Google Merchandise Store Transaction Revenue Prediction

Zhenru Han
GR5261 Final Project (Spring 2019)
Instructor: Prof. Zhiliang Ying, George Chu, Jitong Qi

1. Introduction

Given data comprise of 1,708,337 records for transaction histories at Google Merchandise Store from August 2016 to April 2018 obtained from Google Customer Revenue Competition on Kaggle, the case study is designed for achieving the following objectives:

1. Build regression models to investigate which model is the most useful in explaining transaction revenue.
2. Implement time series methodology to forecast daily revenue of G-Store.

1.1 Data Cleaning

The original data contains 36 factors and their descriptions are attached to Appendix 1. The response variable is "Transaction Revenue", but even within the response variable there exist missing values (98.7%). For both research I am interested in, some features are not necessary to be included.

I. Regression Models for Predicting Individual Transaction Revenue

a. Convert "Date" (format = YYYYmmdd) to two columns "year", "weekday" (1-7 represent Mon-Sun).
b. Convert "Visit Start Time" to "hour" (1-24 represent hours in a day).
c. Remove all ids, number of hits, date and visit start time.
d. Factorize character level features and "year", "weekday", "hour".
e. Create five new features each interpret the average number of pageviews by visit number, country, city, network domain and referral path.
f. Replace the NAs in "Transaction Revenue" to 0, since there is no revenue generated by these visits.

II. Time Series Forecast on Daily Transaction Revenue

a. Format "Date" (YYYYmmdd) to YYYY-mm-dd.
b. Calculate daily average transaction revenue (log transformed) site-wide
c. Keep two columns "Date" and "Transaction Revenue" for further investigation.

1.2 Train/ Prediction Split

For both objectives, cross-validation is a reasonable method for model evaluation and optimization. Since the last day contains non-NA value in "Transaction Revenue" is Aug 1st, 2017, the training/validation set should include observations from 2016-08-01 to 2017-08-01. The remaining observations do not have records of "Transaction Revenue", so they are perfect for predictions.

2. Statistical Models

2.1 Regression Models for Predicting Individual Transaction Revenue

Given data having 37 features after preprocessing, reducing the overfitting risk is necessary. Principle component analysis is a useful tool in feature engineering, ridge regression gives a shrinkage in parameters and lasso regression automatically select features based on penalty. XGboost is also a proper approach in building regression model under this condition since it randomly tune features and observations to reach the best combination.

The evaluation metric for regression models is RMSE (root mean squared error): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ and

MSE (mean squared error): $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Transformation of "Transaction Revenue" is needed since it is not normally distributed and contains extreme values. The largest value is 23028.54 while the majority lies between 0.99 and 99.99. The log transformation on "Transaction Revenue"

+1 (+1 to avoid problem caused by 0's) results in an approximate normal distribution (See Appendix 2), which is more appropriate for modelling comparing to the original values.

I. Ridge Regression (with PCA feature engineering)

a. Exclude "Transaction Revenue" as y, and convert all factor level features to dummy variables.
b. According to the principle component summary, observed that the first 6 features are able to explain 95% of the data.

```
Importance of components:
                           PC1    PC2     PC3     PC4     PC5     PC6    PC7     PC8     PC9
Standard deviation     9.6606 9.1964 3.57810 2.69856 2.04291 1.54069 1.1742 0.97558 0.93132
Proportion of Variance 0.4334 0.3928 0.05946 0.03382 0.01938 0.01102 0.0064 0.00442 0.00403
Cumulative Proportion  0.4334 0.8262 0.88568 0.91950 0.93888 0.94990 0.9563 0.96073 0.96476
```

c. Keep the first 6 features and fit ridge regression to the log revenue, train the model by cross validation using number of folds = 5, family = "gaussian" and type of measure = "mse".
d. Calculate RMSE of the best model obtained from cross-validation, which equals to 0.4251. The relative MSE is 0.1807. In the scale of 9.2 to 23.9, the metrics represent a good performance of this model.
e. Predict future transaction revenue for each visit from 2017-08-02 to 2018-04-30 and save the results.

II. Ridge Regression & Lasso Regression (without PCA feature engineering)

a. Exclude "Transaction Revenue" as y and fit ridge and lasso regression to the log revenue, train the models by cross validation using number of folds = 5, family = "gaussian" and type of measure = "mse".
b. Calculate RMSE of the best models obtained from cross-validation.
    i. Ridge Regression: RMSE = 0.4213 MSE = 0.1775.
    ii. Lasso Regression: RMSE = 0.4219 MSE = 0.1780
c. Predict future transaction revenue for each visit from 2017-08-02 to 2018-04-30 and save the results.

