Kaggle Home Depot Search Relevance Prediction

Alastair Hamilton

Code available on GitHub: https://github.com/AHamilton94/KAG_home-depot

Problem Statement

Accurately predict relevance scores for a search query and search result pair.

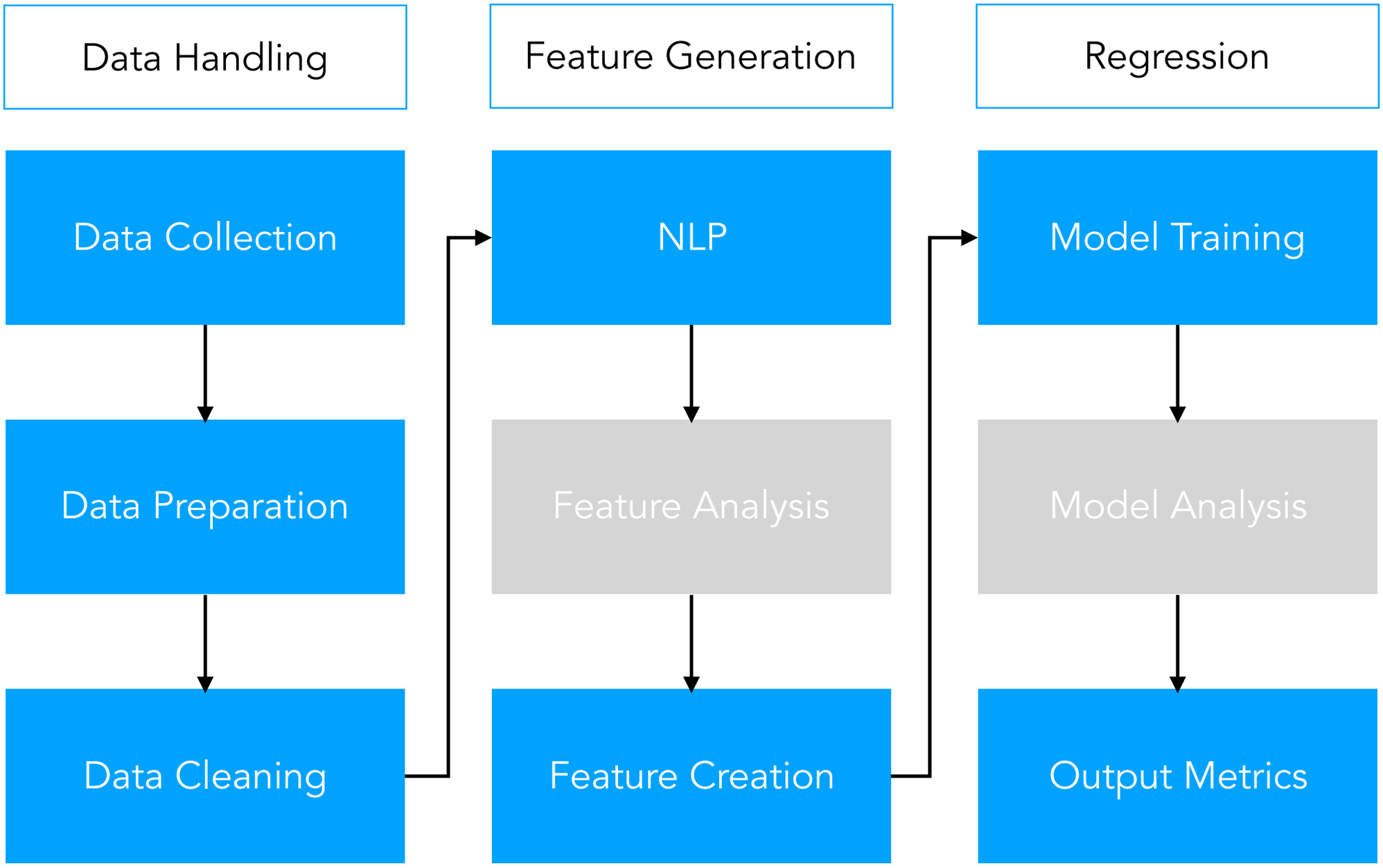
Problem Outline

Relevance calculated as a score between 1 and 3, where 1 is irrelevant and 3 is a perfect match.

Data provided:

- Search Queries
- Product IDs
- Product Titles
- Product Descriptions
- Product Attributes

Model Flow


Data Handling
Feature Generation
Regression
Data Collection
Data Preparation
Data Cleaning
NLP
Feature Analysis
Feature Creation
Model Training
Model Analysis
Output Metrics

Data Handling

Look-up relationship between UID and description.

Product Descriptions

	product_uid	product_description
0	100001	Not only do angles make joints stronger, they ...

Look-up relationship between id and other fields.

