PHBS 北京大学汇丰商学院

Final project of Big Data

Prediction of Weibo Interaction Behaviors

Paul1801212781	1801212781
Reynard	1801213136
Yanrong Wu	1801212952
Anlei Liu	1801212887

Contents

01. Background & Topic selection

02. Descriptive Statistics

03. Feature Extraction

04. Algorithm Design

05. Implementation and Result

PART 01

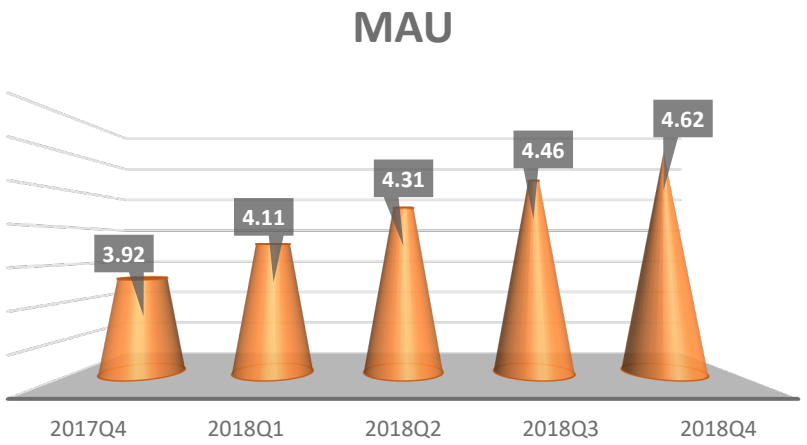
Background & Topic selection

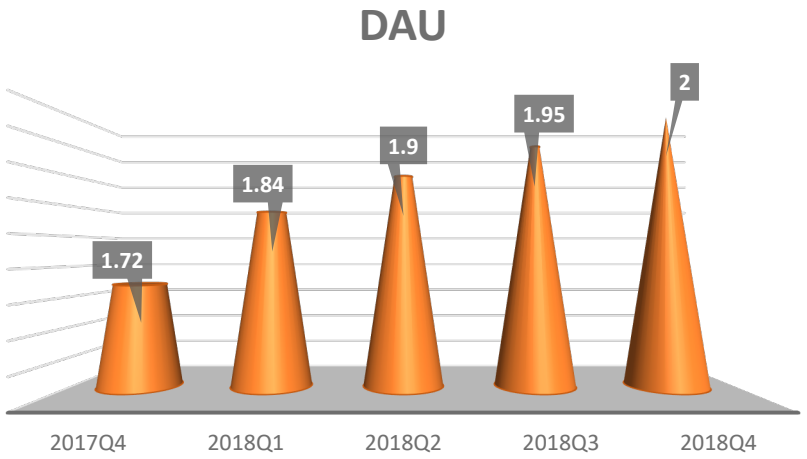
1.1 Background

Weibo has become the most popular channel for information spread.

In 2018, Weibo had a massive growth：

- **130** million texts posted per day. (+50%)
- **1.5** million vlog/live posted per day. (+50%)
- **120** million picture posted per day.(+20%)
- **50** thousands Q&A per day.(+400%)





Source: Weibo User Development Report 2018, in 10 billion

1.1 Background

As a new media form, Weibo has many special features.

Short contents and fragmented information

Dissemination of information has become very simple and convenient, and the content is easily lost or changed during the spread.

Fission spread of information

The spread of information breaks the limits of time and space, once negative rumors appear, it is completely impossible to control public opinion.

High user complexity due to low entry barriers

Weibo‘s users have large differences, which may cause serious emotional and irrational behavior, and spread of false information

1.2 Implication

Character set management

Celebrity can always follow Weibo trends to prevent the spread of false and fake news.

Weibo Marketing

Use the real-time and accurate characteristics of Weibo for precision marketing

Enterprise Promotion

The PR department can use the forecast results to control the direction of public opinion and establish a positive image of the company.

Decision Support

take Weibo as an important source of corporate intelligence, strengthen the analysis.

Government and public

Increase the communication channels between government departments and the masses by establishing official Weibo.

Public opinion early warning

strengthen the rational use of Weibo and pay more attention to Weibo's public opinion, especially during major emergencies.

1.3 Topic selection

Prediction of Weibo has huge application prospects

TIANCHI Weibo forecast competition hold by Aliyun
Forecast the repost, like and comment Weibo

46 thousand user data in 2015.
Provided a zero value baseline

1.4 Contribution and Innovation

01

Feature Extraction

- Key word: using Baidu hot words and result of Jieba
- TF-IDF: Key words' TFIDF value as a feature input
- Advertisement, weblink and other related words

02

Data Filter

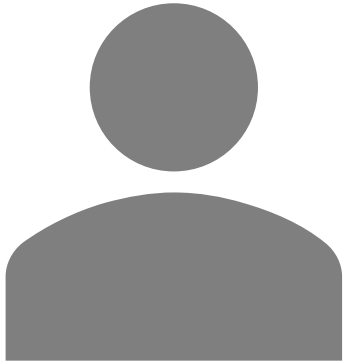
- Non-equilibrium distribution problem in dataset
- 0 value has the highest frequency
- Logistic regression filter to select 0 value

03

Combined Algorithms

- Compared Decision tree random forest and median prediction
- The combination of median prediction and random forest has highest precision

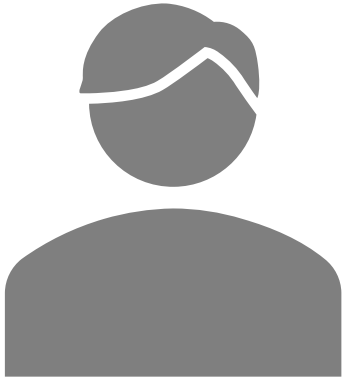
1.5 Organizational Structure and Division of Labor



Project Manager

@paul1801212781

Responsible for the overall project planning and implementation, and the core code compilation of the system, technical difficulties



Algorithm Engineer

@Reynard

Responsible for algorithm design, algorithm development and algorithm implementation and optimization, Complete technical difficulties

Architecture Engineer

@Yanrong Wu

Responsible for process design and visual display of results, and use data analysis and development tools to discover the processing of unknown information.

Operations Manager

@Anlei Liu

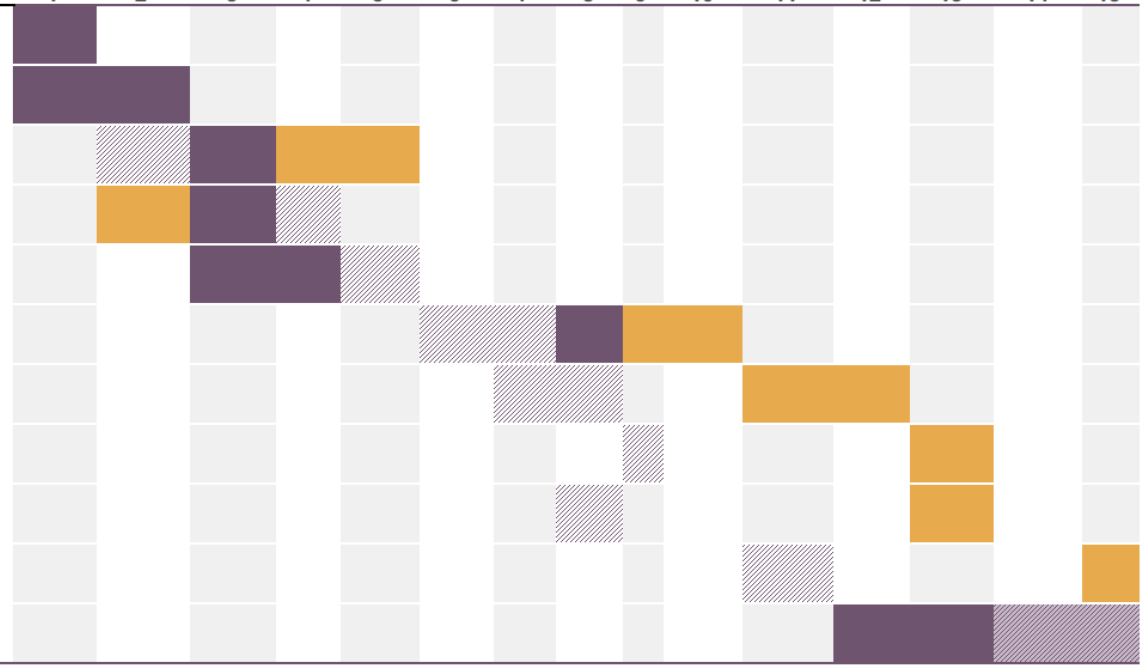
Starting from scenarios and problems, use internal and external data to identify scenarios for business data and discover and mine the value of big data.