III. XGboost Regression

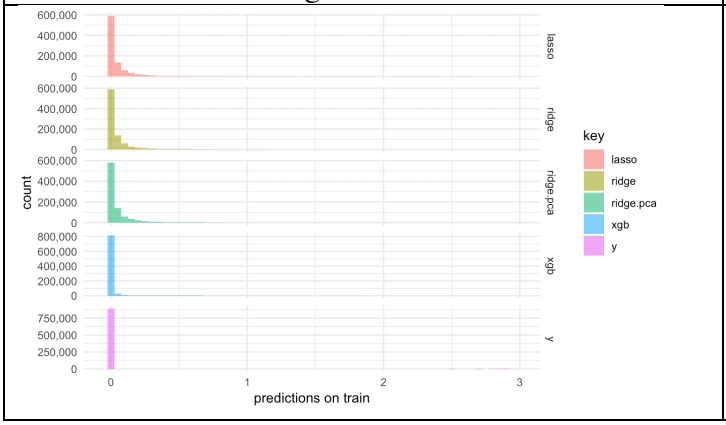
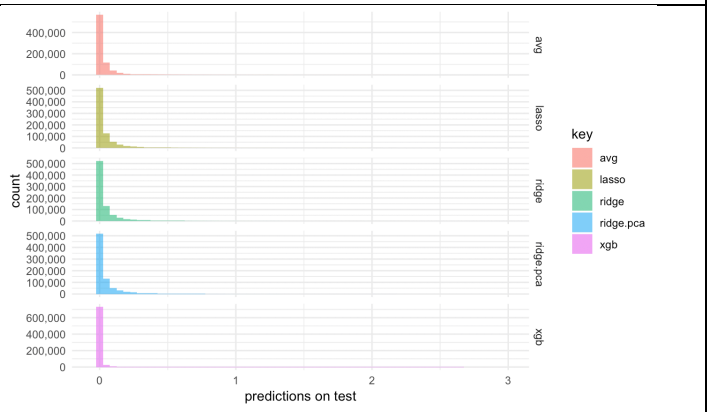
a. Exclude "Transaction Revenue" as y and set training set index to be date from 2016-08-01 to 2017-05-15 (set two and a half months aside, similar to the number of observations in one fold in glmnet).
b. Construct xgb.DMatrix of training and validation sets. Training set includes the observations lie within 2016-08-01 to 2017-05-15 and validation set lies within 2017-05-16 to 2017-08-01.
c. Train the XGboost regression model by setting booster = "gbtree", evaluation metric = "rmse", number of thread = 4, eta = 0.05, maximum depth = 7, minimum child weight = 5, gamma = 0, subsample = 0.8, column sample by tree = 0.7, column sample by level = 0.6, and number of rounds = 2000 (The parameters are automatically tuned by package nlme).
d. RMSE of the best model obtained from validation is 0.3972. It performs the best among all four models constructed. From features' percentage importance plot (See Appendix 3), it is clear to observe that the average pageviews by network domain and pageviews contribute the most, both greater than 20% among all features selected for the final model.
e. Predict future transaction revenue for each visit from 2017-08-02 to 2018-04-30 and save the results.

Models Comparison:

	Ridge with PCA	Ridge without PCA	Lasso	**XGboost**
Validation Method	Cross-Validation, folds = 5	Cross-Validation, folds = 5	Cross-Validation, folds = 5	**Set aside observations of two and a half months**
RMSE	0.4251	0.4213	0.4219	**0.3972**
MSE	0.1807	0.1775	0.1780	**0.1580**

Clearly XGboost gives the best model performance through validation. Interestingly, other regression models are not significantly different from each other, which indicates that the PCA feature selection method is reasonable to be applied under this condition. The evaluate metrics (MSE and RMSE) of all four models prove their usefulness in explaining the data, so these models' predictions on the test set also have high reference value.

Predictions Comparison:

Predictions on Training Set	Predictions on Test Set
	

Looking at the predictions on training set, linear models constructed using package glmnet all have similar distributions. Comparing to the distribution of true log revenue, XGboost gives the best approximation. Predictions on test set have similar behavior as the predictions on training set. These comparisons also represent the reference value of XGboost model results.

2.2 Time Series Models for Forecasting Daily Average Transaction Revenue at G-Store

Given the daily mean transaction revenue at G-store indexed in time order, it is proper to apply time series methods to make predictions. I majorly implemented two methods:

a. Smooth the original time series to identify trends and predict common patterns.
b. Apply ARIMA model to make the series stationary.

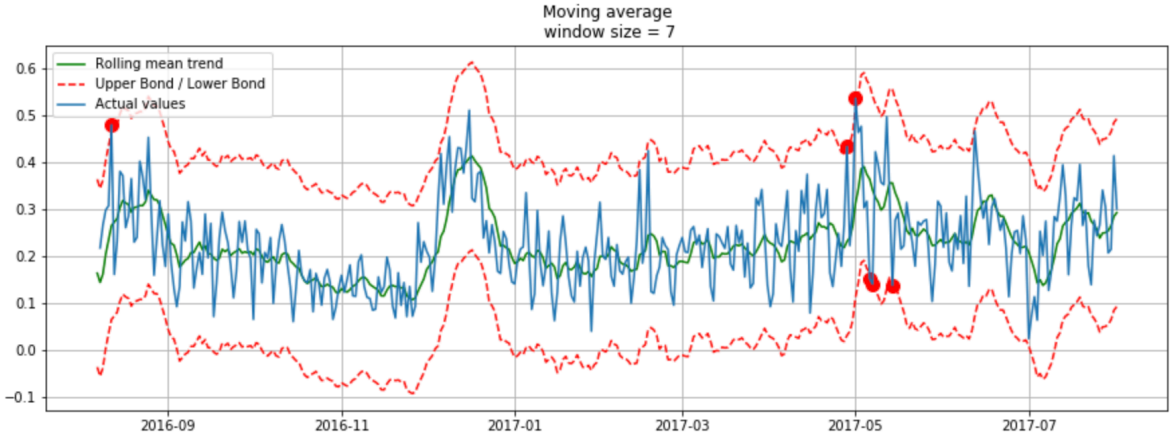
The evaluation metric for time series models are RMSE (root mean squared error): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ and

$$\text{AIC: } 2k - 2\ln(\hat{L}_t).$$

Usually the error estimator should be mean absolute percentage error, but since the data contain zero, mean absolute percentage error will be infinity. Using RMSE is also consistent with the metric of regression models.