Search and Result Information

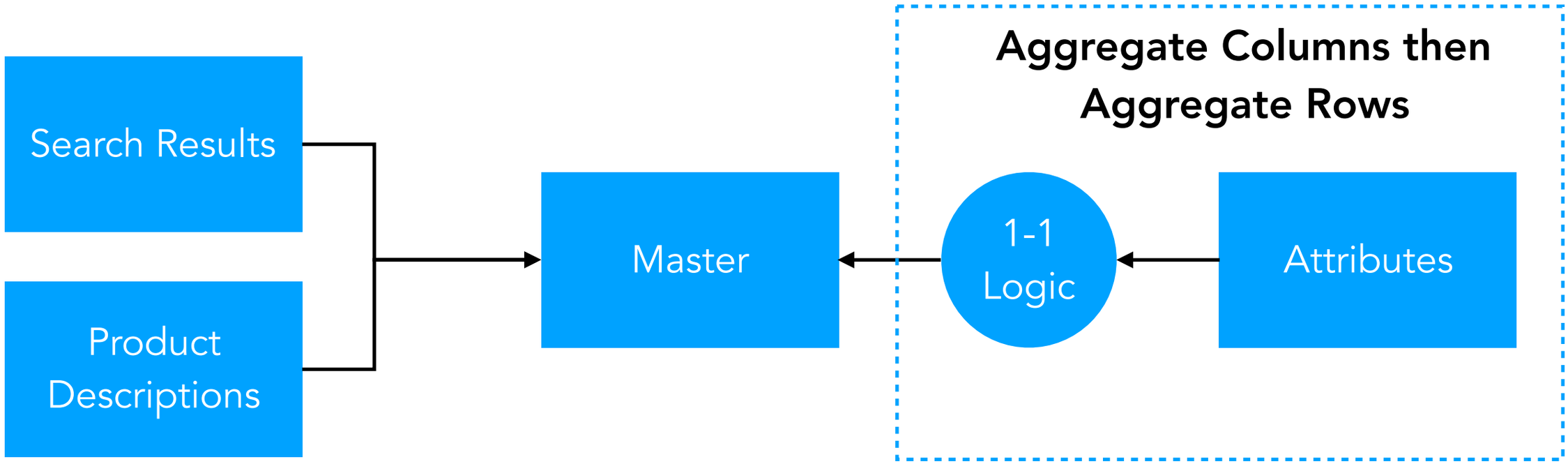
	id	product_uid	product_title	search_term	relevance
0	2	100001	Simpson Strong-Tie 12-Gauge Angle	angle bracket	3.00

One-to-many mapping between UID and name and value.

Attribute Information

	product_uid	name	value
0	100001.0	Bullet01	Versatile connector for various 90Â° connectio...

Data Handling



Master

	index	product_uid	product_title	search_term	relevance	product_description	Attributes
0	0	100001	Simpson Strong-Tie 12-Gauge Angle	angle bracket	3.00	Not only do angles make joints stronger, they ...	\tVersatile connector for various 90Â° connect...

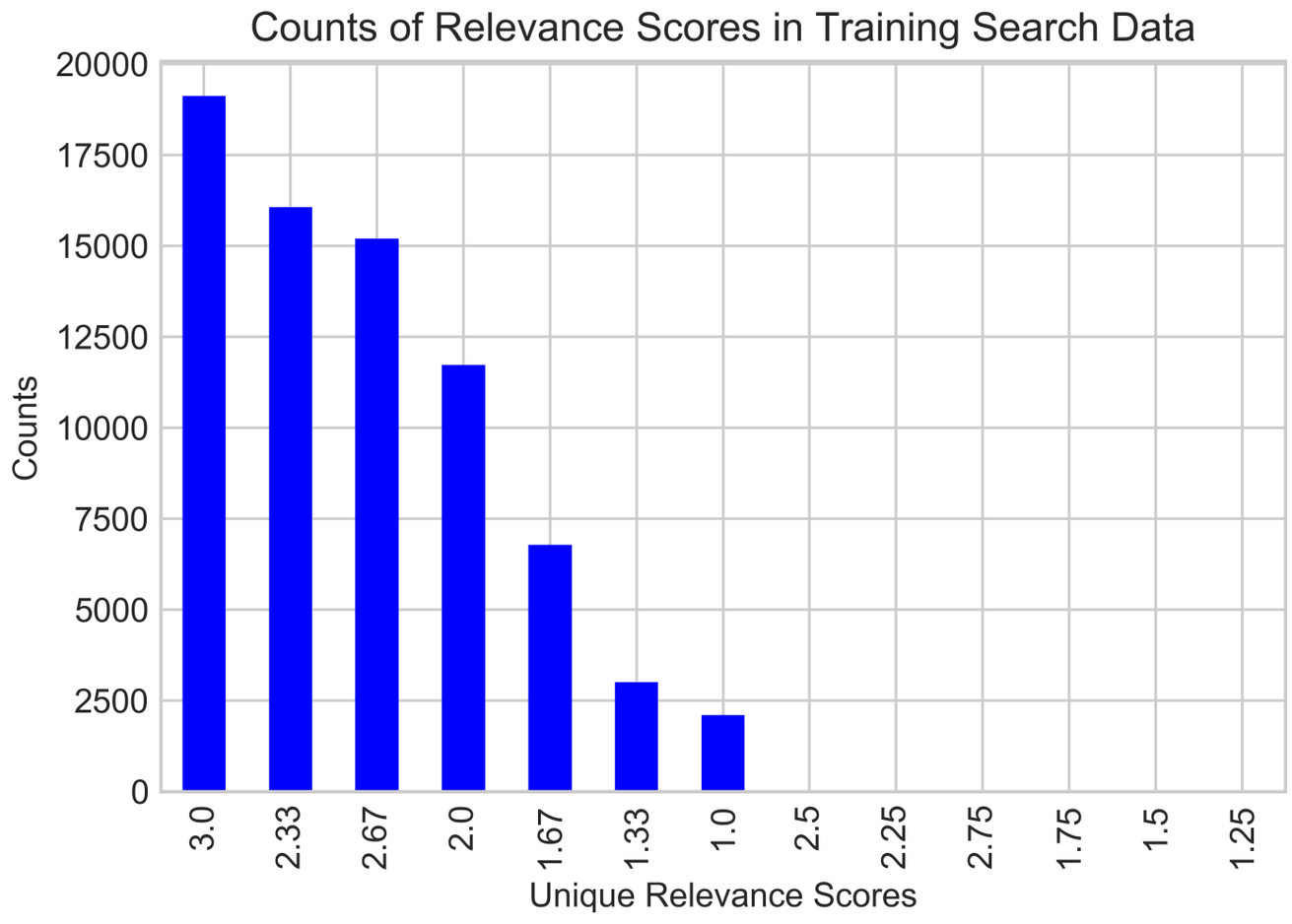
Data Handling

Data Cleaning

Null values found in attributes' data. Either removed through mapping to search data or filled as blank.