1.5 Project Timeline

Legend

Planned Duration | Start Day (Real) | Completion Percentage | Actual (out of plan) | Completed (Over Plan)

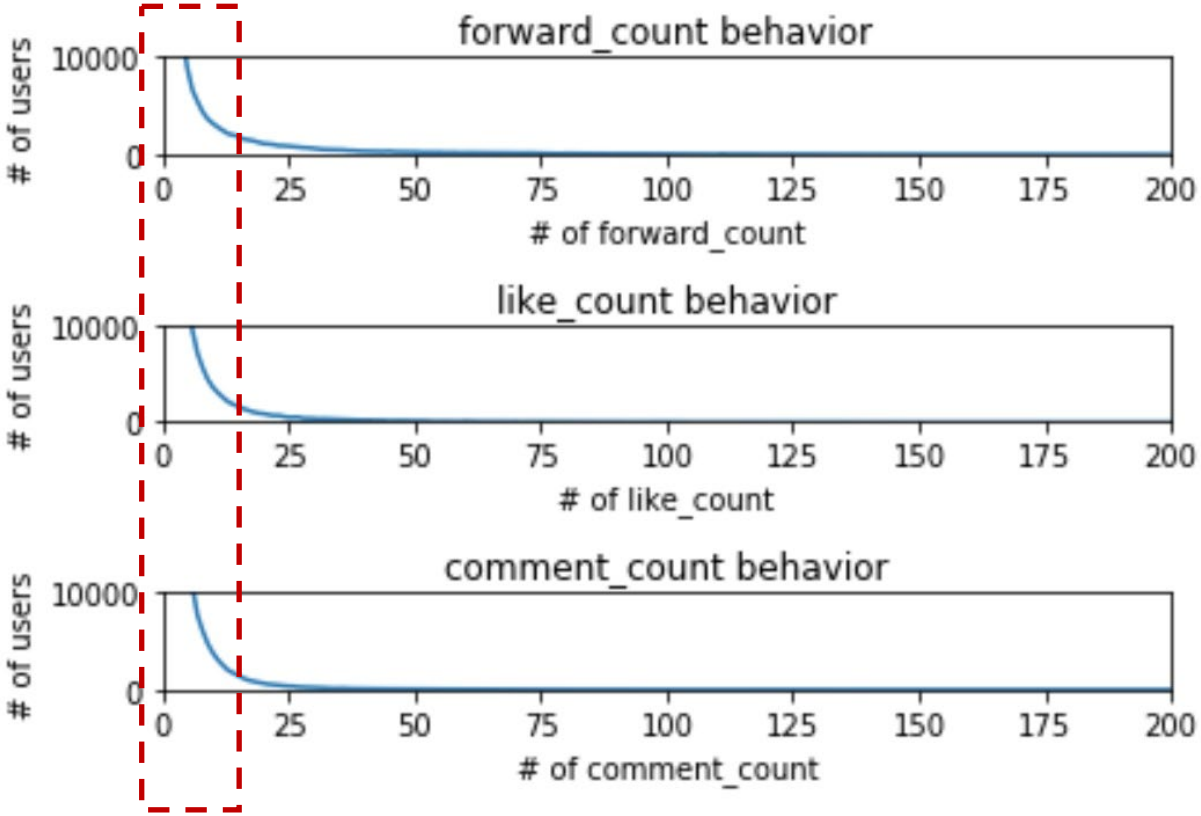
Activity	Start Day (Planning)	Duration (Planning)	Start Day (Real)	Manager	Duration (Real)	Completion percentage
Project Planning Proposal	1	1	1	Yanrong	1	**100%**
Data Collection	1	2	1	Yongqiao	2	**100%**
Feature Selection	2	2	3	Anlei	3	**100%**
Descriptive Statistics	3	2	2	Ouyang	2	**100%**
Algorithm Selection	3	3	3	Ouyang	2	**100%**
Model Building	6	3	8	Yongqiao	3	**100%**
Data Analysis	7	2	11	Yanrong	2	**100%**
Visualization	9	1	13	Ouyang	1	**100%**
Review	8	1	13	Anlei	1	**60%**
Presentation	11	1	15	Yongqiao	1	**50%**
Review and Final Report	12	4	12	Anlei	4	**50%**