I. Smooth by the Previous Seven Days

Assume that the future value of transaction revenue depends on the average of its seven previous values: $\hat{y}_t = \frac{1}{7}\sum_{i=1}^{7} y_{t-i}$. However, applying weekly smoothing has a downside. It captures the main trend of the series but ignores all the peaks. It is better to consider a more complex model.



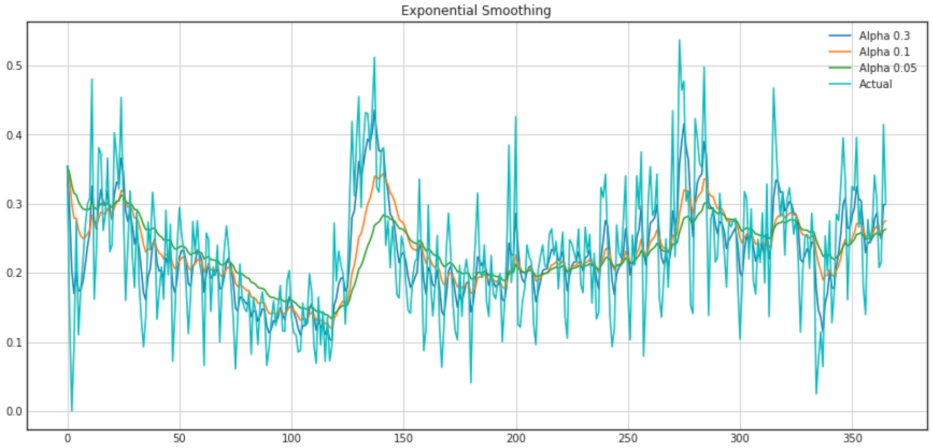
II. Exponential and Double Exponential Smoothing

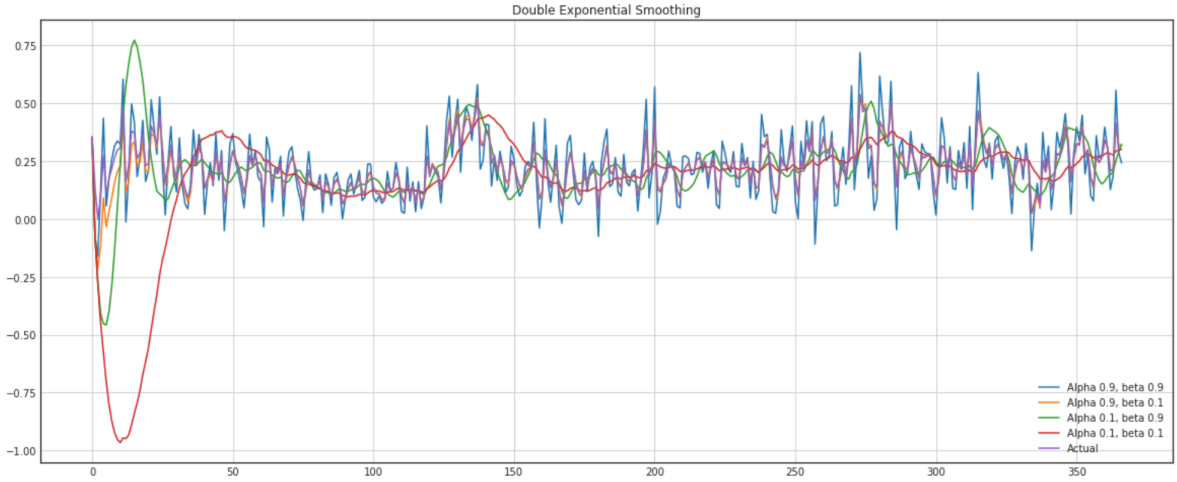
Instead of weighting the last 7 values of the time series, weighting all available observations while exponentially decreasing the weights as moving further back in time can be another approach.

$$\hat{y}_t = \alpha \cdot y_t + (1-\alpha) \cdot \hat{y}_{t-1}$$

Extend exponential smoothing to predict two future points, which is called double exponential smoothing.

$$l_x = \alpha \cdot y_x + (1-\alpha) \cdot (l_{x-1} + b_{x-1})$$
$$b_x = \beta \cdot (l_x - l_{x-1}) + (1-\beta) \cdot b_{x-1}$$
$$\hat{y}_{x+1} = l_x + b_x$$





Certain combinations of the parameters produce strange results, especially if set manually (see the plot of Double Exponential Smoothing). Also, it is reasonable to think about the influence of seasonality, which is the third dimension of exponential smoothing: Holt-Winters.

