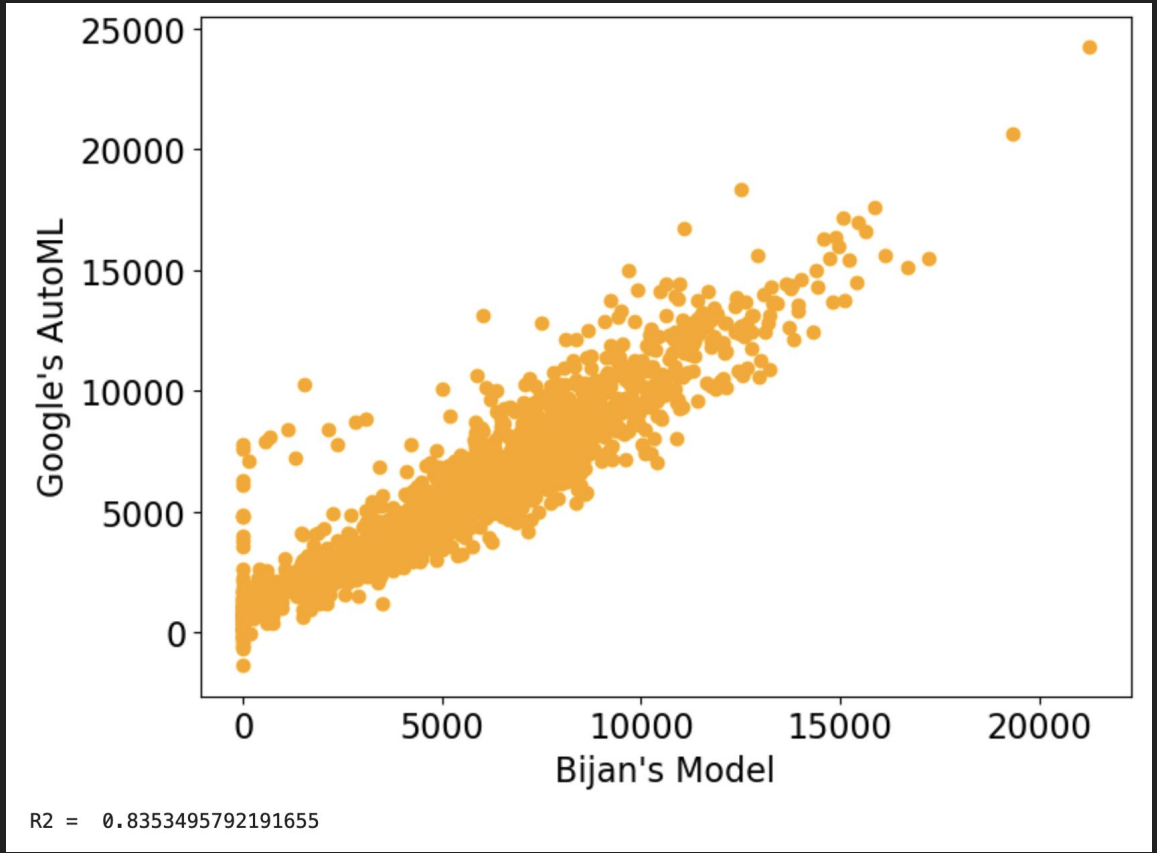
```
holdout_preds = cs.predict(df= cs.holdout_data)

holdout_preds = pd.DataFrame({'flip_token': cs.holdout_data.flip_token.to_list(),
                              'prediction': holdout_preds.reshape(len(holdout_preds))})

holdout_preds.head()
```

	flip_token	prediction
0	560EQGK7A1YM	8852.687377
1	N283R19X7QWF	8320.343073
2	7HNA53BSS2D92	3220.380259
3	NN5R58HDGFB2	1295.606664
4	2XDMCE3HDXNNP	12495.564361

CommonScope vs. GCP's AutoML



Analyzing for the Business Value

On average the change in estimates is -45.66 USD,

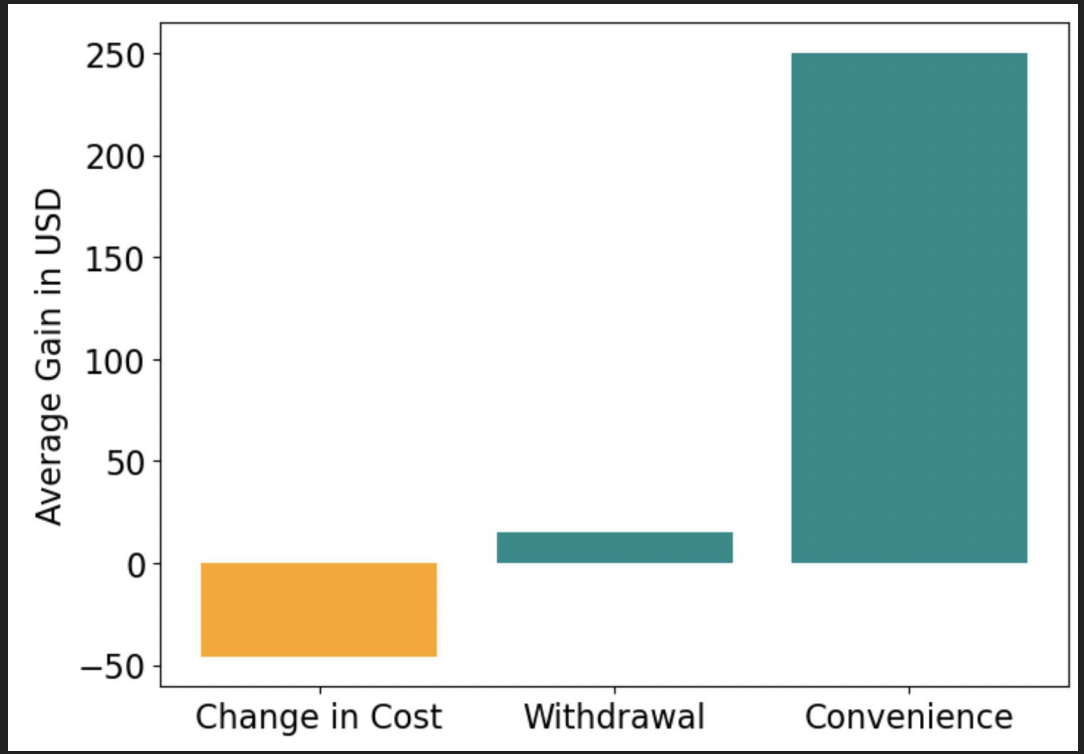
which is equivalent of -45.66 USD.