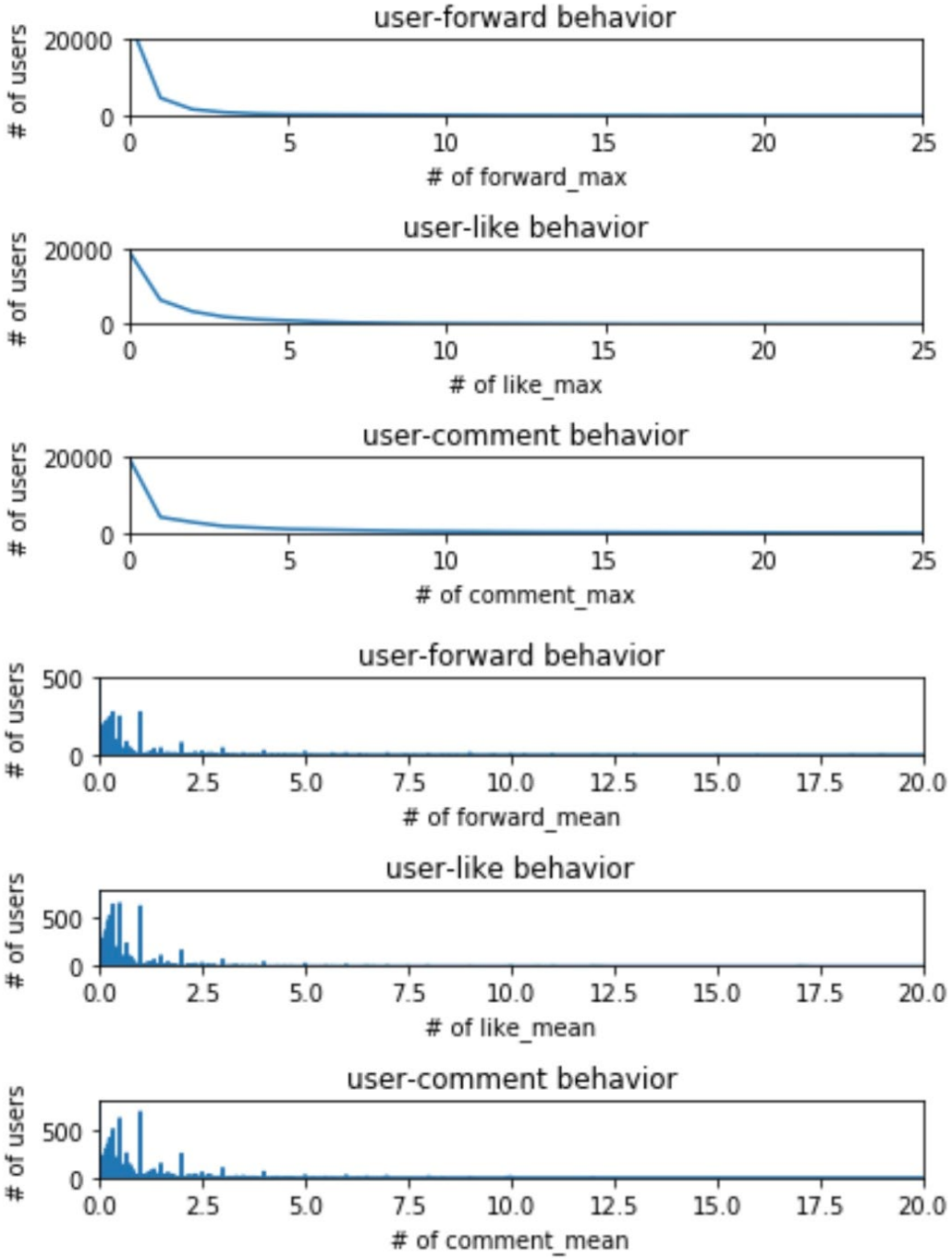


PART 02

Descriptive Statistics

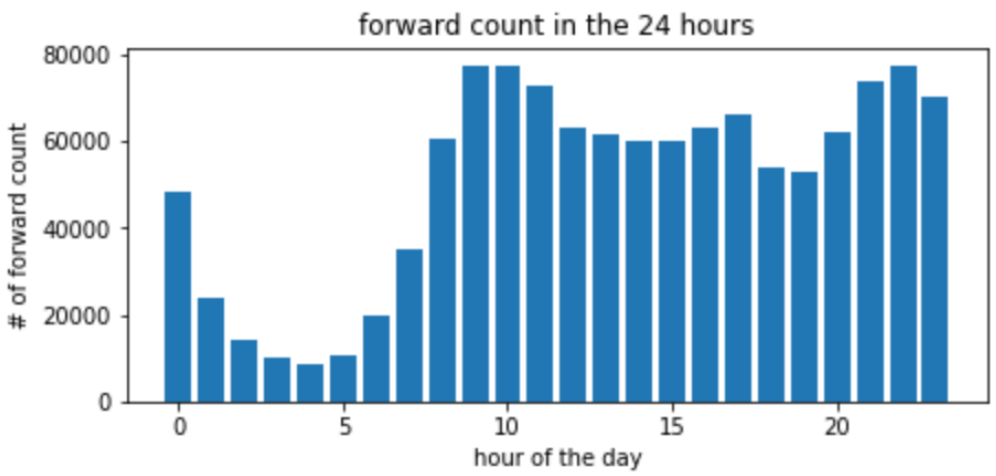
2. Descriptive Statistics

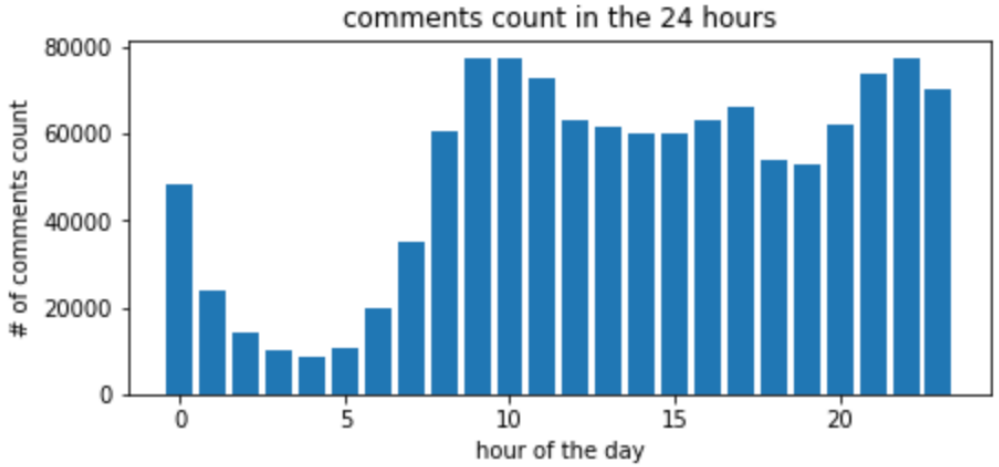




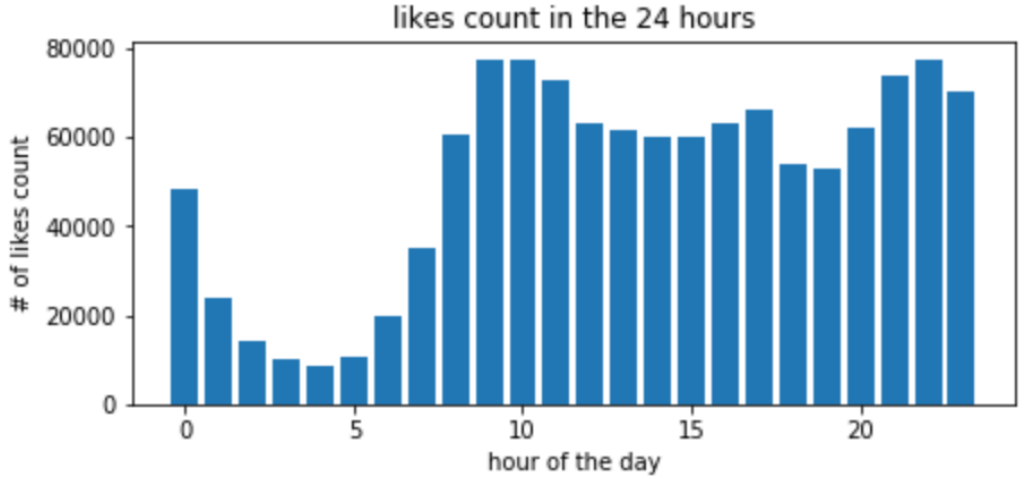
- Weibo interactions show a power-law distribution(冪律分布).
- There are a lot of 0 interactions in the dataset.
- Only a few people get a lot of interaction.

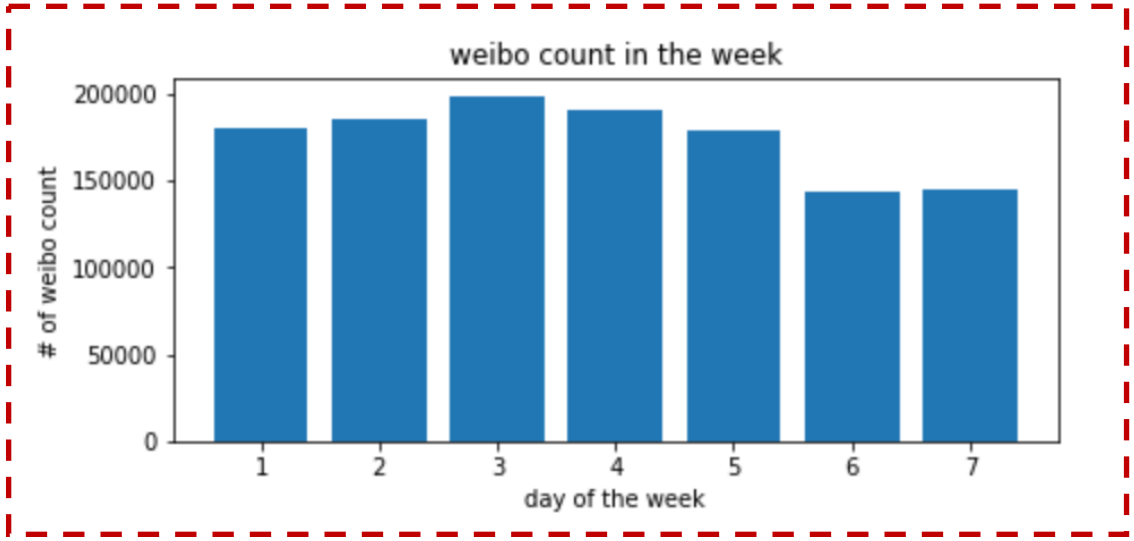
2. Descriptive Statistics





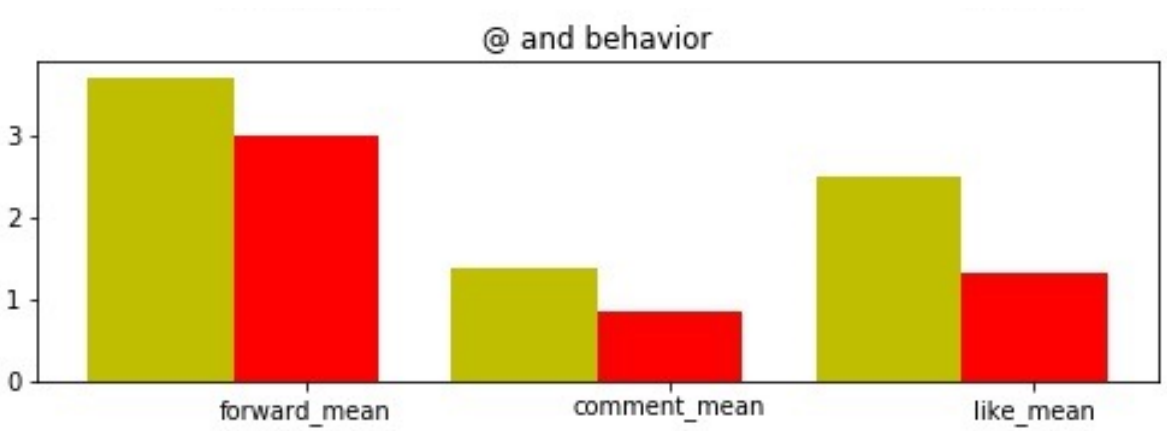
The number of Weibo interactions in the early morning time period is lower than that in other time periods. This distribution pattern can provide a reference for our subsequent feature extraction.