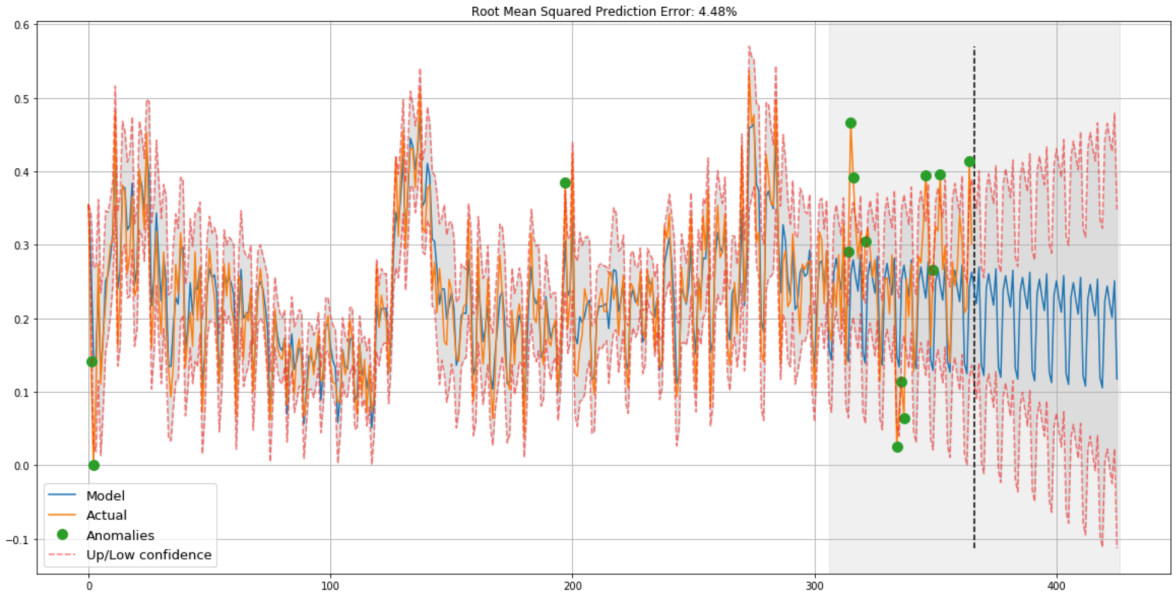
III. Triple Exponential Smoothing (Holt-Winters)

Expecting the time series has seasonality, seasonal components in the model will explain repeated variations around intercept and trend, and it will be specified by the length of the season. By adjusting seasonal component to be 7 (each for a day in a week), 24, and 30, I am able to tune the other parameters based on each seasonal component. Using the parameters tuned to build models, predict the future, and measure the root mean squared errors, the best model obtained will have the seasonal component to be 7 (a week).

a. Evaluate the gradient descent and apply the time series cross-validation, which is based on chronological split (See appendix 4).
b. Use the loss function of triple exponential smoothing to tune the parameters during cross-validation:
T is the length of the season, d is the predicted deviation:

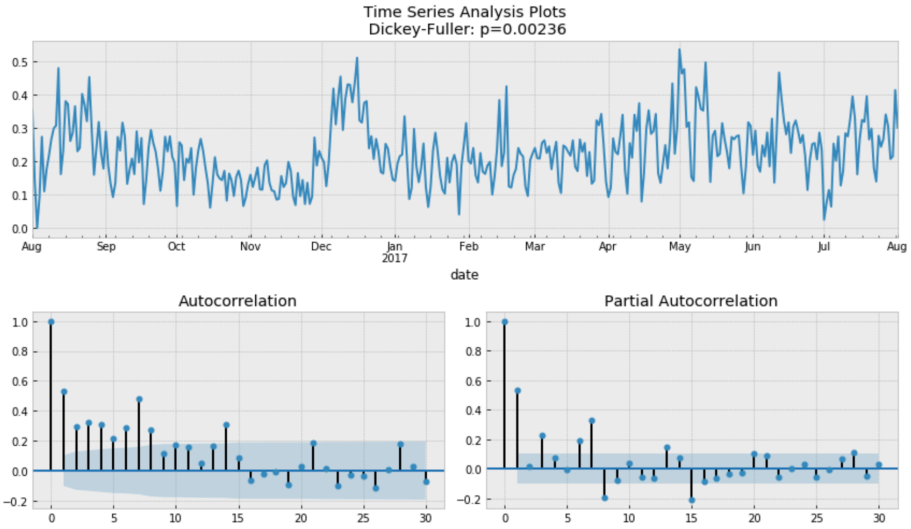
$$\hat{y}_{max} = l_{x-1} + b_{x-1} + S_{x-T} + m \cdot d_{t-T}$$
$$\hat{y}_{min} = l_{x-1} + b_{x-1} + S_{x-T} - m \cdot d_{t-T}$$
$$d_t = \gamma \cdot |y_t - \hat{y}_t| + (1-\gamma) \cdot d_{t-T}$$

The parameters α, β, γ = (0.4364463370941392, 0.020444060064036373, 0.1515457842751512)
c. Leave 60 observations as the test set and calculate the RMSE = 4.48%, and make predictions on the following 60 days.



IV. SARIMA Model

ARIMA model is a good way to make a series stationary.



Surprisingly, the initial series are stationary; the Dickey-Fuller test reject the null hypothesis that a unit root is present. Seasonality is significant and has a week shift. After taking a week shift, I am able to make assumptions on the parameters before auto-tune.