Possible outliers found in relevance data.

Poor representation of low relevances.



Feature Generation

Feature generation requires understanding of the domain.

From the relevance instructions we can understand how the relevances were scored and then which features to initially investigate.

Feature Generation

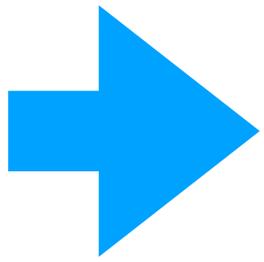
Features found using relevance instructions

Criteria for Relevance

"One of the many products searched in the query is shown"

Product metadata matching query

"The result is an item that is used with search term such as tool, accessory, extra part, cover or case"



Generated Features

Number of words in query in product title

Number of words in query in product description

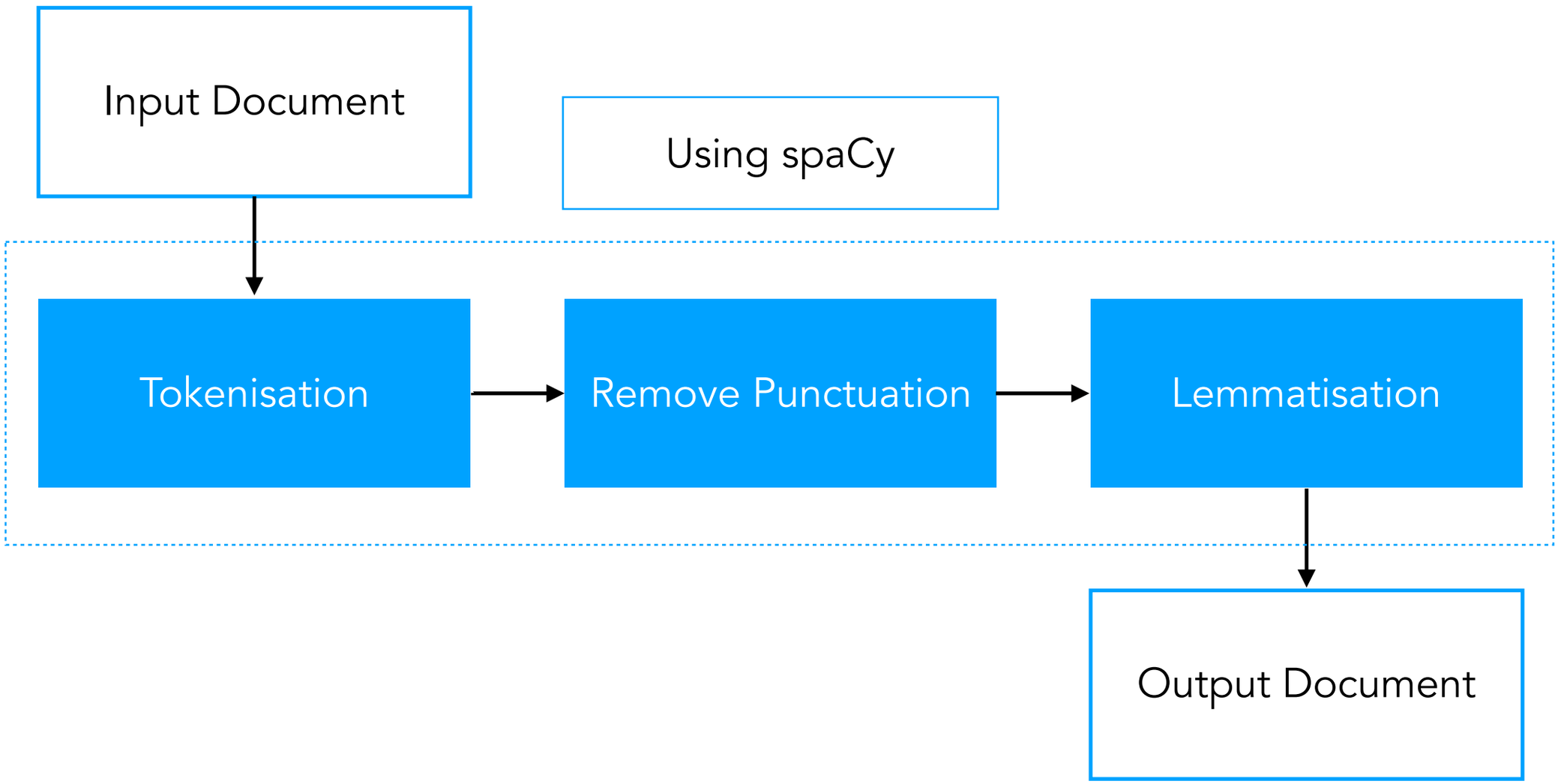
Number of words in query in product attributes