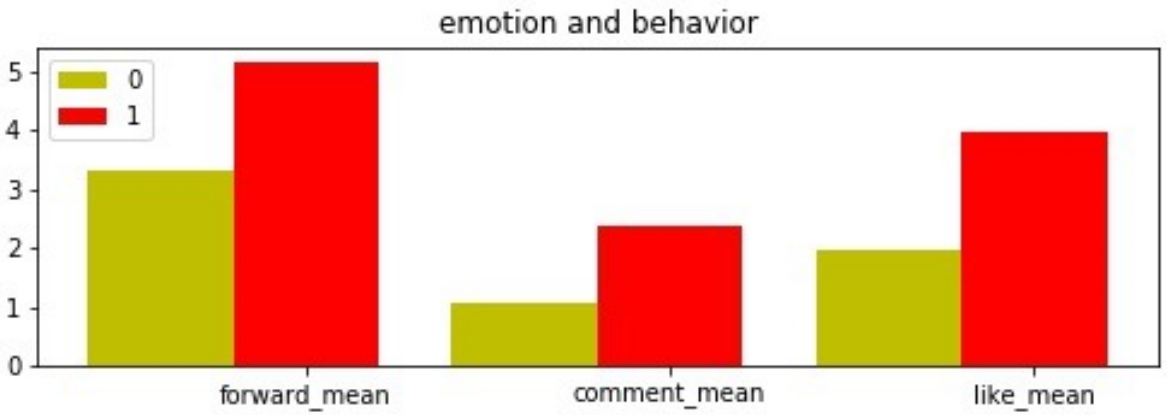


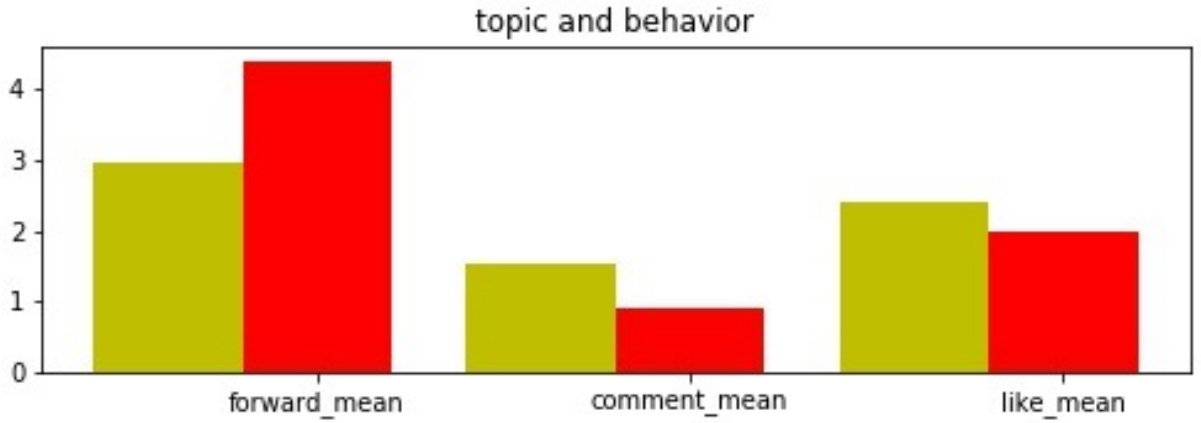


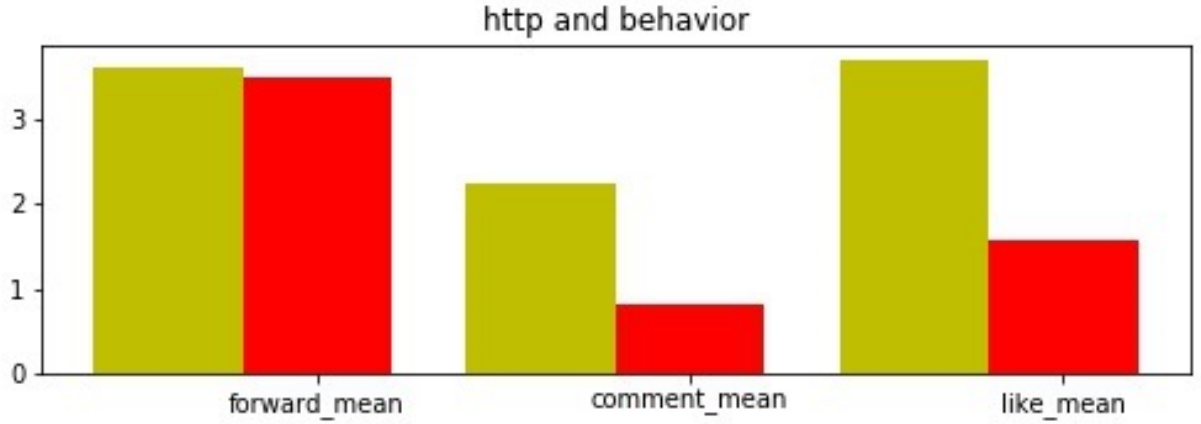
Data shows that the number of Weibo on the weekend is lower than usual.

2. Descriptive Statistics









- Text with @ will reap more interactions.
- The amount of interaction with text containing emoji is lower.

- The repost of Weibo with topics becomes lower, but like and comment are higher.
- Weibo with links has higher interactions

PART 03

Feature Extraction

3.1 Feature Extraction——User Feature

Feature	Outcome	Explanation
number_in_train	int	Find the number of times the user appears in the training set.
forward_max	int	Find the maximum of forward.
comment_max	int	Find the maximum of comment.
like_max	int	Find the maximum of like.
forward_min	int	Find the minimum of forward.
comment_min	int	Find the minimum of comment.
like_min	int	Find the minimum of comment.
forward_mean	float	Find the mean of comment.
comment_mean	float	Find the mean of comment.
like_mean	float	Find the mean of comment.
forward_judge	float	Count the number of weibo posts above than the forward average.
comment_judge	float	Count the number of weibo posts above than the comment average.
like_judge	float	Count the number of weibo posts above than the like average.

3.2 Feature Extraction——Time Feature

Feature	Outcome	Explanation
time_weekday	1,2,…,7	Determine the day of the week
time_weekend	0,1	Determine if the weibo post date is weekend
time_hour	1,…,24	Determine when the weibo post on the day
panduan	1,2,3,4	Judging Posting Period.

We segment the time of day, make 1 to 7 as 1, 7 to 12 as 2, …

```
    data['time_hour']=data['time'].apply(lambda x: datetime.datetime.strptime(x,"%Y-%m-%d %H:%M:%S").hour)
    data.loc[data.apply(lambda data:(data['time_hour']>=1)and(data['time_hour']<7), axis=1), 'panduan']=1
    data.loc[data.apply(lambda data:(data['time_hour']>=7)and(data['time_hour']<12), axis=1), 'panduan']=2
    data.loc[data.apply(lambda data:(data['time_hour']>=12)and(data['time_hour']<18), axis=1), 'panduan']=3
    data.loc[data.apply(lambda data:(data['time_hour']>=18)and(data['time_hour']<24) or (data['time_hour']==0), axis=1), 'panduan']=4
    data.drop(['time_date','time_weekend1','time_weekend2'],axis=1, inplace=True)
    return data
```

3.3 Feature Extraction——Text Feature

Feature	Outcome	Explanation
length_all	int	Weibo original length
length_chinese	int	Length of Chinese characters in the weibo
english	binary (0,1)	Whether it is English content, more than half of the words are English letters, then the Weibo is English.
non_ch	binary (0,1)	Whether it is non-Chinese content
sharing	binary (0,1)	Whether the content is sharing content
auto	binary (0,1)	Whether the text content is auto-posted in response (the text contains '我…了' and '@' or a link to the web page)
interaction	binary (0,1)	Whether the Weibo text is interactive content (whether it contains '//', but the '//' in the web link should be considered)
book	binary (0,1)	Does the text contain the title number ‘《》’
mention	binary (0,1)	Does the text contain @
vote	binary (0,1)	Does the text contain vote
lottery	binary (0,1)	Does the text contain lottery
emoji	binary (0,1)	Does the text contain emoji
video	binary (0,1)	Does the text contain video

3.3 Feature Extraction——Text Feature (Con'd)

Feature	Outcome	Explanation
http	binary (0,1)	Does the text contain link.
stock	binary (0,1)	Whether the content is a stock tweet.
app	binary (0,1)	Is there a third-party platform interactive message in the text ("我在#xxx").
title	binary (0,1)	Does the text contain 【】or title (most likely news).
ad	binary (0,1)	Does the text contain advertise.
hotwords	binary (0,1)	Does the text contain baidu hot words. ['2015阅兵', '奔跑吧兄弟', '花千骨', 'duang', 'DUANG', '毕福剑', '完美世界', '清华大学', '九寨沟', '天津爆炸', '快乐大本营', \ '校花的贴身高手', '车震', '金星', '大主宰', '武汉大学', '泰山', '全面开放二孩政策', 'running man','Running Man','Running man',\...
keywords	binary (0,1)	Jieba word segmentation; Find high-frequency hot words, and see if each Weibo contains high-frequency hot words.

We introduce the baidu hotwords in 2015 to determine whether the hotwords will appear in the weibo. (because we can get the history hot search list of weibo)

PART 04

Algorithm Design

Road Map

Descriptive Analysis

1

User Behavior

Time Pattern

Text Distribution

Feature Extraction

2

User Profile:

Historical Range

Post Frequency

Time Feature:

Weekdays or Weekends

Time Period

Text Feature:

Content Classification

Pre-process Filter

3

Noise Identification:

Non-Chinese

Short Length

Auto Generated

Vote

Advertisement

Logistic Regression:

Non-interaction prediction

Predictive Model

4

Baseline Model:

Zero Prediction

Available Model:

Linear Regression

Decision Tree

Random Forest

General Model

Uneven Distribution

Notation: data_train: Feb to Jun, data_valid: July, data_test: Aug

symptom: 64.67% of sample has zero interaction (forward + comment + like).

Strategy: filter sample to get rid of irregular text

Noise identification:

Feature	Interaction	Mean	Std.Dev	Max
Non-Chinese	Repost	0.48	5.28	556
	Comments	0.41	6.04	601
	Likes	0.94	21.57	1787
Length<5	Repost	0.91	45.71	7240
	Comments	1.05	10.17	909
	Likes	1.66	27.19	1840
Auto Generated	Repost	1.3	45.2	9431
	Comments	0.44	7.79	1244
	Likes	0.7	22.09	6306
Vote	Repost	0.48	5.49	303
	Comments	0.27	3.15	183
	Likes	0.42	2.87	90
Advertisement	Repost	1.31	73.71	12436
	Comments	0.45	27.62	7467
	Likes	0.5	39.65	10815

Logistic Regression as Filter

Dependent Variable: is_zero = 1 if no interaction else 0

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.3129	0.012	26.403	0.000	0.290	0.336
tfidf	0.0306	0.002	17.001	0.000	0.027	0.034
number_in_train	0.0001	1.08e-06	94.046	0.000	9.93e-05	0.000
forward_max	-9.147e-05	1.46e-05	-6.258	0.000	-0.000	-6.28e-05
forward_mean	-0.1979	0.001	-160.760	0.000	-0.200	-0.195
time_weekend	-0.0506	0.005	-9.381	0.000	-0.061	-0.040
length_all	-0.0013	6.19e-05	-21.292	0.000	-0.001	-0.001
sharing	0.0606	0.008	7.659	0.000	0.045	0.076
book	0.1315	0.010	13.786	0.000	0.113	0.150
mention	0.2222	0.006	34.941	0.000	0.210	0.235
emoji	-0.3634	0.007	-54.116	0.000	-0.377	-0.350
video	0.1764	0.031	5.744	0.000	0.116	0.237
http	0.9429	0.006	171.334	0.000	0.932	0.954
title	-0.3152	0.009	-36.352	0.000	-0.332	-0.298
hotwords	-0.0654	0.011	-5.923	0.000	-0.087	-0.044
keywords	-0.1553	0.010	-15.543	0.000	-0.175	-0.136
stock	0.9832	0.040	24.323	0.000	0.904	1.062
is_noise	0.5013	0.006	78.701	0.000	0.489	0.514
night	0.1419	0.013	10.819	0.000	0.116	0.168
lottery	1.5636	0.019	81.148	0.000	1.526	1.601