· AR(p) - autoregression model, p is most probably 0 since most others are not significant.
· I(d) - order of integration (non-stationary differences), d equals 0 as trying the first difference does not perform well.
· MA(q) - moving average model, q should be 0 as well as seen on the ACF.
· P - order of autoregression for the seasonal component of the model, which can be derived from PACF. P might be 4, since 7-th, 14-th, 21th and 28-th lags are somewhat significant on the PACF.
· D - order of seasonal integration. D equals 1 because we performed seasonal differentiation.
· Q - order of autoregression for the seasonal component of the model, which can be derived from ACF. Q is probably 1. The 7-th lag on ACF is significant while the others are not.

The best model obtained from cross-validation is SARIMAX(p = 1, d = 0, q = 1)x(P = 4, D = 1, Q = 1, s = 7) based on the smallest AIC = 1126.913. Interestingly, during the cross validation, the AIC stays the same as P walk through 0 to 4, which means the order of autoregression for the seasonal component of the model derived from PACF is not significant (Appendix 5), it can also be acquired from model results table. (Appendix 6)

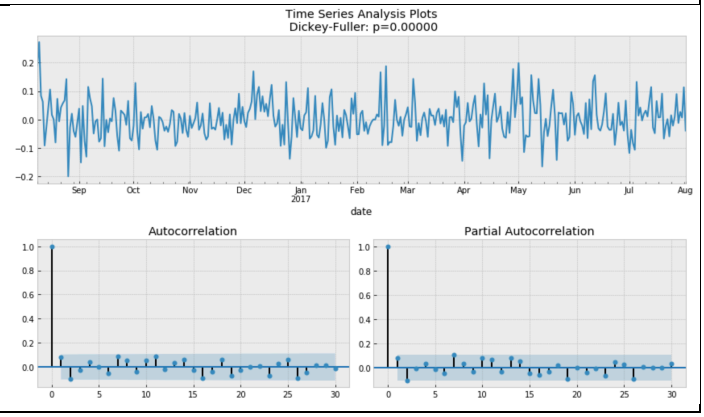
Hence, the final model selected for SARIMA is SARIMAX(p = 1, d = 0, q = 1)x(P = 0, D = 1, Q = 1, s = 7), and the model results are as followed:

Model Summary

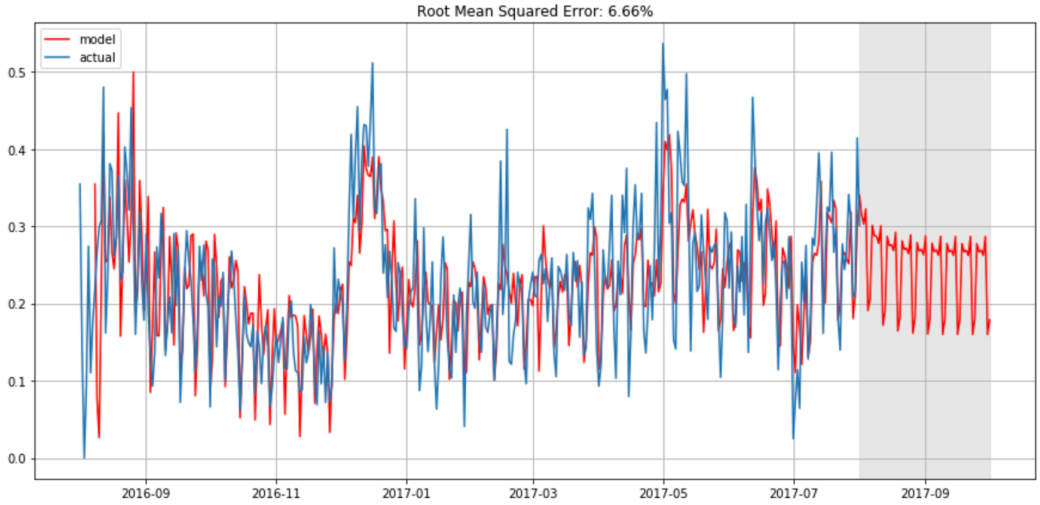
```
                                Statespace Model Results
==========================================================================================
Dep. Variable:                transactionRevenue   No. Observations:                  366
Model:             SARIMAX(1, 0, 1)x(0, 1, 1, 7)   Log Likelihood                 469.512
Date:                           Mon, 22 Jul 2019   AIC                           -931.024
Time:                                   10:10:34   BIC                           -915.491
Sample:                                 08-01-2016 HQIC                          -924.847
                                      - 08-01-2017
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.8742      0.039     22.586      0.000       0.798       0.950
ma.L1         -0.4474      0.068     -6.547      0.000      -0.581      -0.313
ma.S.L7       -0.9497      0.029    -33.085      0.000      -1.006      -0.893
sigma2         0.0041      0.000     14.778      0.000       0.004       0.005
==========================================================================================
Ljung-Box (Q):                       43.92   Jarque-Bera (JB):                 9.77
Prob(Q):                              0.31   Prob(JB):                         0.01
Heteroskedasticity (H):               1.30   Skew:                             0.34
Prob(H) (two-sided):                  0.15   Kurtosis:                         3.42
==========================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

ACF & PACF



It is clear that the residuals are stationary, and there are no apparent autocorrelations. The prediction results on the next 60 days are as followed, RMSE = 6.66%:



3. Conclusion and Discussion

3.1 Regression Models

Clearly XGboost regression model is the most useful for transaction revenue calculation and prediction, because it provides the closest distribution as that of the true response variable and has the lowest RMSE. In the scale of 0 to 99.99, having RMSE = 0.3972 is in a good range. From XGboosting features distribution, it is obvious that the variables related to pageviews contribute the most to models, especially the average number of pageviews by network domain. In this case, the network providers seem to have influence on customer's willingness of purchase, which is beyond expectation.

3.2 Time Series Models