Length of query

Number of nouns in query matching partial compound nouns in product title

Feature Generation

Natural Language Processing Pipeline

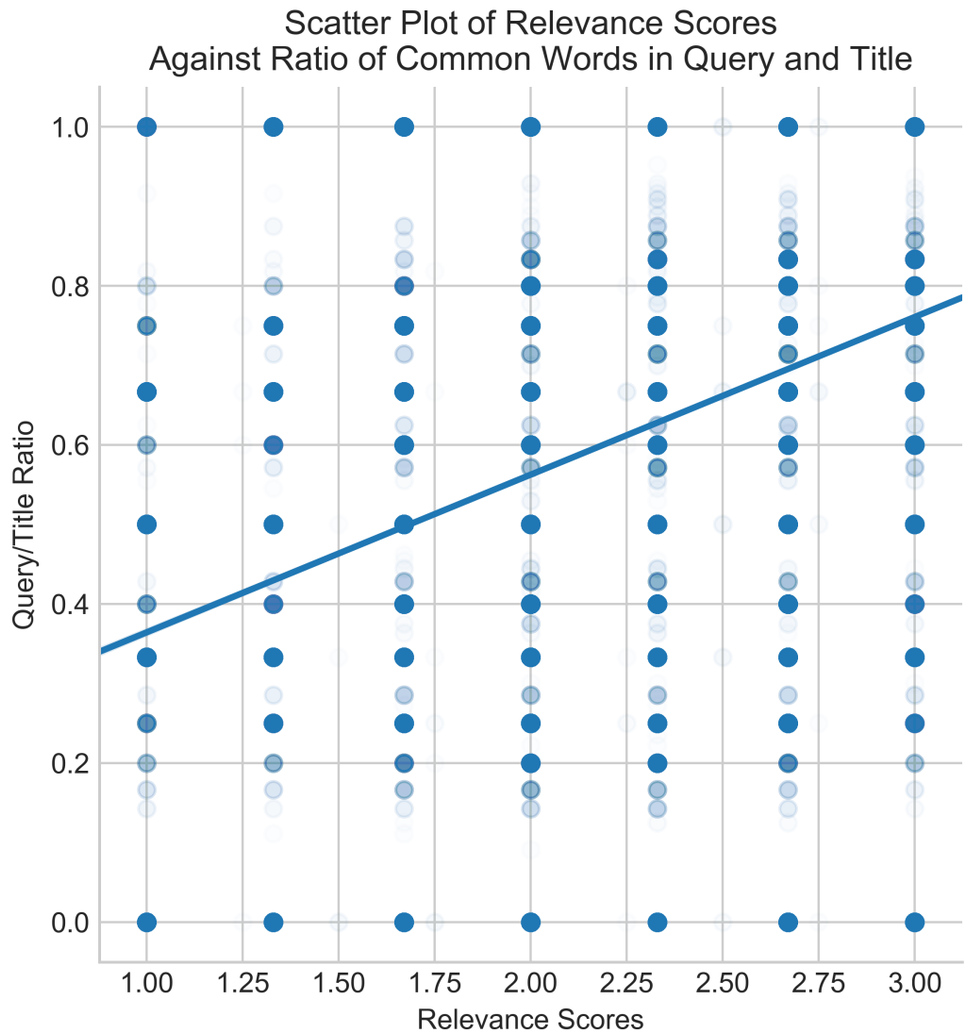


Feature Generation

Feature Analysis

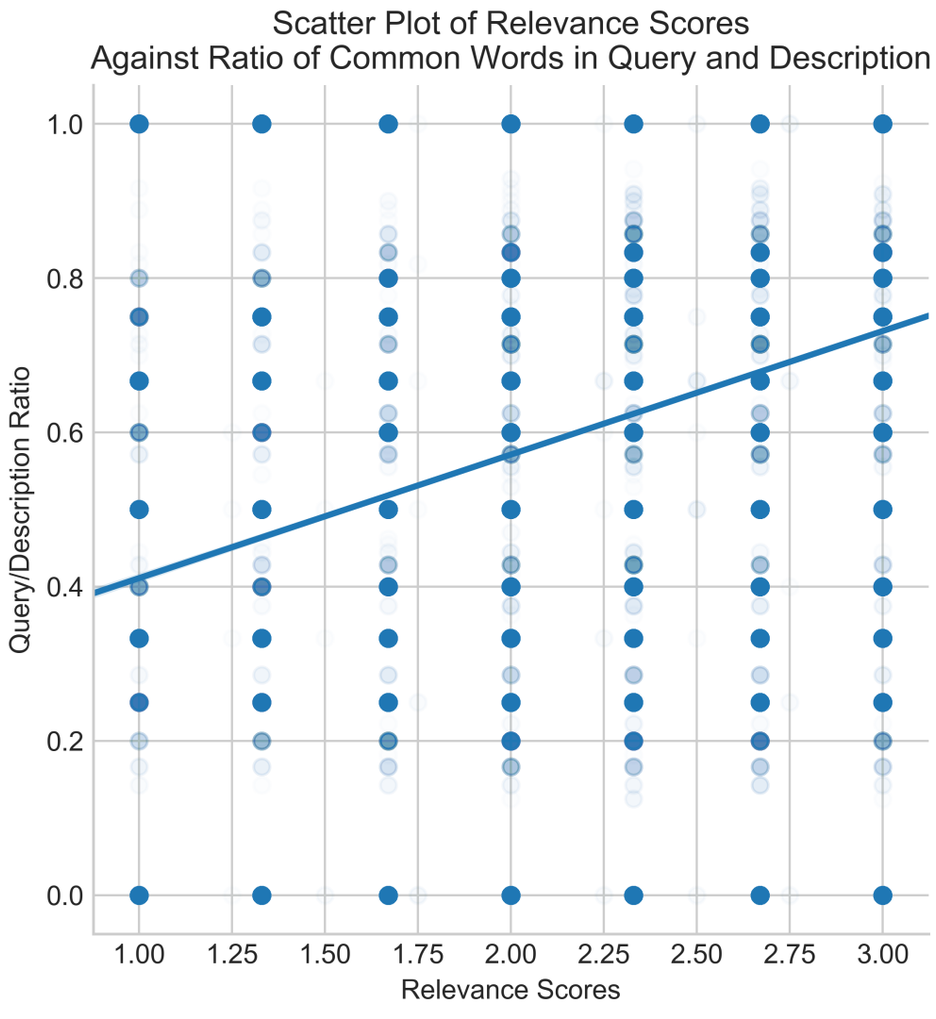
Possible correlation found for these features.