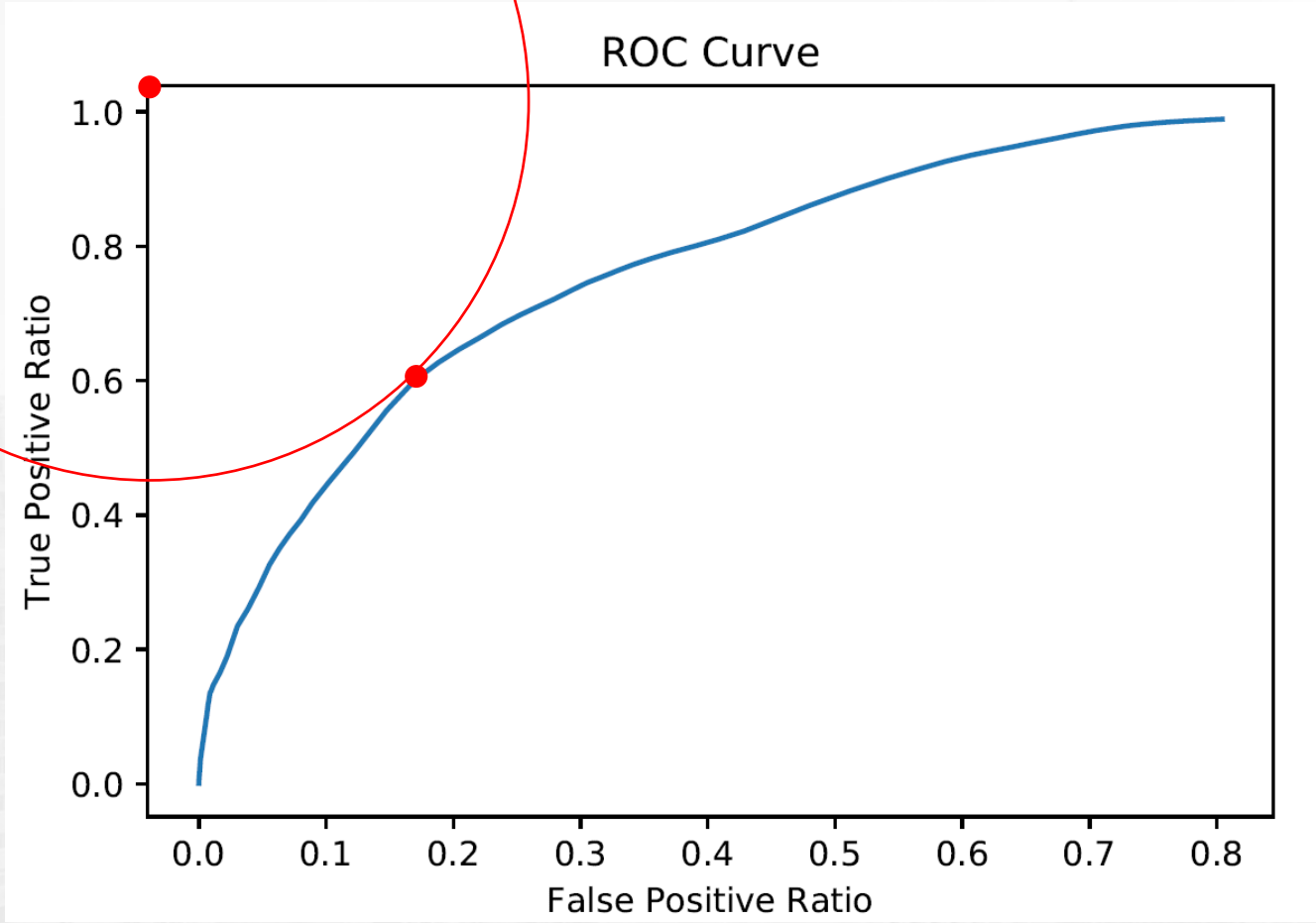
Threshold=0.75

	Estimated Value	Actual Value	Predicted Value
0	0.775459	1	1
1	0.738916	1	0
2	0.756537	1	1
3	0.775459	1	1
4	0.775459	1	1
5	0.738916	1	0
6	0.757228	1	1
7	0.763188	1	1
8	0.749281	1	0
9	0.779484	1	1
10	0.748341	1	0
11	0.738916	1	0
12	0.763188	1	1
13	0.761382	1	1
14	0.775459	1	1
15	0.748307	1	0

ROC Curves and Optimal Threshold

True Positive Ratio: ratio of actual value=1 and predicted value=1

False Positive Ratio: ratio of actual value=0 and predicted value=1

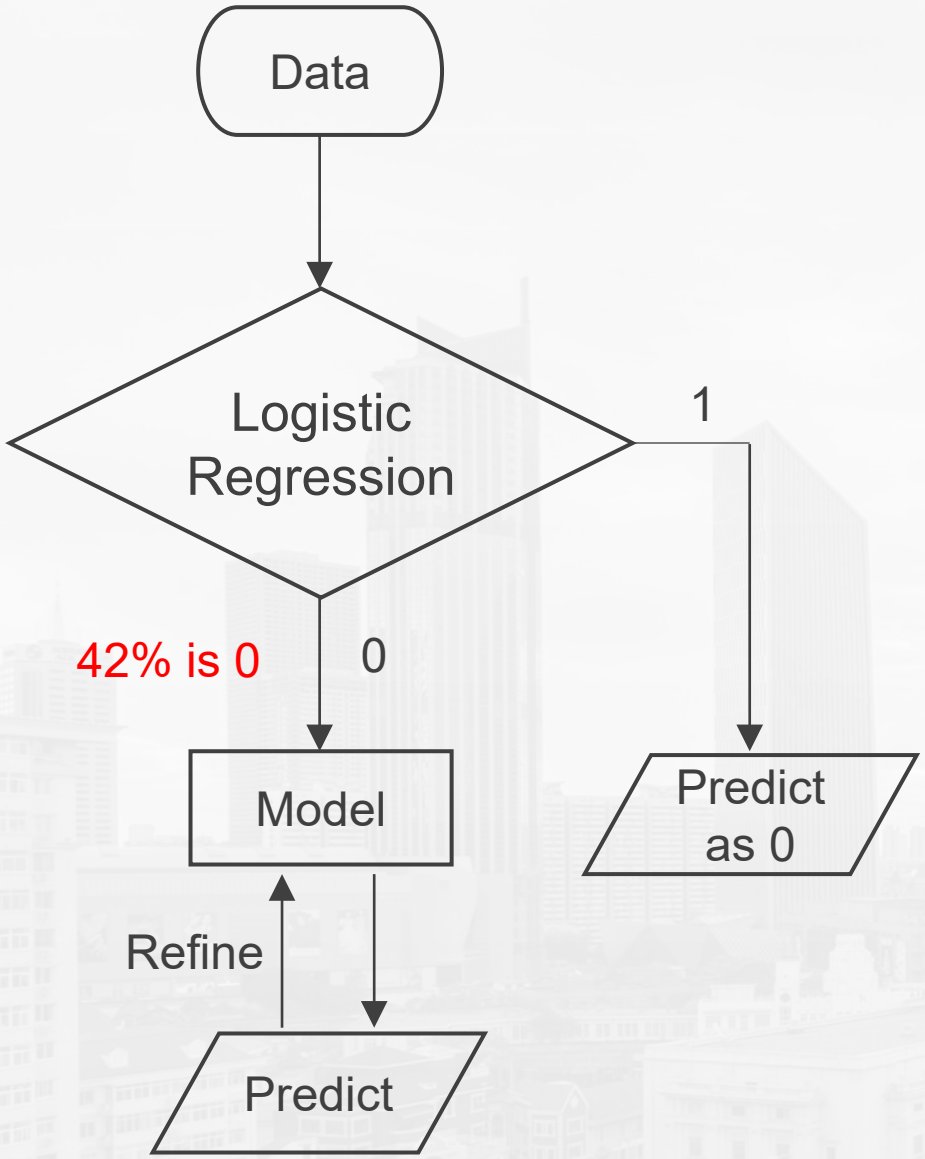


Optimal Threshold: 0.67

True Positive Rate: 70.94%

False Positive Rate: 26.56%

Work Flow



Model	Note
Zero Prediction	All sample is predicted as 0
Linear Regression	If predicted value<0, then predict 0
Decision Tree	optimal depth and leaf number
Random Forest	optimal number of trees
General Model	Combination of models mentioned above

Evaluation Methods

Mean Squared Prediction Error (MSE):

$$MSE = \frac{\sum(Predicted\ Value - Actual\ Value)^2}{Number\ of\ Observations}$$

Evaluation Methods

Precision Score:

$$Deviation_{Forward} = \frac{|\text{Forward}_{predicted} - Forward_{actual}|}{Forward_{actual} + 5}$$

$$Deviation_{comment} = \frac{\sum Comment_{predicted} - Comment_{actual}|}{Comment_{actual} + 3}$$

$$Deviation_{like} = \frac{|like_{predicted} - like_{actual}|}{like_{actual} + 3}$$

Precision for single Weibo:

$$Precision_i = 1 - 0.5 \times Deviation_{repost} - 0.25 \times Deviation_{comment} - 0.25 \times Deviation_{like}$$

weighted Averaged Score:

$$Precision\ Score = \frac{\sum(interaction_i + 1) \times sign(Precision_i - 0.8)}{\sum(interaction_i + 1)}$$