Both approaches of time series modelling methodology, triple exponential smoothing (Holt-Winters) and ARIMA, have successfully captured the main trends but fail to trace the extreme values as predicting daily transaction revenue at G-Store. Both models also have relatively small RMSE's in forecasting two-month future daily revenue. Triple exponential method gives 4.48% and SARIMA has 6.66%. The data records indicate that it is proper for time series analysis.

3.3 Potential Research Objective – Classification

Besides predicting individual transaction revenue by regression and time series forecast on the whole store, classification methods in predicting whether a customer will purchase at G-Store is another division of exploring this data. However, it is necessary to consider the imbalance of this dataset since 98.7% of the records did not purchase. The accuracy rate of

classification models will be high but only due to predicting the majority. Simply randomly sampling the data will not help since the samples are still imbalanced. Two methods can solve this problem:

a. Balanced resample: resample the observations to have purchased and not-purchased customers 1:1.
b. Class weight: Set different weights to two types of behavior.

3.4 Conclusion

Overall, the models' results are satisfying and successfully fulfilled the objectives I hope to achieve. There are also more possibilities for investigation based on this dataset.

Reference

"R EDA for GStore + GLM + KERAS + XGB", kxx, Nov. 19th, 2018, https://www.kaggle.com/kailex/r-eda-forgstore-glm-keras-xgb/code

"Google Analytics Customer Revenue Prediction: Predict how much GStore customers will spend", Sept 4th, 2018, https://www.kaggle.com/c/ga-customer-revenue-prediction

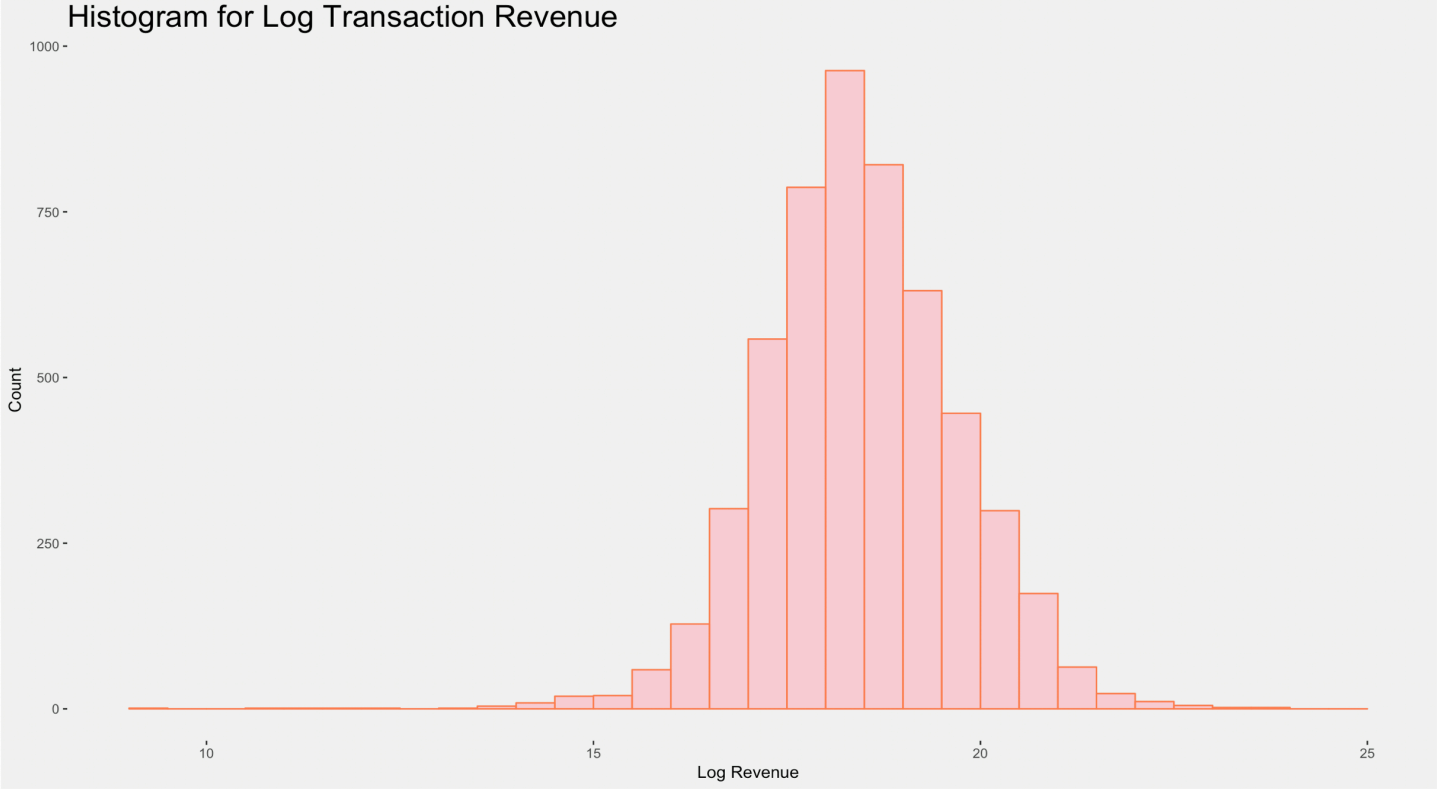
"A comprehensive beginner's guide to create a Time Series Forecast", JAIN Aarshay, Feb 6th, 2018, https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/

"Model Matrices in R", BATES Douglas, Aug. 23rd, 2010, http://www.stat.wisc.edu/~st849-1/Rnotes/ModelMatrices.html

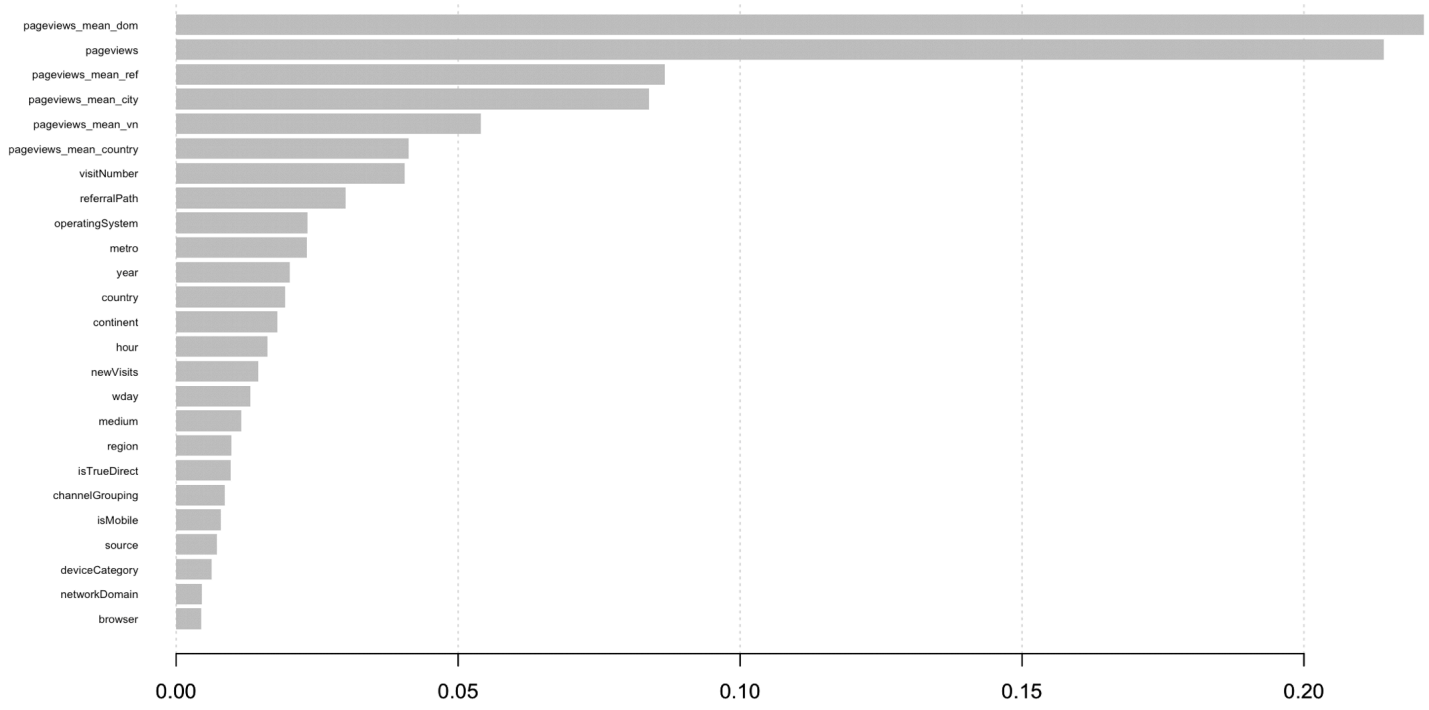
Appendix 1. Variables Descriptions

Variables	Data type	Description	Level
Date	Quantitative	The date on which a customer visited the store	
Full Visitor Id	Categorical	A unique identifier for each user of the store	
Session Id	Categorical	A unique identifier for each visit to the store	
Visit Number	Quantitative	The session number for the user	1-395
Visit Time	POSIXct	The start time of each visit	Eg: "2016-09-02 15:33:05"
Browser	Categorical	The browser a visitor uses	Eg: Chrome, Safari
Operating System	Categorical	The operating system a user accesses the page	Windows; Android; iOS; Macintosh; Chrome OS; Linux
Is Mobile	Categorical	Whether a customer visit from a mobile device	True; False