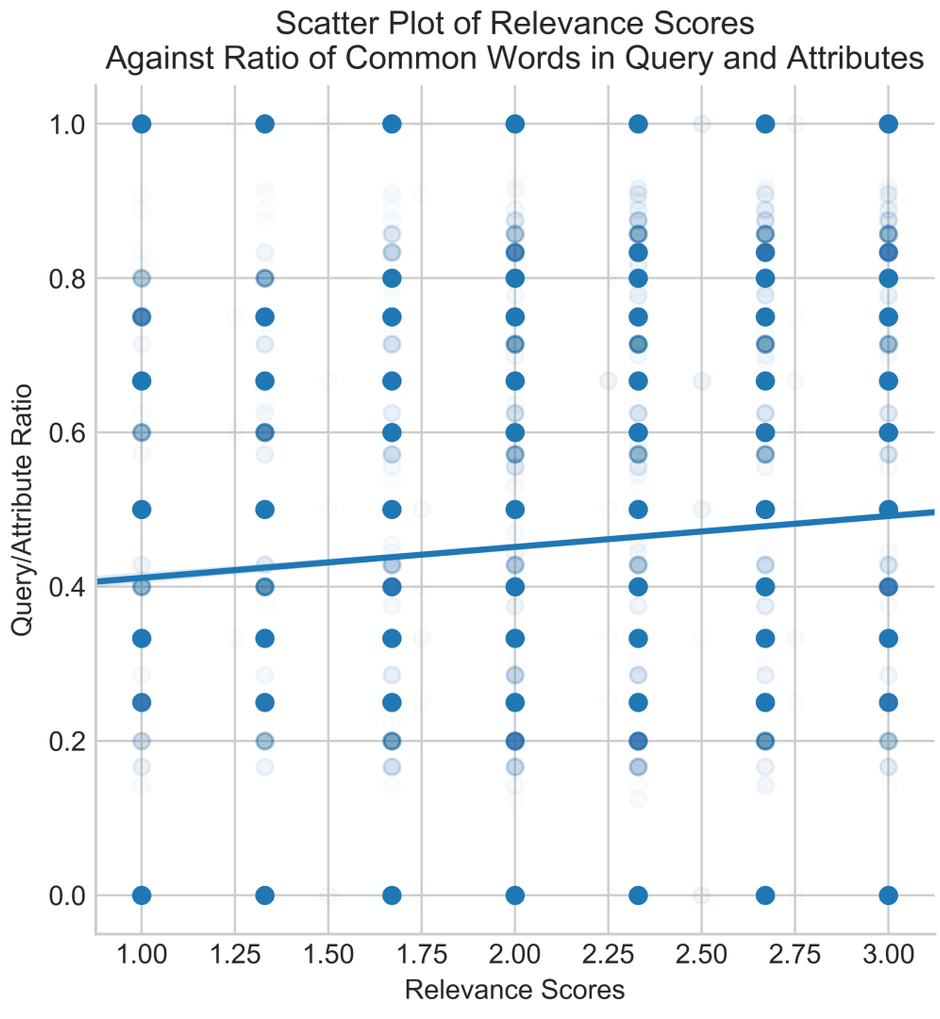
However, there is significant variation so we can’t expect these features to give a very accurate model.



Feature Generation

Feature Analysis





Regression

Machine Learning Algorithm

scikit-learn used for machine learning and statistics.

Gradient Boosted Tree and **Random Forest** were the machine learning algorithms used:

- Multiple algorithms to cross-validate model predictions
- Gradient Tree Boosting offers investigation into overfitting through the plotting of test and train deviance against boosting iterations
- For each algorithm the MSE, MAE, R^2 and residuals were output

Regression

Machine Learning Algorithm

The snippets below show the hyper-parameters used

```
# Gradient Boosted Tree Regressor Model
gbr = GradientBoostingRegressor(n_estimators:150, learning_rate: 0.4, loss: huber, alpha:0.9)
```

```
# Random Forest Model
rf = RandomForestRegressor(n_estimators=15, max_depth=6, random_state=0)
```