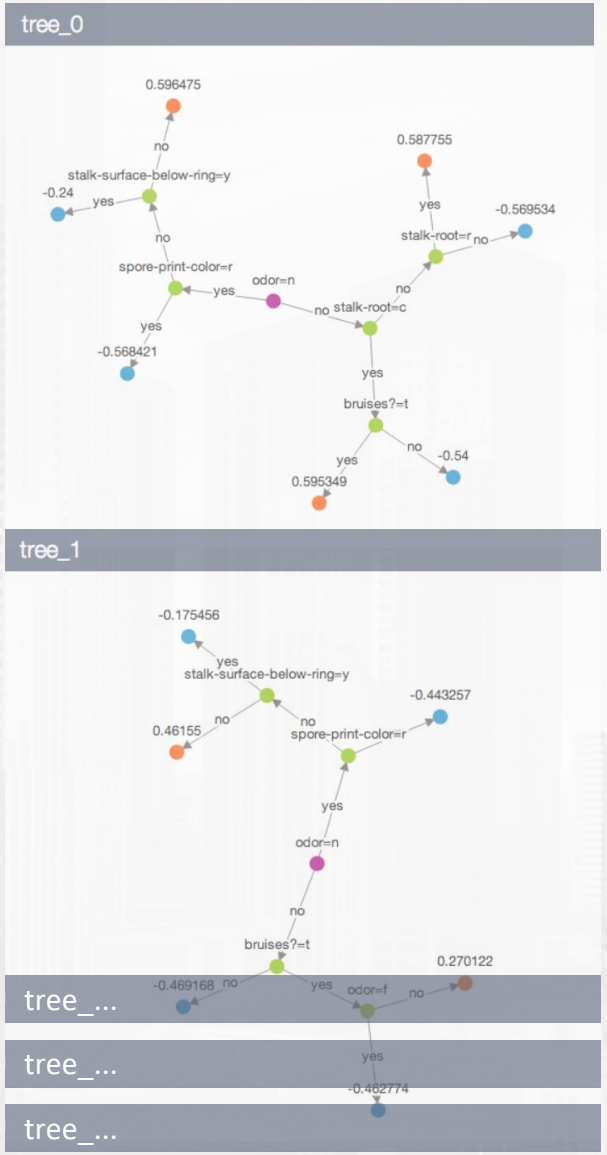
PART 05

Implementation & Result

Methodology

Why random forest

- Non-linear relationships
- A decision tree is prone to overfitting
- Almost the best supervised learning algorithms
- Efficient estimates of the test error without the cost of cross-validation
- Reliable feature importance estimate



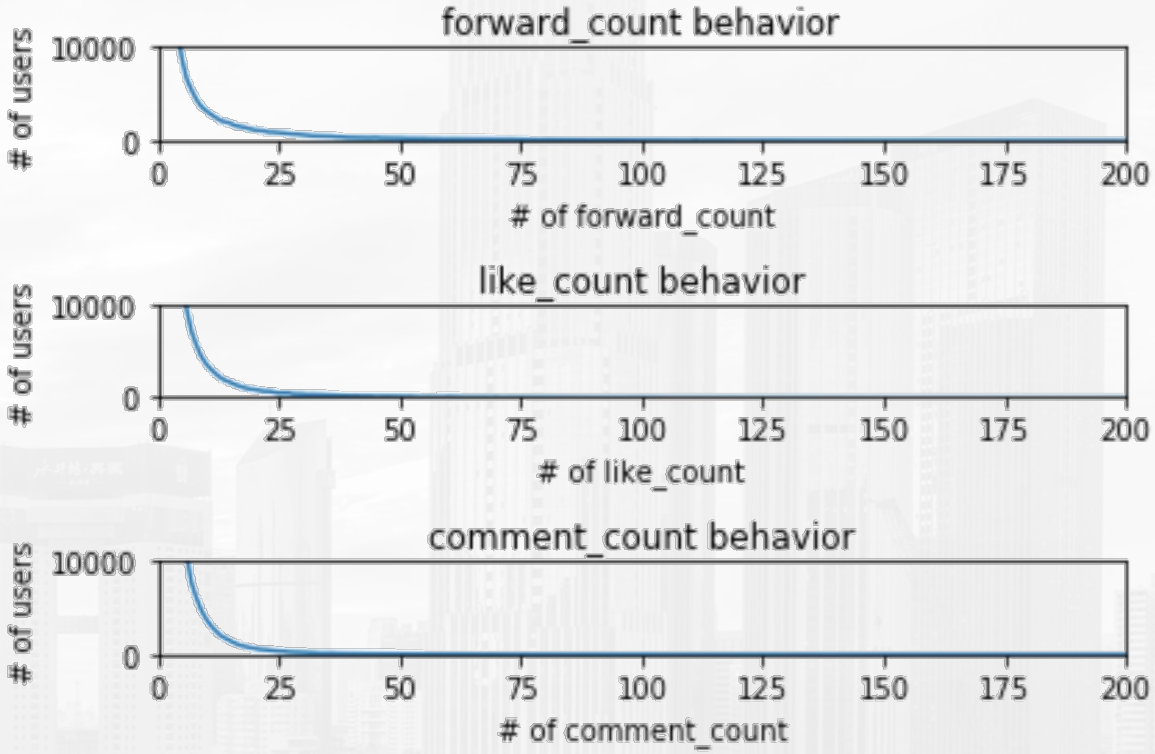
Jason, S.(2018). *Hands-On Machine Learning for Algorithmic Trading*. Birmingham: Packt Publishing. pp.311-312

Diagram from: https://turi.com/learn/userguide/supervised-learning/random_forest_regression.html

Methodology

Why a composite model

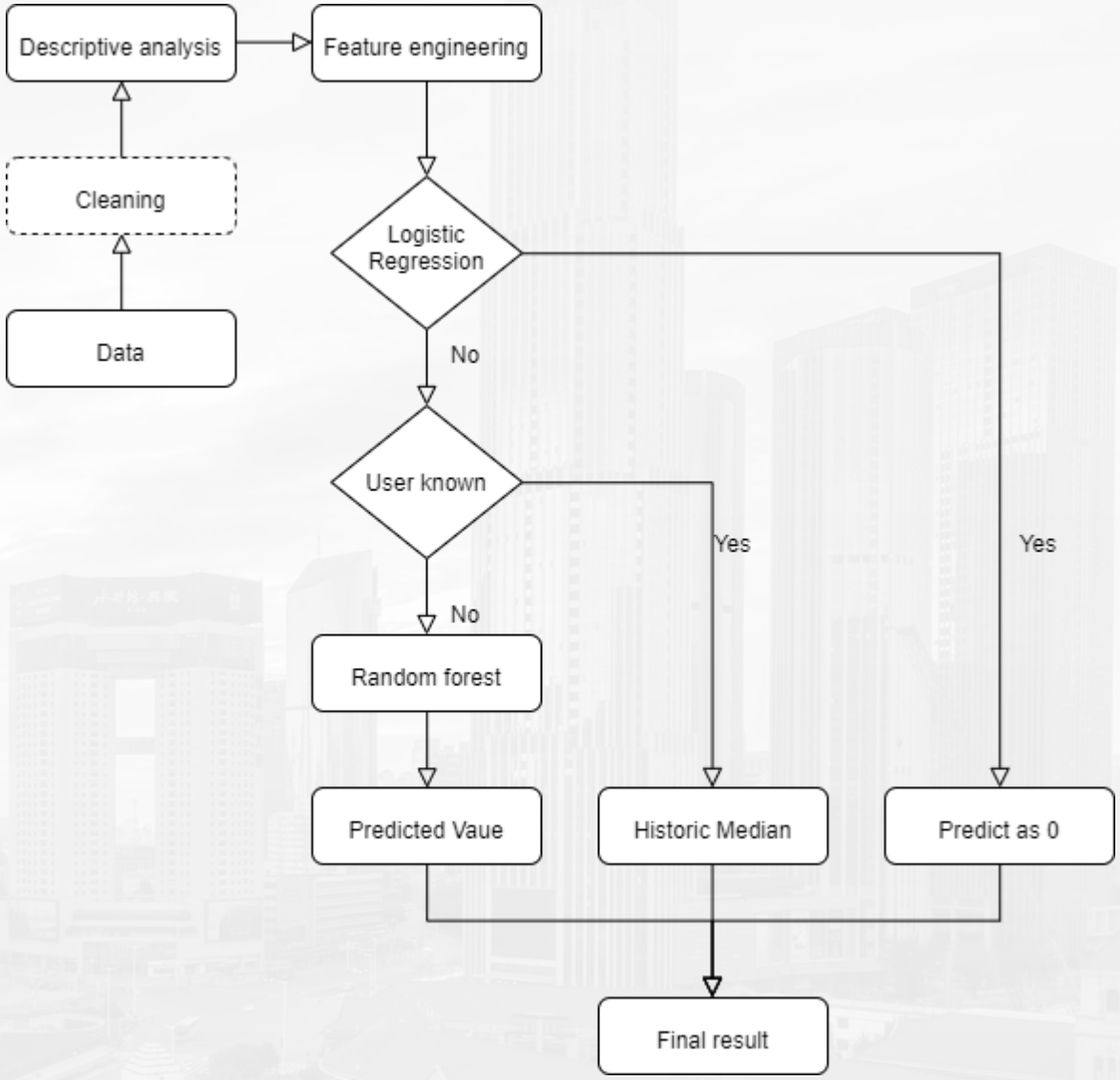
- Communication theory
  - Opinion leader
  - Echo chamber
  - …
- Data distribution