DeviceCategory	Categorical	The device a visitor uses	desktop; tablet; mobile
Geo Networks	Categorical	Information about geographic location of a visitor. It can be further down into continents, sub continents, countries, regions, metro and cities.	Eg: Europe, Southern Europe, Spain, Madrid.
Network Domain	Categorical	The information of network providers	ex: dodo.net.au
Hits	Quantitative	The hit times of a visitor	1-500
Page Views	Quantitative	Number of pages a visitor views	1-469
Bounces	Quantitative	A *bounce* is a single-page session on the site. Bounce rate is the percentage of all sessions in which users view only a single page and trigger only a single request to the analytics server.	
Transaction Revenue	Quantitative	The revenue made from a single customer visit.	
Source	Categorical	Info from which the session originated	Eg: Google, Baidu
Medium	Categorical	Categorized source path	Eg: organic, referral
Keyword	Character	Keywords a visitor searched in the search engine	Eg: buy google souvenirs
Is True Direct	Categorical	Whether a visit is direct or through referral path.	True; False
Ad content	Categorical	The content visitors click to access G store	Eg: Google Merchandise Collection

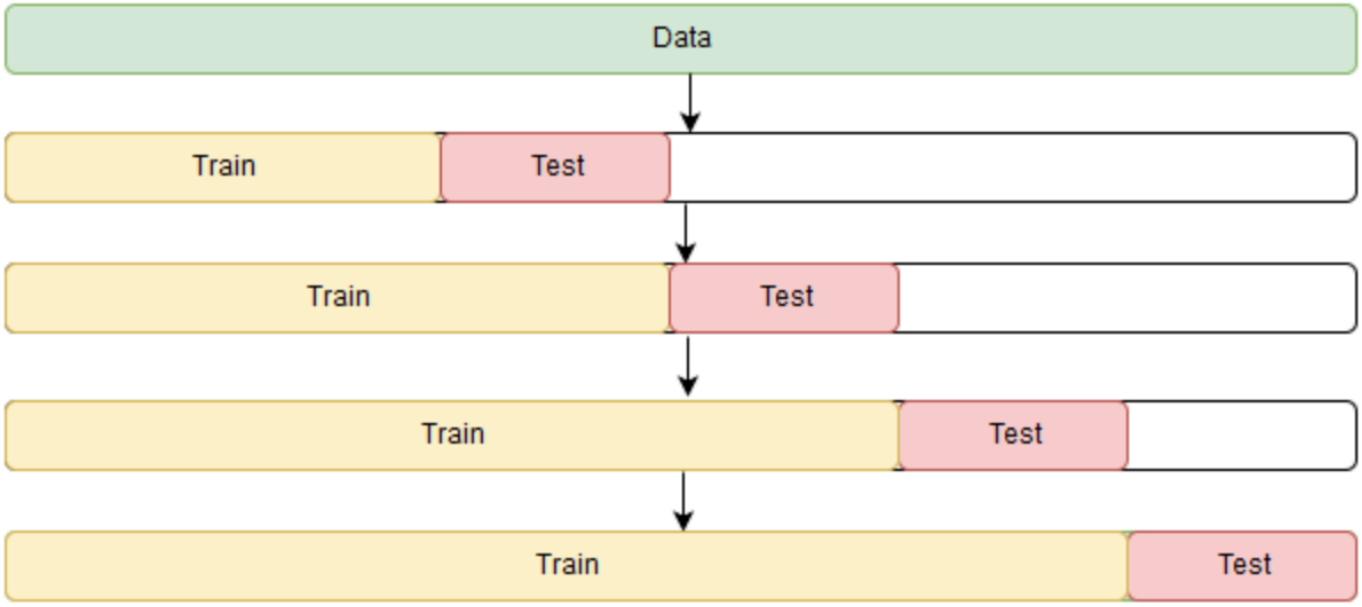
Appendix 2. Distribution of Log Revenue



Appendix 3. Percentage Importance of Features (Xgboost Regression)



Appendix 4. Time Series Chronologically Split



Appendix 5. ARIMA Autotuned Parameters Results (P, D, Q, s)

	parameters	aic
0	(1, 1, 4, 1)	1126.913337
1	(1, 1, 3, 1)	1126.913337
2	(1, 1, 2, 1)	1126.913337
3	(1, 1, 1, 1)	1126.913337
4	(1, 1, 0, 1)	1126.913337
5	(1, 0, 4, 1)	1134.959928
6	(1, 0, 3, 1)	1134.959928
7	(1, 0, 2, 1)	1134.959928
8	(1, 0, 1, 1)	1134.959928
9	(1, 0, 0, 1)	1134.959928

Appendix 6. Best Autotuned Model Results

Statespace Model Results

Dep. Variable:	transactionRevenue	No. Observations:	366
Model:	SARIMAX(1, 0, 1)x(4, 1, 1, 7)	Log Likelihood	470.408
Date:	Mon, 22 Jul 2019	AIC	-924.817
Time:	10:08:39	BIC	-893.750
Sample:	08-01-2016	HQIC	-912.463
	- 08-01-2017		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.8537	0.049	17.468	0.000	0.758	0.949
ma.L1	-0.4125	0.082	-5.047	0.000	-0.573	-0.252
ar.S.L7	0.0589	0.070	0.846	0.398	-0.078	0.195
ar.S.L14	0.0369	0.062	0.600	0.549	-0.084	0.157
ar.S.L21	-0.0150	0.060	-0.251	0.801	-0.132	0.102
ar.S.L28	0.0009	0.061	0.014	0.988	-0.118	0.120
ma.S.L7	-0.9686	0.043	-22.323	0.000	-1.054	-0.884
sigma2	0.0040	0.000	14.614	0.000	0.004	0.005

Ljung-Box (Q):	38.56	Jarque-Bera (JB):	10.35
Prob(Q):	0.54	Prob(JB):	0.01
Heteroskedasticity (H):	1.31	Skew:	0.35
Prob(H) (two-sided):	0.14	Kurtosis:	3.44