Regression

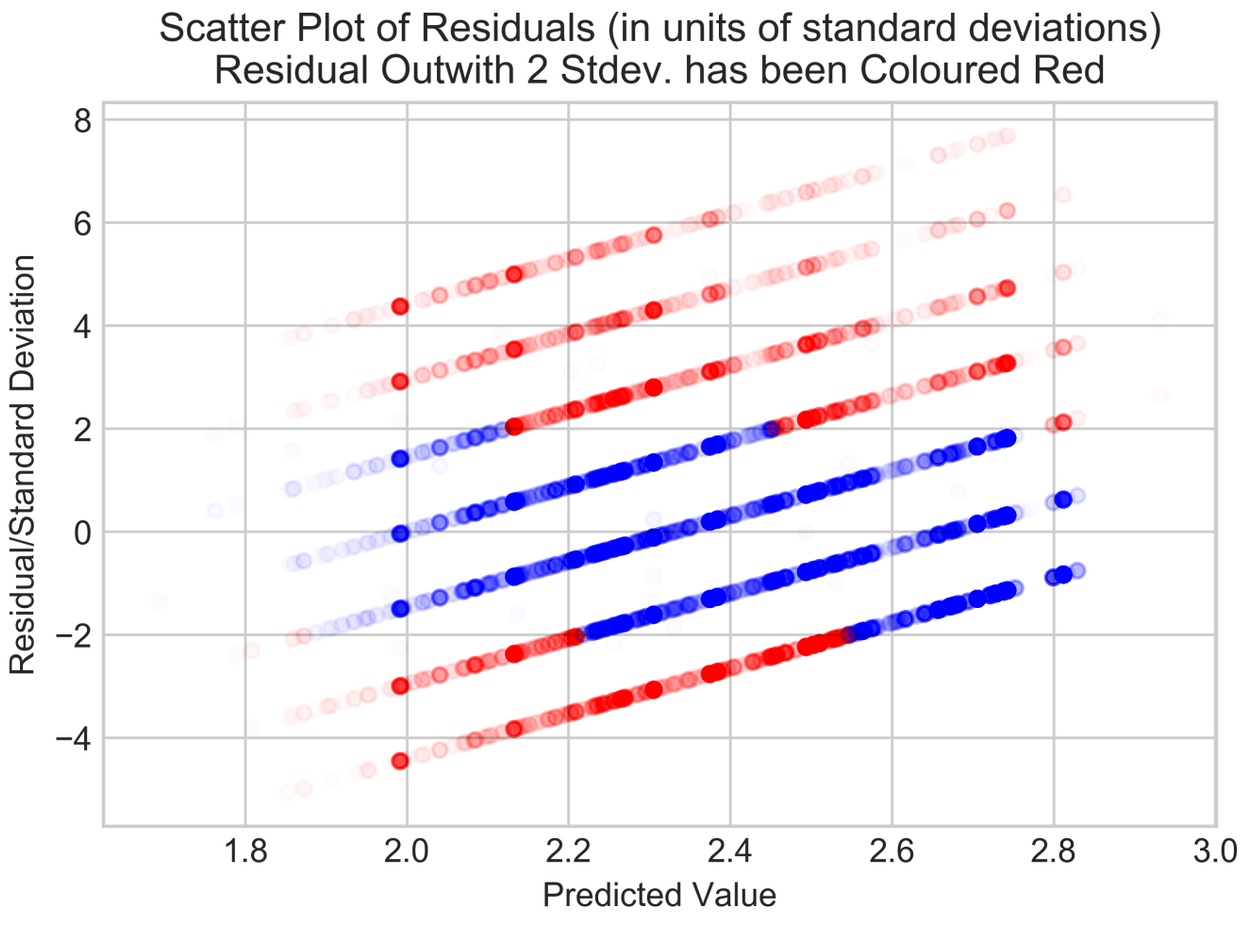
Model Analysis: Gradient Boosted Tree

MSE and MAE are good and suggest that, on average, the predictions are close to the expected values.

R^2 value and residuals (66% > 2σ) suggest that the model does not explain the error.

```
MSE: 0.2368
MAE: 0.3902
R2: 0.1684

Cross Validation
[0.16457313 0.16818818 0.17720016]
```



Regression

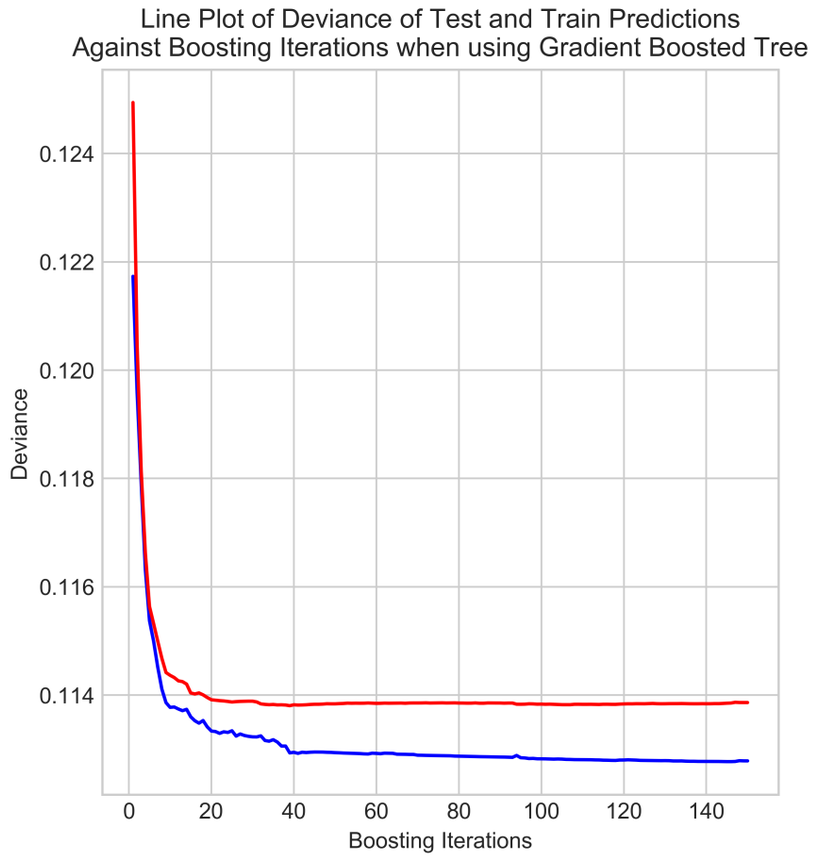
Model Analysis: Gradient Boosted Tree

Comparing the train and test variance across boosting iterations suggests some over-fitting.

