Optimal POS Placement

Waqas Bukhari, PhD

Study objective

Determine the facilities (amenities) that can impact POS performance.

Data Sources

Sales information of POS

Record of 90 types of amenities around 546 POS

Sales information of POS

Record of 90 types of amenities around 546 POS

Target variable design

Assumption

Amenities around a POS impact its sale

Assumption

Amenities can influence overall sales potential

Amenities can influence overall sales potential

Manipulate sales data to an intermediate resolution

store_code	10055	10077	10079	10081	10085	10086
2017-06-25 18:00:00	NaN	NaN	NaN	150.0	NaN	NaN
2017-06-25 19:00:00	NaN	NaN	NaN	600.0	NaN	60.0
2017-06-25 20:00:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-06-25 21:00:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-06-25 22:00:00	NaN	NaN	NaN	NaN	NaN	NaN

store_code	10055	10077	10079	10081	10085	10086
week_numb						
94	990.0	30.0	5430.0	13260.0	1650.0	3420.0
95	90.0	0.0	4260.0	7980.0	2220.0	1020.0
96	390.0	630.0	3270.0	6690.0	1560.0	2190.0
97	870.0	270.0	3360.0	6270.0	1110.0	1500.0
98	900.0	1050.0	3360.0	8490.0	1050.0	1470.0

Possible target variable

mean weekly sales

Exponentially weighted mean weekly sales

Target variable design

Possible target variable

mean weekly sales

Exponentially weighted mean weekly sales

Possible target variable

mean weekly sales

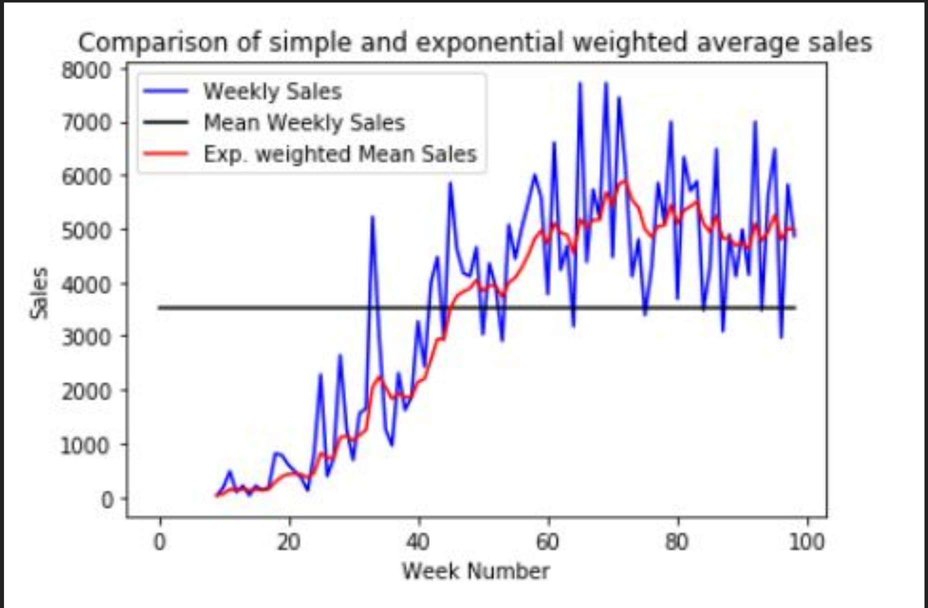
Exponentially weighted mean weekly sales

Target variable design

Possible target variable

mean weekly sales

Exponentially weighted mean weekly sales



Data Preparation

Data Preparation aka feature engineering

Count on each type of amenity around POS

Existence of each type of amenity around POS

Total and average ratings of each type of amenity

Representation and overall count on 90 amenities around store

Two way interactions on the existence of amenities

Why not location features?

Log transformation on target variable

Count on each type of amenity around POS

	accounting	airport	amusement_park	aquarium	art_gallery
store_code					
10055	3	0	0	0	1
10077	0	0	0	0	0
10079	1	0	0	0	0
10086	0	0	0	0	0
10111	0	0	0	0	0

Existence of each type of amenity around POS

	has_subway_station	has_department_store	has_embassy	has_beauty_salon	has_police
store_code					
10055	0	0	0	1	0
10077	0	0	0	0	0
10079	0	1	0	1	0
10086	0	0	0	1	0
10111	0	0	0	0	0

Total and average ratings of each type of amenity

	amusement_park_rating_count	art_gallery_avg_rating	art_gallery_rating_count	atm_avg_rating	atm_rating_count
store_code					
10377	0	4.652381	21	4.323077	13
10441	0	4.999995	1	4.999998	2
10545	0	0.000000	0	2.999997	1
10548	0	0.000000	0	2.999997	1
10672	0	0.000000	0	0.000000	0

Two way interactions on the existence of amenities

	has_subway_station_and_has_department_store	has_subway_station_and_has_embassy	has_subway_station_and_has_beauty_salon	has_subway_station_and_has_pharmacy	has_subway_station_and_has_local_government_office
store_code					
10814	0	0	0	0	0
10820	0	0	0	0	0
10871	1	0	1	1	0
10883	0	0	0	0	0
10928	0	0	0	0	0

Why not location features

Good to know 2 POSs around Shibuya, Tokyo did well?