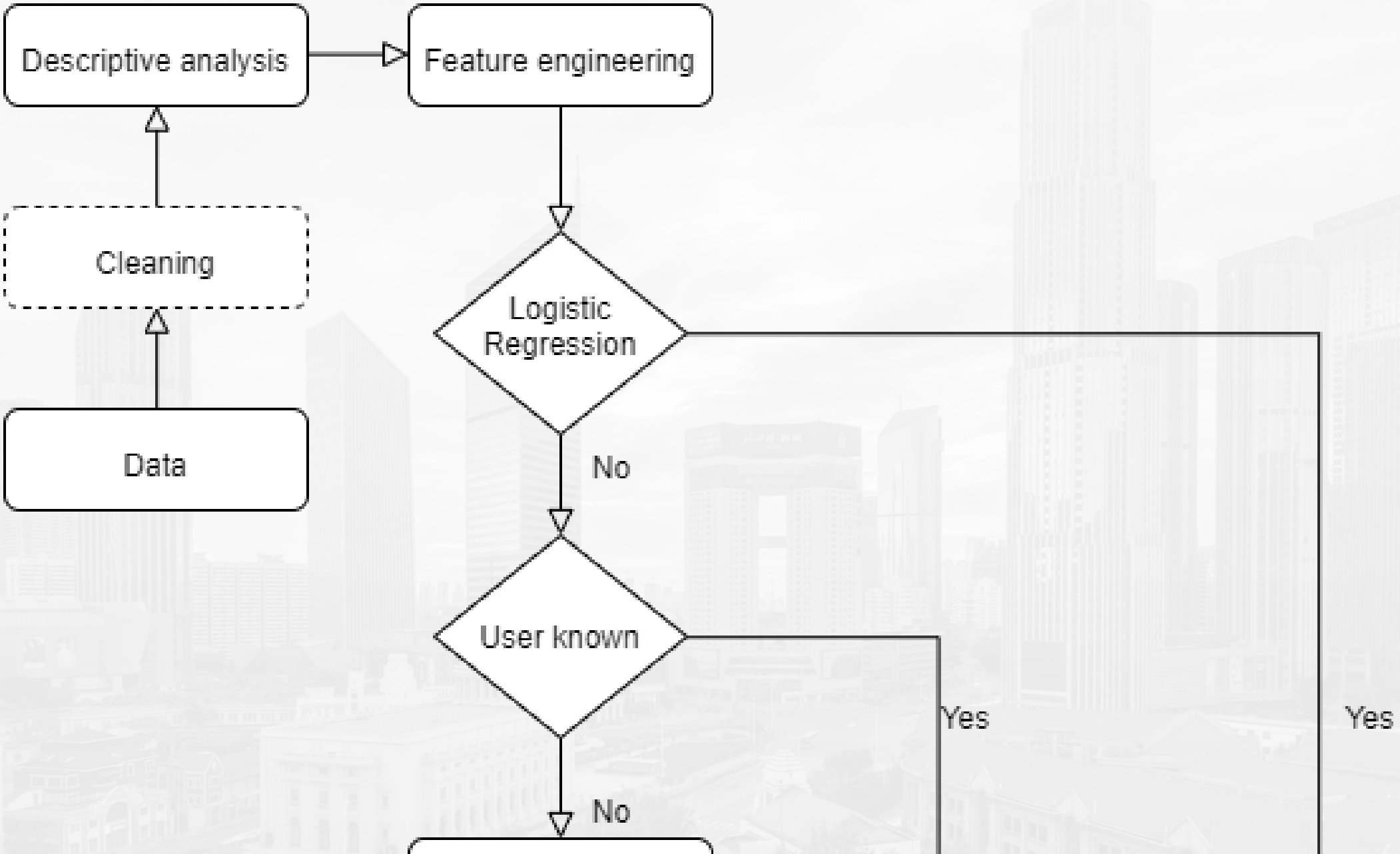


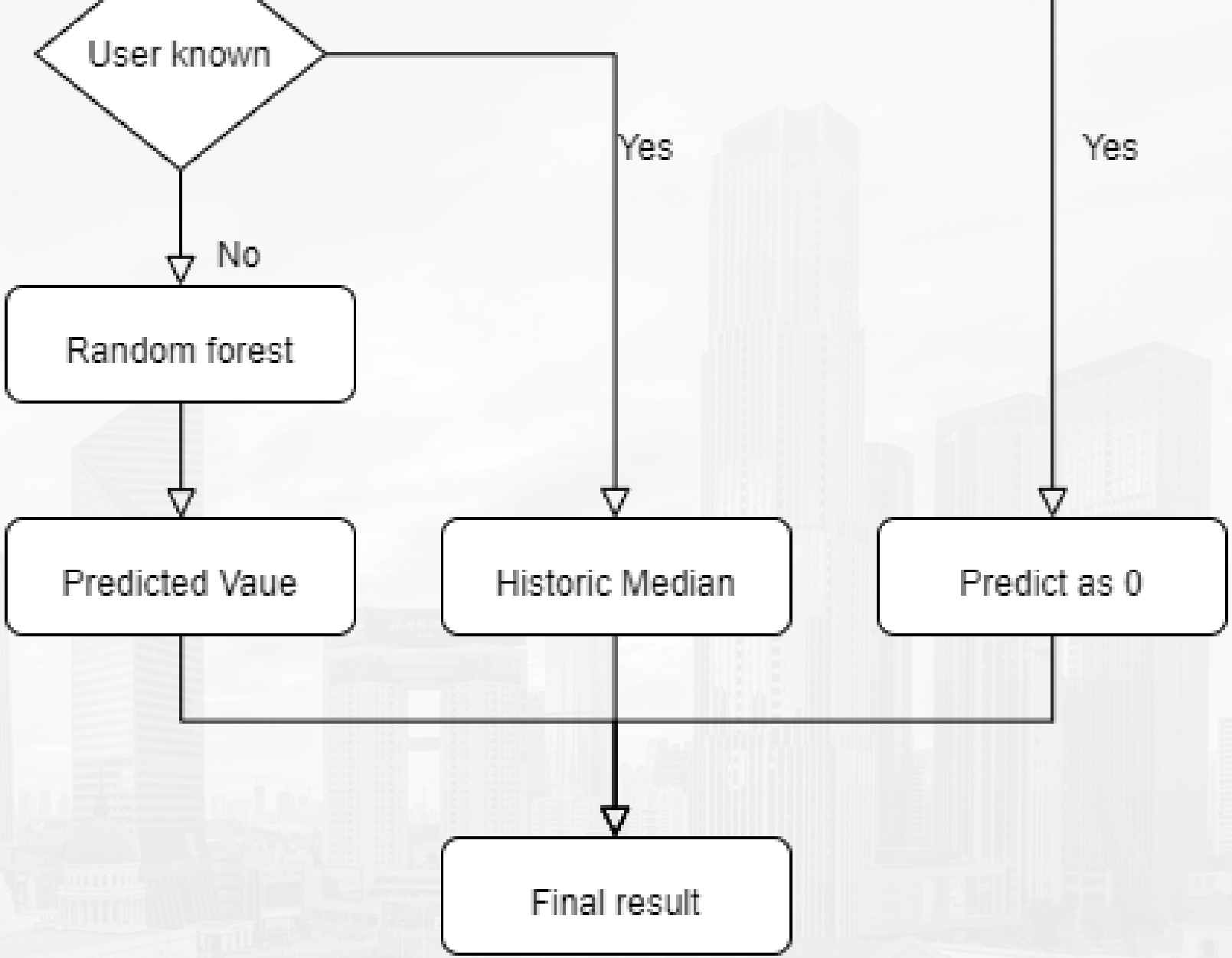
Methodology

Processing

- Logistic regression to find noise
- Random forest to predict
- Conservative prediction




Descriptive analysis
Feature engineering
Cleaning
Logistic Regression
Data
No
User known
Yes
Yes
No


User known
Yes
Yes
No
Random forest
Predicted Vaue
Historic Median
Predict as 0
Final result

Implementation

Module Selection

- Eligibility
- Accuracy
- Performance

Module	Rows	Time(s)	RAM(GB)	Accuracy
scikit-learn	10,000	1.22	0.15	0.922
	100,000	10.88	0.49	0.917
	1,000,000	162.18	4.1	0.914
Spark MLlib	10,000	111.99	10.0	0.912
	100,000	186.86	40.5	0.915
	1,000,000	NA	OOM	NA
DolphinDB	10,000	0.33	0.33	0.928
	100,000	1.74	3.0	0.915
	1,000,000	21.01	28.6	0.906
XGBoost	10,000	3.61	0.09	0.916
	100,000	29.46	0.19	0.905
	1,000,000	379.38	1.2	0.893

500 trees with max_depth = 10, Intel E5-2650 v4, 512 GB RAM,
Source: https://www.infoq.cn/article/RZAj8mVWTu5clOcT-fOU

Implementation

Parameter tuning

- Computation resource limitation
- Experience based experiment
- 1000, 50, 10, 50

n_estimator 10, 50, 100, 500, 1000

depth = 10, 20, 50

leaf_num = 10, 20, 50

split_num = 10, 50, 100

njobs * = -1

```
RandomForestRegressor(
      n_estimators = 1000 ,
      max_depth = 50,
      min_samples_leaf = 10,
      min_samples_split = 50,
      oob_score = oob,
      random_state = 42,
      n_jobs = -1,
      max_features = "auto")
```

Implementation

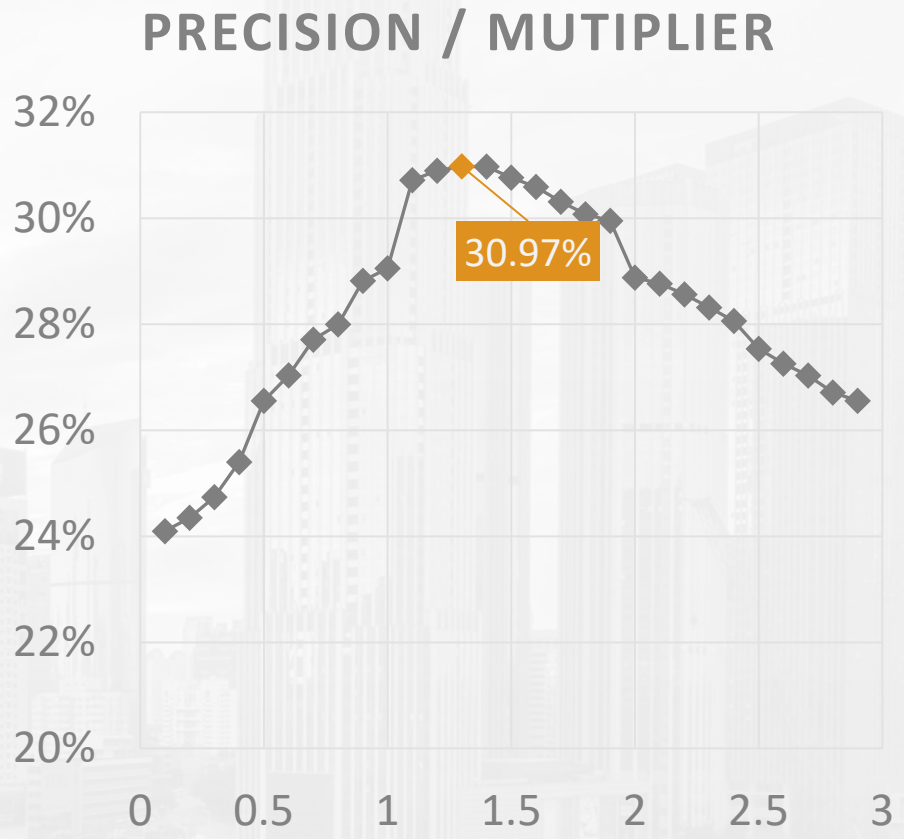
Type Conversion

Float values generated but integers expected.