Changed withdrawal for estimates change: -0.30%

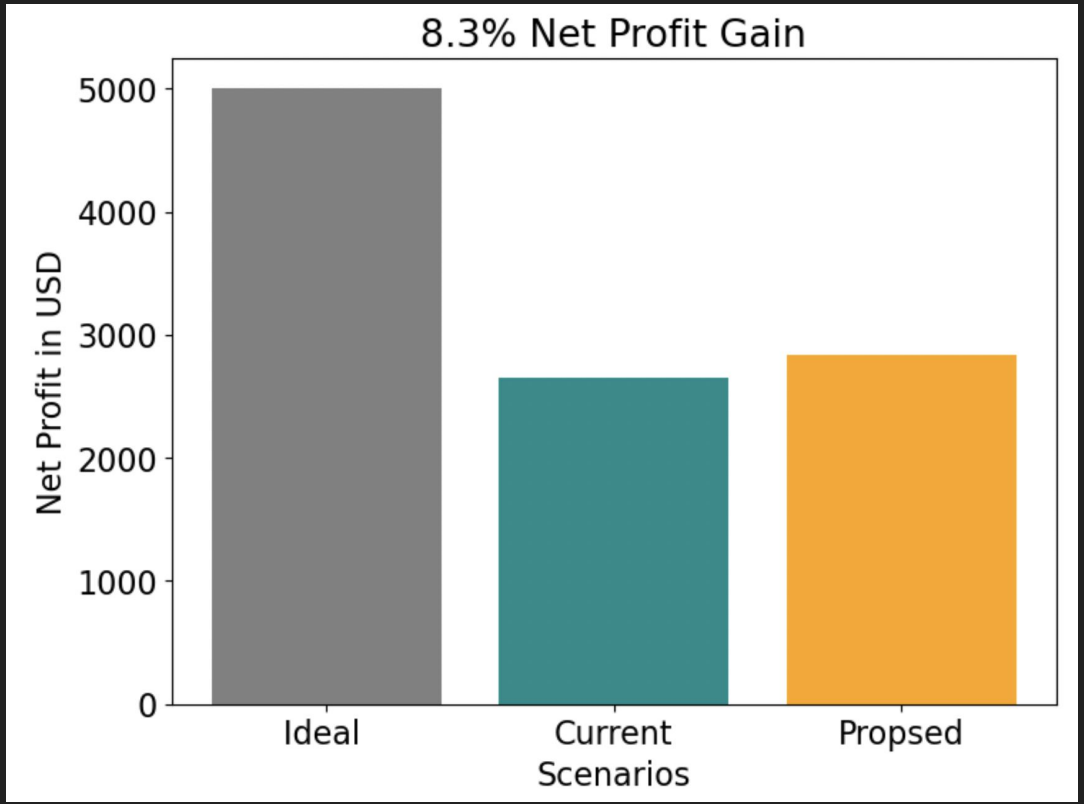
which is equivalent of 15.22 USD.

Decreased withdrawal for convenience: 5.00%

which is equivalent of 250.00 USD.



Analyzing for the Business Value



Road Map - V0 solution (1 Month)

- Improve data quality
- Increase the coverage
- Diversify the data as much as possible
- Utilize external datasets, especially localized datasets for each market
- Apply the new data data to th
- Report how the model predictions have (likely) improved with the new dataset

By the end of this milestone, an improved, more diverse and larger dataset is applied to the model.

Road Map - V1 solution (1 Quarter)

- Perform a sensitivity analysis of the model to each variable
- Identify the most important features interacting with each other
- Include interaction effects in the features
- Perform a model selection analysis such as Akaike Information Criterion
- Identify where (and possibly why) the model suffers the most
- Diagnose the model for underestimations at the higher end

By the end of this milestone, the model has been improved and an improved, more diverse and larger dataset is applied to the improved model.

Summary

- **The estimates from our crude model showed improvement over the manual inspection**
- **There is business value as net profit in adopting an ML approach**
- **Improving data quality / data quantity could lead to overall improvement**
- **Strengthening the model's weaknesses can improve the outcome**