The cross validation across different test and train samples suggests there is no significant bias in the test split.

```
MSE: 0.2368
MAE: 0.3902
R2: 0.1684

Cross Validation
[0.16457313 0.16818818 0.17720016]
```

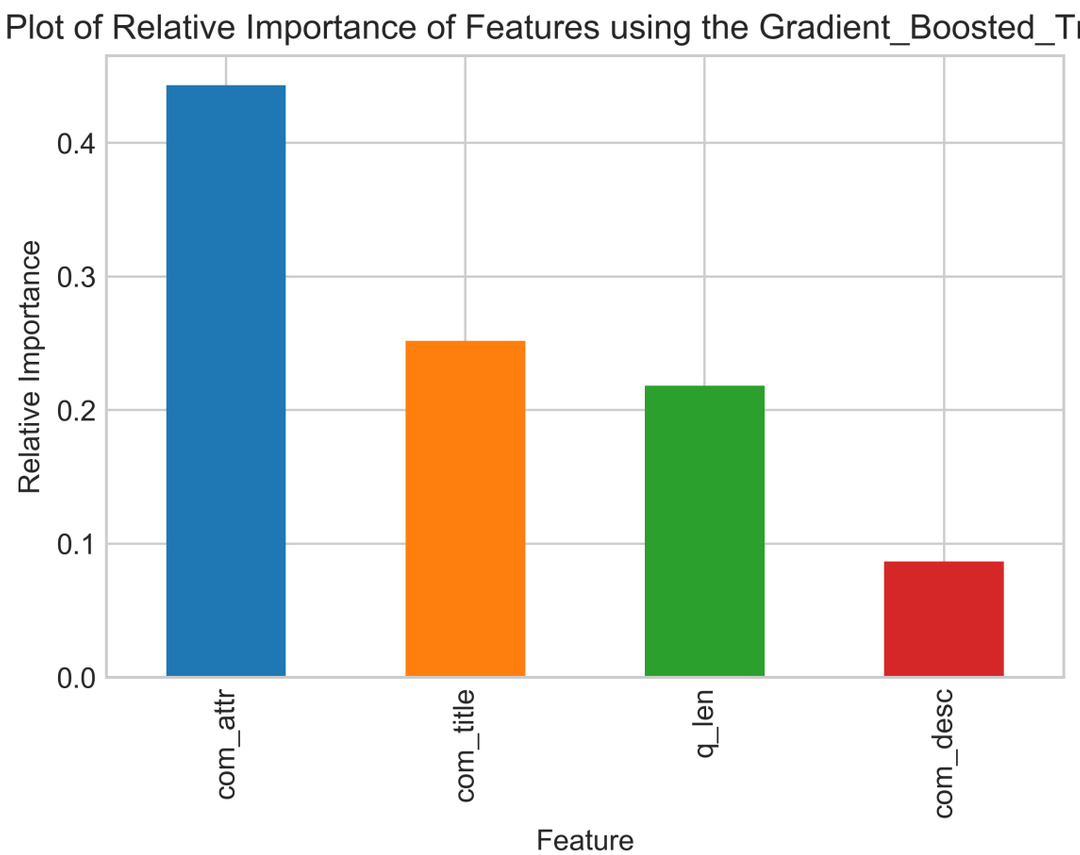


Regression

Model Analysis: Gradient Boosted Tree

Relative importances derived from this fitted model differ slightly from the original feature analysis.

Consider removing com_desc.



