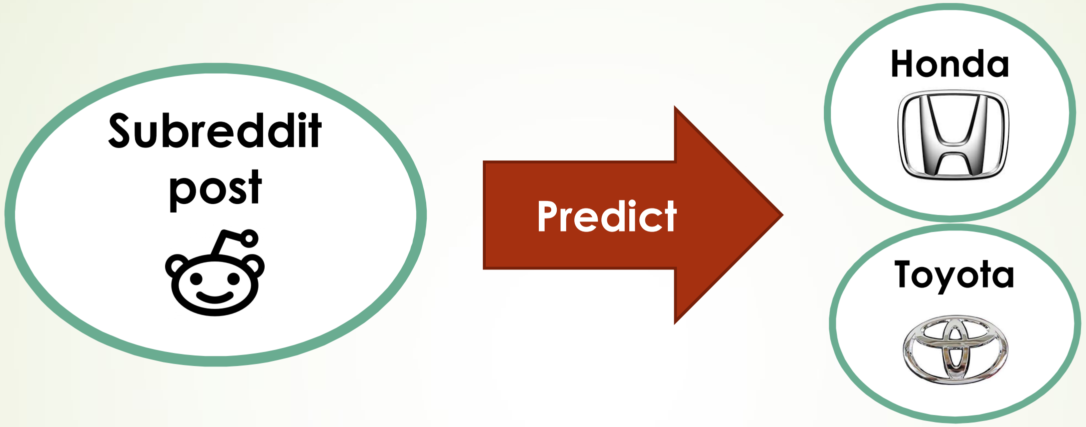
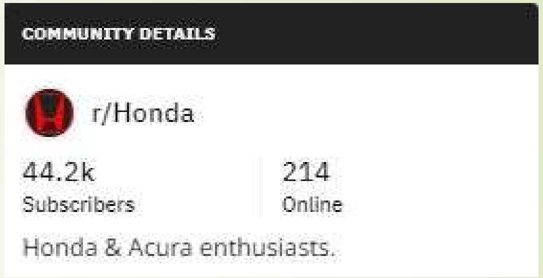
Reddit pediction model

Bernard Kurka

December 20, 2018

Problem:

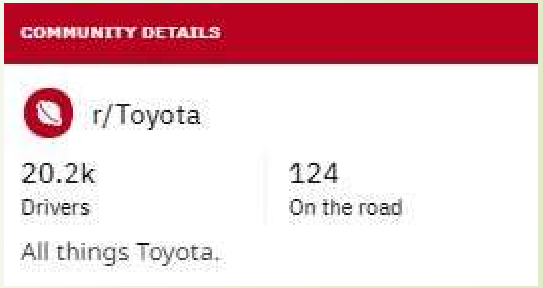



COMMUNITY DETAILS

r/Honda

44.2k Subscribers | 214 Online

Honda & Acura enthusiasts.

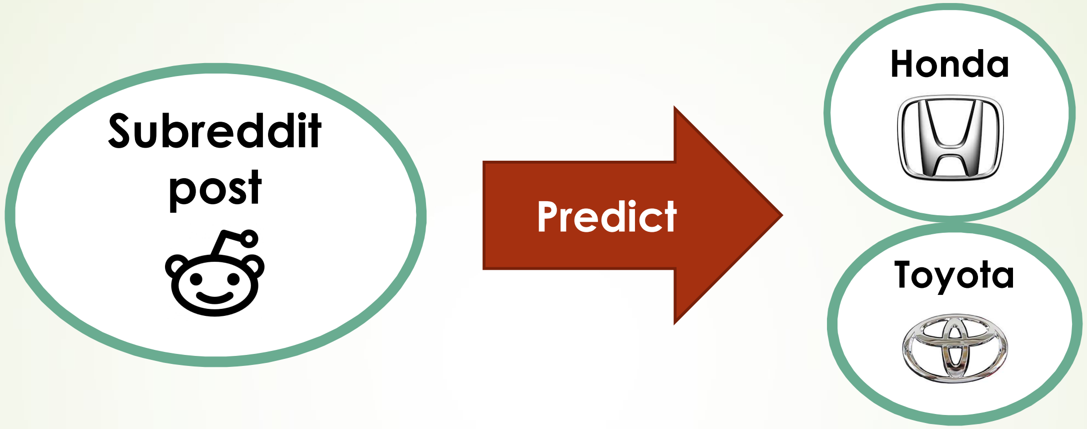

COMMUNITY DETAILS

r/Toyota

20.2k Drivers | 124 On the road

All things Toyota.