- `int(predicted_value)`?
- `int(multiplier * predicted_value)`

Where *multiplier* is defined as:

$$Max(precision(reposts, comments, likes))$$



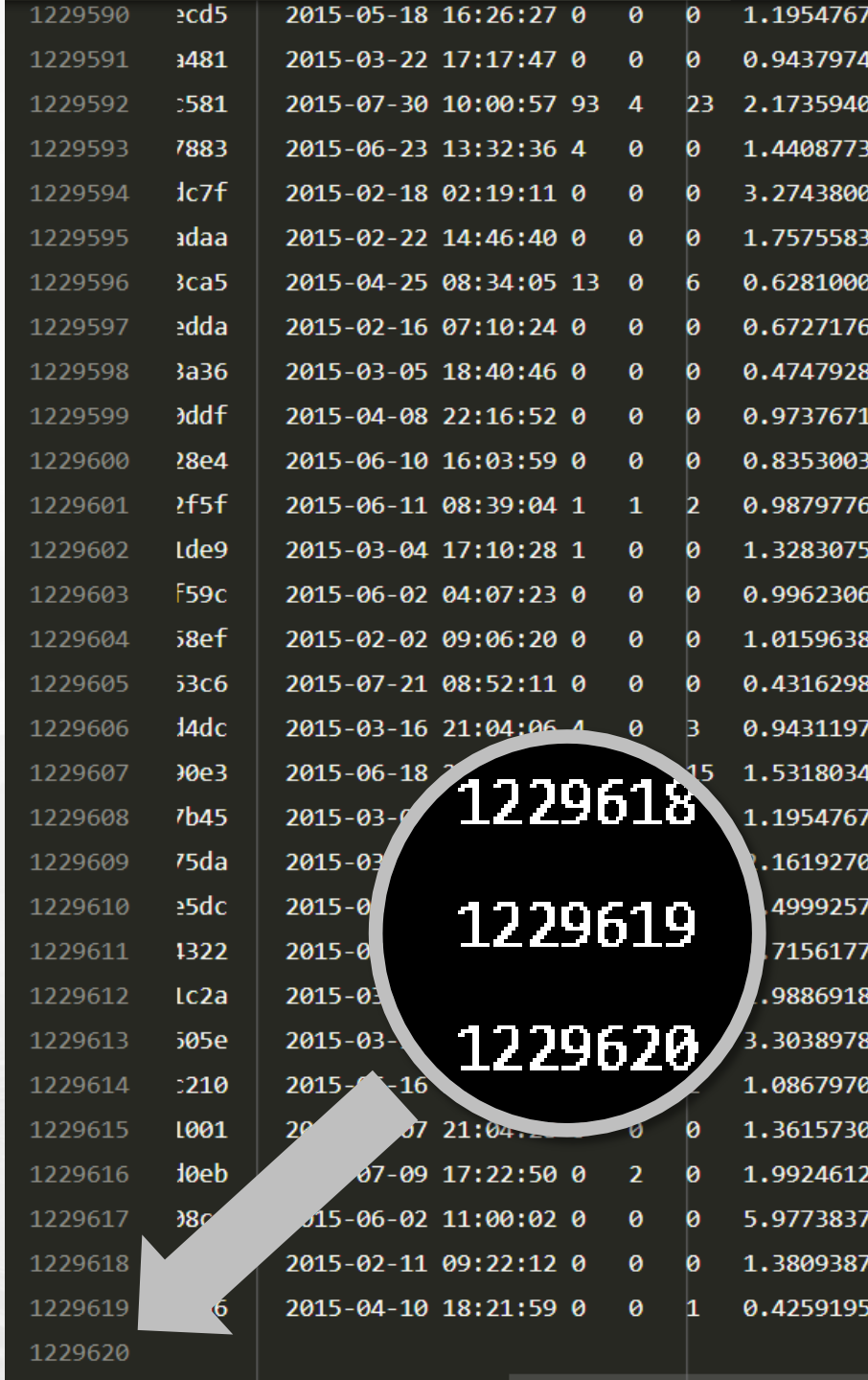
* This model with score 30.97% is not the final one due to its failure in online prediction.

Implementation

“A big question”

Volume and variety in this case.

- *1, 229, 619 cases*
- Vectorization with Pandas
- For-loop or df.apply() unpreferred
- Jieba_fast: rewritten in C
- *Spark

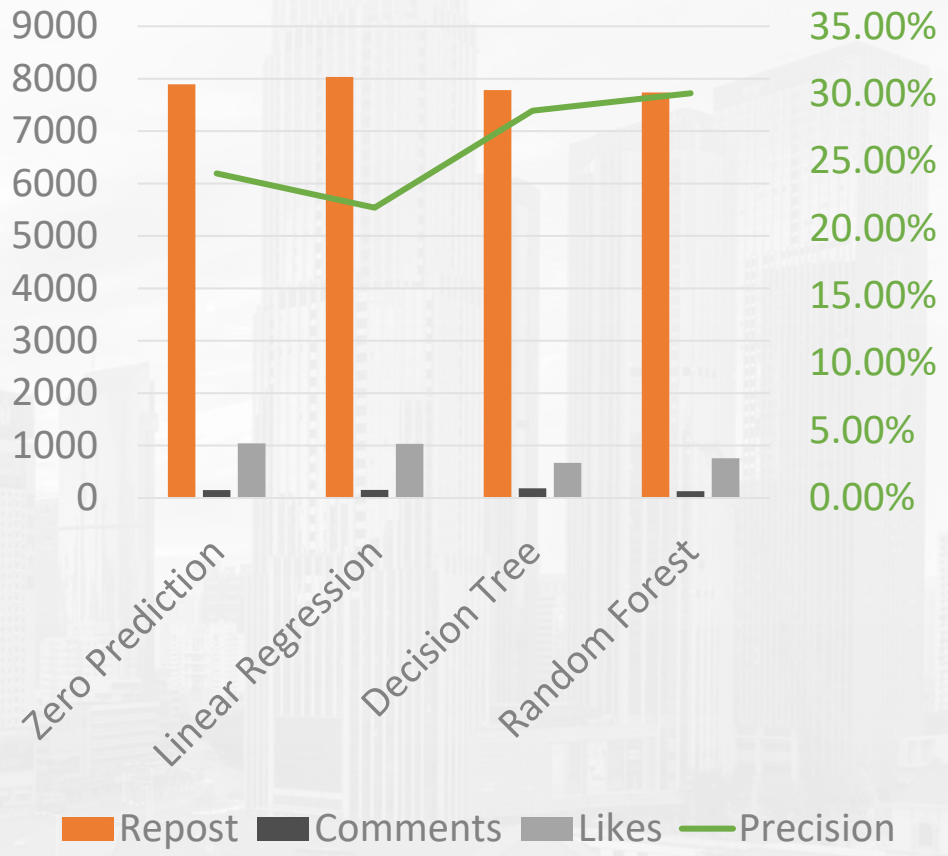


Result

Offline Results

MSE	Zero Prediction	Linear Regression	Decision Tree	Random Forest*
Repost	7893.77	8035.61	7785.75	7739.45
Comments	149.56	152.26	184	129.67
Likes	1041.5	1033.93	670.22	757.2
Precision Score	24.09%	21.54%	28.74%	30.03%

* Random forest + history median.



Result

Online Ranks

- From 23.26% to 29.74%
- Top 5% in five years
- Best open-source solution

Methods	Zero Prediction	Random Forest*
Precision	23.26%	29.74%
Rank	Top 16% / 693th	Top 5% 251th
Rank (PKUers)	Last one	3rd
Badge from Ali.	New Bee	Data Master
Open-source	Average	Best ever

* Full rank-list at: https://tianchi.aliyun.com/competition/entrance/231574/rankingList

Reference

- Feng, D., & Wu, X. (2018). Weibo interaction in the discourse of internet anti-corruption: The case of “Brother Watch” event. Discourse, Context & Media, 24, 99-108.
- Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. EURASIP journal on wireless communications and networking, 2017(1), 211.
- Jason, S.(2018). *Hands-On Machine Learning for Algorithmic Trading*. Birmingham: Packt Publishing. pp.311-312
- Yao, W., Jiao, P., Wang, W., & Sun, Y. (2019). Understanding human reposting patterns on Sina Weibo from a global perspective. Physica A: Statistical Mechanics and its Applications, 518, 374-383.