Regression

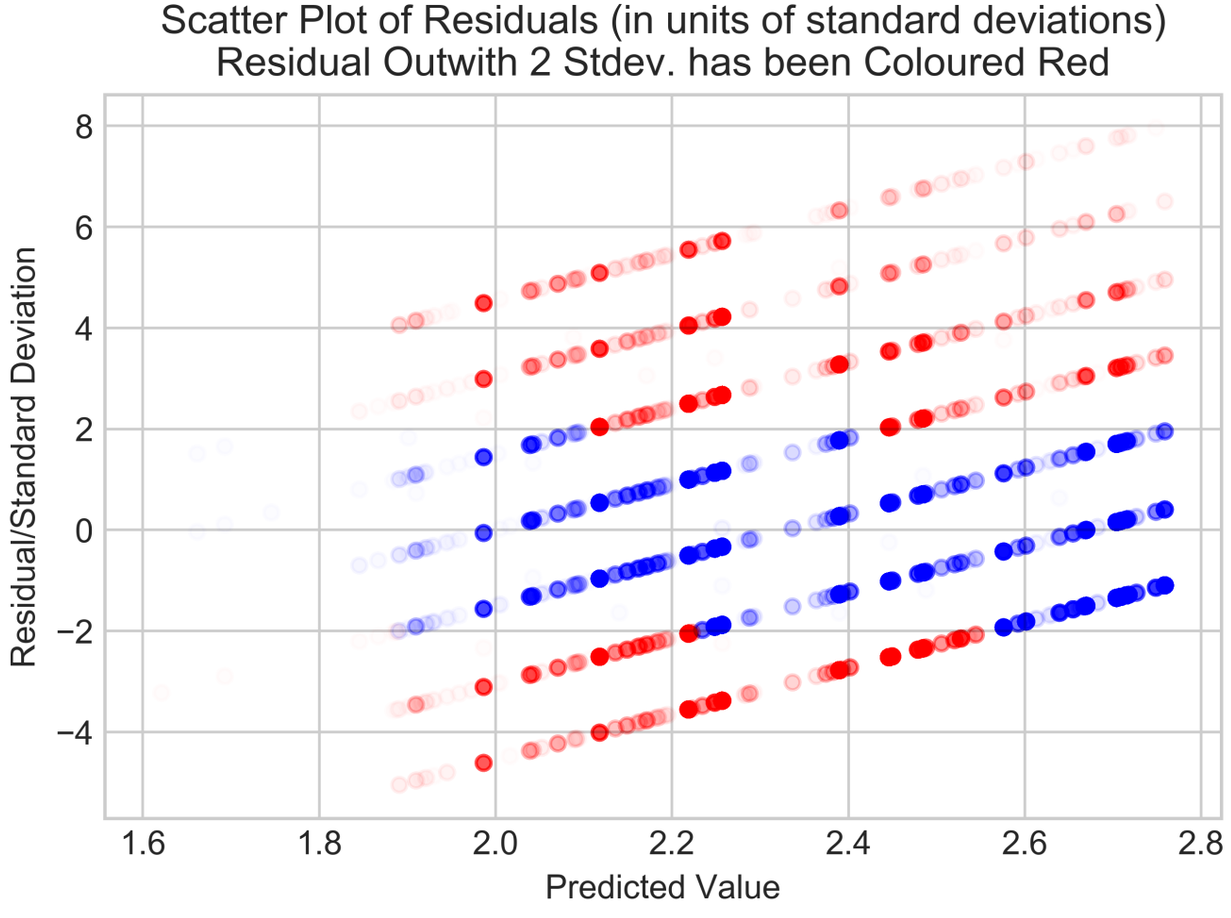
Model Analysis: Random Forest

Similar to gradient boosted tree model.

MSE and MAE are good but the R^2 and residuals (63% > 2σ) suggest more work is required in developing the features.

```
MSE: 0.2376
MAE: 0.3953
R2: 0.1657

Cross Validation
[0.1602177  0.16522195 0.17632193]
```

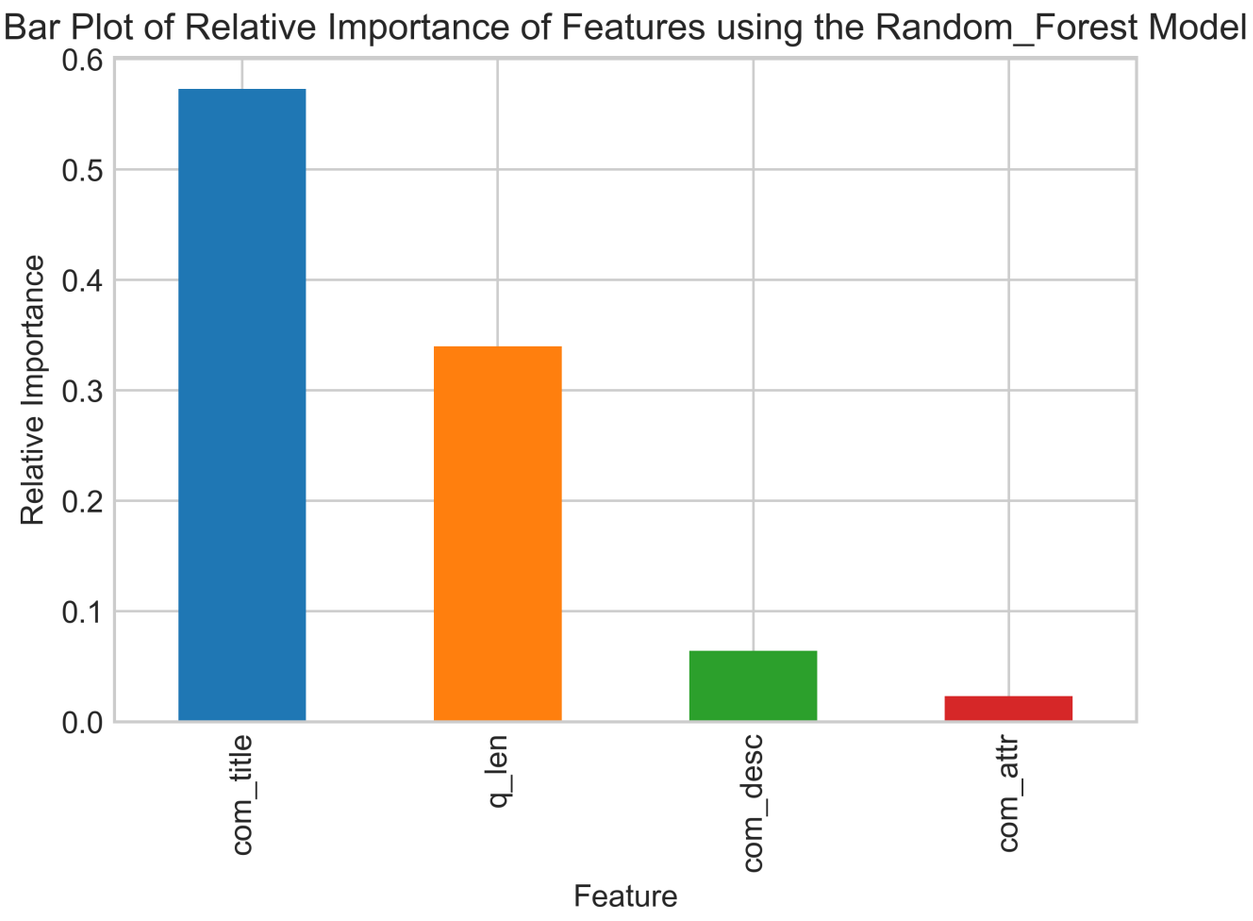


Regression

Model Analysis: Random Forest

Relative importances derived from this fitted model are very different from those derived from the gradient boosted tree model.

This substantiates the evidence in the feature analysis that the size of the union of title and query is an important feature.



Evaluation

Model was effective at proving preliminary correlation and indicating which features are most important.

Future considerations:

- Data processing was slow (~40 minutes) - is spaCy necessary?
- Rigorous feature investigation required
- Trying a classification method
- Adding more features
- Testing more models (bagging regressor, ridge regression, neural net)
- Hyper-parameter optimisation