Why not location features

Make another POS at “good” location?

Data Split

80% train and 20% test data

Stratified sampling to split dataset

Training data ~ model building and selection

Test data ~ model testing

80% train and 20% test data

Stratified sampling to split dataset

Training data ~ model building and selection

Test data ~ model testing

80% train and 20% test data

Stratified sampling to split dataset

Training data ~ model building and selection

Test data ~ model testing

Data split

80% train and 20% test data

Stratified sampling to split dataset

Training data ~ model building and selection

Test data ~ model testing

Data Modeling

Modeling Objective ~ Find amenities that drive sales ?

Technical translation ~ Feature extraction

Modeling Objective ~ Find amenities that drive sales ?

Technical translation ~ Feature extraction

Linear Regression for feature extraction

Bag of Linear Regression for feature extraction and then Simple Linear Regression

Doing in Linear and Log space and find the best model

Feature extraction algorithm

data in two folds

F = {} ~ Extracted features

V ~ Candidate features

h(F) ~ model based on F

P(F) ~ performance of h(F) over two folds

P(F1) > P(F2)

v_k ~ candidate feature

```
While True:
    For each v_k in V not in F:
        F_k = F U v_k
        Maintain k that yields best P(F_k)
    If P(F_k) > P(F)
        Add v_k to F
```

Feature extraction algorithm

Split data into two folds

F = {} ~ Extracted features

V ~ Candidate features

h(F) ~ model based on F

P(F) ~ performance of h(F) over two folds

P(F1) > P(F2) iff h(F1) is strictly better than h(F2) over both folds

v_k ~ candidate feature

```
While True:
    For each v_k in V not in F:
        F_k = F U v_k
        Maintain k that yields best P(F_k)
    If P(F_k) > P(F)
        Add v_k to F
    Else
        break
```

Let , F = [a,b], V = [c,d,e]

Evaluate P([a,b,c]), P([a,b,d]), P([a,b,e])

Let P([a,b,d]) is best.

If P([a,b,d]) > P(F)

Update F to [a,b,d]

```
While True:
    For each v_k in V not in F:
        F_k = F U v_k
        Maintain k that yields best P(F_k)
    If P(F_k) > P(F)
        Add v_k to F
    Else
        break
```

Linear Regression

Linear Regression

		mean	std	sample size
has_university_and_has_library	**False**	436.403427	743.188716	403.0
	True	2669.633061	3218.556594	33.0
has_university_and_has_museum	**False**	592.169741	1282.407088	412.0
	True	833.105786	1111.568663	24.0
cafe	**False**	375.918164	758.930580	155.0
	True	732.032587	1469.066132	281.0
has_subway_station_and_has_gym	**False**	566.780021	1169.782709	433.0
	True	6184.24	3200.85	3.0
has_university_and_has_painter	**False**	480.070921	912.160842	408.0
	True	2432.126298	3145.664540	28.0

Bag of Linear Regression models

Bag of Linear Regression models

Bagging Features

----

cafe
book_store
laundry
laundry_avg_rating
electronics_store
has_gas_station_and_has_laundry
has_shopping_mall_and_has_locksmith
has_shopping_mall_and_has_movie_theater
doctor_rating_count
cafe_avg_rating
bus_station
local_government_office_avg_rating
has_city_hall_and_has_lodging
has_gas_station_and_has_museum
book_store_avg_rating

Bag of Linear Regression models

Bagging Features
----
cafe
book_store
laundry
laundry_avg_rating
electronics_store
has_gas_station_and_has_laundry
has_shopping_mall_and_has_locksmith
has_shopping_mall_and_has_movie_theater
doctor_rating_count
cafe_avg_rating
bus_station
local_government_office_avg_rating
has_city_hall_and_has_lodging
has_gas_station_and_has_museum
book_store_avg_rating

←- Final Features

Performance Evaluation

Performance evaluation

	Model	Train RMSE	Test RMSE
0	base model	1272.190545	1064.677986
1	linear regression - Y	1041.93	881.51
2	Decision Tree - Y	773.581141	1116.455479
3	linear regression - log(Y)	1217.922197	934.088890
4	Decision Tree - log(Y)	1056.802804	1033.520525

Best model

Best Model

	Regression Coefficient
cafe	183.472003
laundry	582.453222
laundry_avg_rating	-232.807086
book_store	-154.566553
electronics_store	-96.292089

Best Model

	Regression Coefficient
cafe	183.472003
laundry	**582.453222**
laundry_avg_rating	**-232.807086**
book_store	-154.566553
electronics_store	-96.292089

Best Model

	Regression Coefficient
cafe	183.472003
laundry	582.453222
laundry_avg_rating	-232.807086
book_store	-154.566553
electronics_store	-96.292089

Conclusions and Limitations

Conclusion

An end-to-end analysis and modeling

Multiple features are created

A stable feature extraction algorithm

Evidence ~ POS sales based on its surroundings

Major limitation ~ geographic demands not in consideration

Questions