Problem:

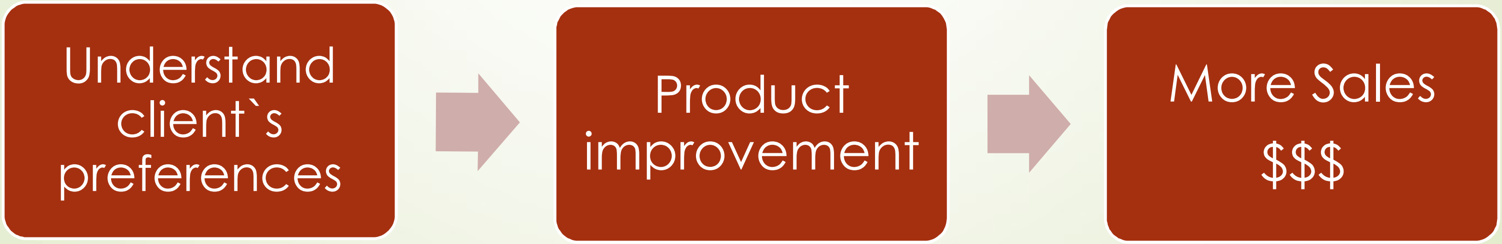

Subreddit post

Predict

Honda

Toyota

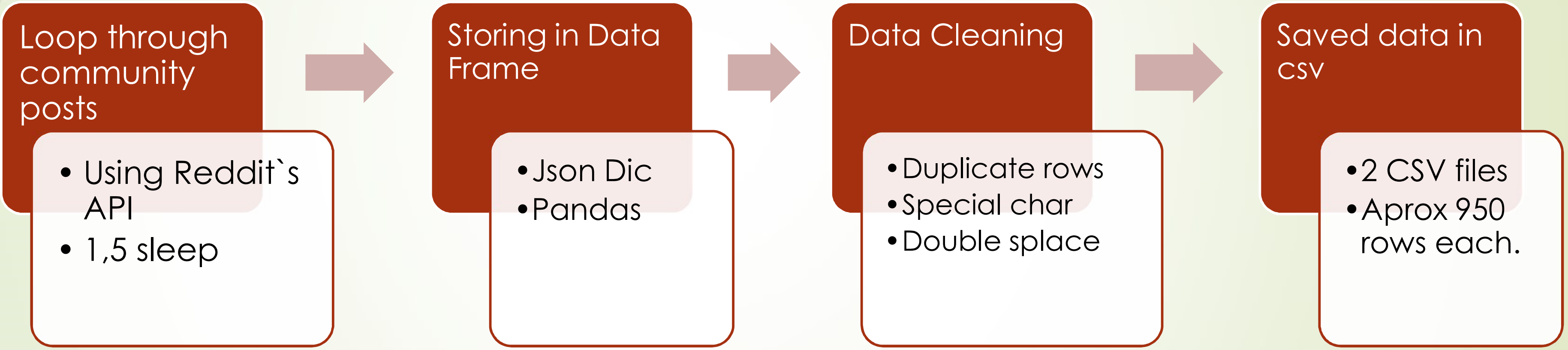
Stakeholder benefit:


Understand client`s preferences

Product improvement

More Sales $$$

Data Gathering and Cleaning:



Loop through community posts

- Using Reddit`s API
- 1,5 sleep

Storing in Data Frame

- Json Dic
- Pandas

Data Cleaning

- Duplicate rows
- Special char
- Double splace

Saved data in csv

- 2 CSV files
- Aprox 950 rows each.

Preprocessing:

Feature Engineering

- Post title
- Post body
- Post ups
- Number of comments in a post

Steam title and body words

- PorterStemmer
- LancasterStemmer
- WordNetLemmatizer

Split Train and Test Subsets

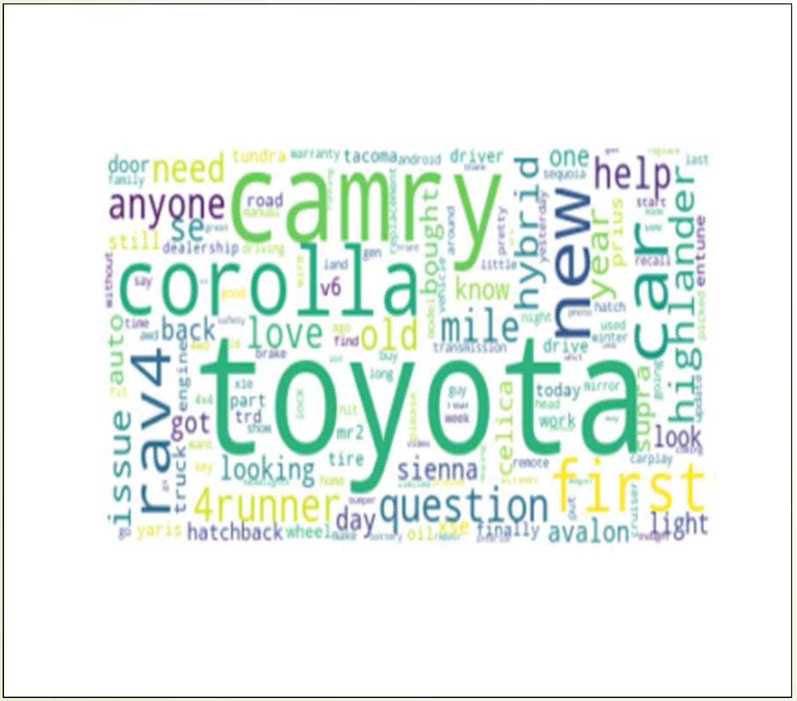
- Test subset with 25% of data.
- No need to Stratify (classes are balanced).

python™
Natural Language Analyses
with NLTK

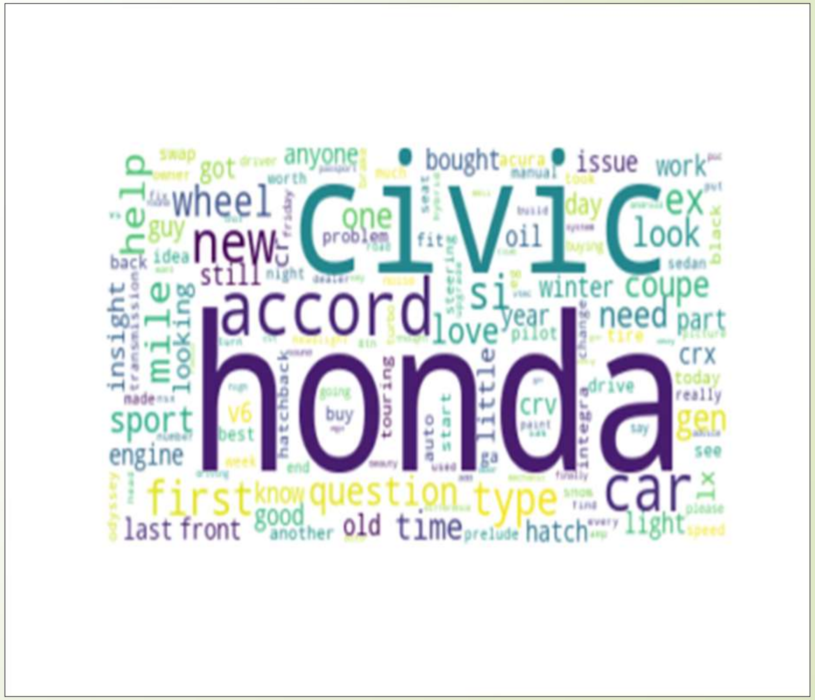
Exploring Data:

- Most frequent words:

Toyota:



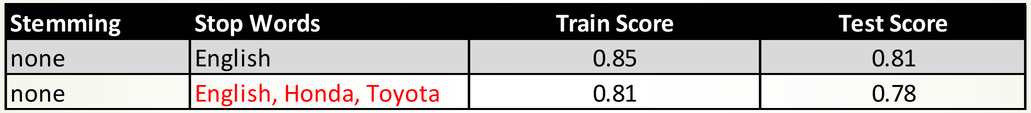
Honda:



Modeling:

- Multinomial Naive Bayes classifier (Using CountVectorizer)

Honda and Toyota Words score impact:



Stemming	Stop Words	Train Score	Test Score
none	English	0.85	0.81
none	English, Honda, Toyota	0.81	0.78

~0.04 score reduction

Modeling:

- Multinomial Naive Bayes classifier (Using CountVectorizer)

Honda and Toyota Words score impact:

Stemming	Stop Words	Train Score	Test Score
none	English	0.85	0.81
none	English, Honda, Toyota	0.81	0.78

~0.4 score reduction

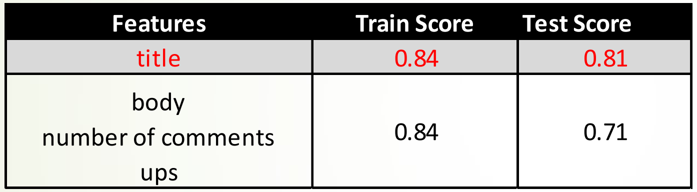
Scored the model with 4 diferent steamming:

Stemming	Train Score	Test Score
none	0.81	0.78
PorterStemmer	0.80	0.78
LancasterStemmer	0.85	0.82
WordNetLemmatizer	0.81	0.78

Modeling:

- Multinomial Naive Bayes classifier

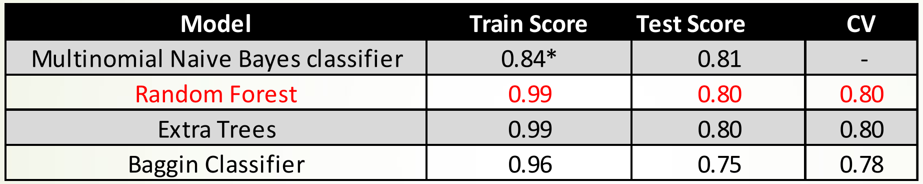
Choosing features:



Features	Train Score	Test Score
title	0.84	0.81
body number of comments ups	0.84	0.71

Modeling:

Choosing model:



Model	Train Score	Test Score	CV
Multinomial Naive Bayes classifier	0.84*	0.81	-
Random Forest	0.99	0.80	0.80
Extra Trees	0.99	0.80	0.80
Baggin Classifier	0.96	0.75	0.78

Most models overfit.

Similar scores in test and cross validation.

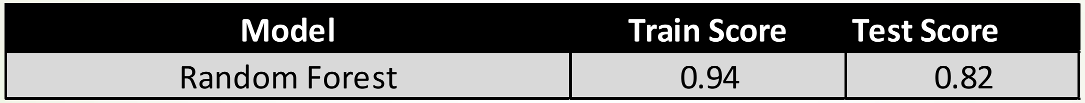
Chose Random forest because of similar test scores.

* GridSearchCV score

Modeling:

Tuning Hyper parameter:

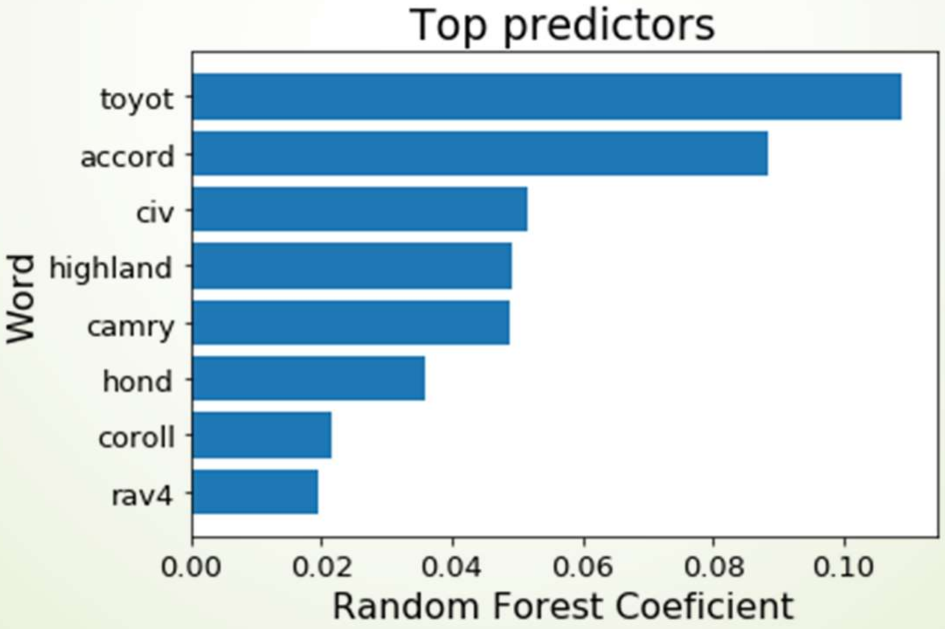
- Max_depth = 68,
- Other parameters set as default



Model	Train Score	Test Score
Random Forest	0.94	0.82

Modeling:

- Random Forest biggest feature coefficients:



Best sellers cars vs best predictors:

Toyota best sellers 2017	Coef Rank
Rav4	7
Camry	4
Corolla	6

Honda best sellers 2017	Coef Rank
CR-V	14
Civic	2
Accord	1

- 2017 best sellers are among the best predicting features.
- CR-V Honda´s best selling Honda car, it´s coefficient rank is 14.

Improvements:

Business insights:

- Further examination if there is a difference in CR-V and Accord client engagement / satisfaction.
- Discuss and evaluate if model can be used to predict 2018 best sellers.

Model improvements:

- Include ‘Hond’ and ‘Toyot’ as stop words.
- Compare Naive Base coefficientes with Random Forest.
- Run Sentiment analysis in posts and group by car name.