Time Series Analysis - Walmart Sales Prediction

Zehui(Bella) Gu, Meiyu(Emily) Li

1 Introduction

Sales prediction is a crucial task for businesses as it allows corporations to set clear goals and to make better business decisions. Specifically in retail industries, sales forecast helps to efficiently allocate resources for future growth and to set up appropriate inventory and service levels, reducing unnecessary waste. Time series models are a widely used method for for sales prediction, as they are well-suited for predicting future values of a series of past sales data that are measured at regular intervals. Therefore, we would like to explore various time series models to make sales prediction of Walmart, the world's largest retail company by revenue. We also aim to evaluate the models based on their accuracy, robustness, and computational efficiency.

There are two main time series models we are using: ARIMA/SARIMA and Gaussian Processes. We use drill-down analysis to analyze aggregate weekly sales per category and per state. For each of per category and per state analysis, We compare ARIMA/SARIMA and GP models' performance on our test data (last 10-week sales) using RMSE as metric to determine which model we would like to use. We find that ARIMA/SARIMA works better both in per category and per state analysis, with normalized RMSE of 0.123 and 0.0829 respectively, smaller than 0.243 and 0.269 for GP models. For state analysis, there is a yearly $(P \approx 52)$ seasonality found in both models. For category analysis, unlike Food or Household sales, Hobby category fails to follow seasonality in both models, thus the prediction is approximating to the mean value.

2 Problem Definition and Algorithm

2.1 Task

For our project, we use a hierarchical sales data from Walmart Daily Sales ranging from Jan 28th, 2011 to Jun 18th, 2016, containing sales information of products in three categories (Food, Hobby, Household) sold in three states (California, Texas, Wisconsin). Given this dataset of past weekly sales as input, we want to build ARIMA/SARIMA and Gaussian Process models to predict aggregate weekly sales on the last 10-week sales for per category and per state analysis. Our output will be the root mean squared error (RMSE) of our predictions on test data.

2.2 Algorithm

ARIMA and SARIMA There are several algorithms we use in this section. The first is augmented Dickey–Fuller(ADF) test which is used to test the null hypothesis that a unit root is present in a time series sample. We uses this test for the original dataset for each category and for each state. Then we decided if we need to do differencing (as well as how many times of differencing) and seasonality after seeing if the P-value is smaller than $\alpha = 0.05$. It turns out only the category Hobby should not be seasonalized, and all state sales and all category sales should be 1st order differencing. Thus, we construct ARIMA for Hobby category and SARIMA for Household, Food category as well as California, Texas, Wisconsin states.

Autoregressive Integrated Moving Average (ARIMA) is used to predict our time series data. For $AR(P)$ a regression model with values X_t until $p-$time, we have the formula

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + w_t$$

where $p =$ the number of lagged observations in the model, w_t is white noise, ϕ are parameters. We decide to choose

For Integrated$I(d)$, we will take the difference with d times until the ADF test gives that the differenced dataset is stationary. For dataset in each state or each category, it gives us 1st order differencing is sufficient. The 1st order differencing is

$$X'_t = X_t - X_{t-1} = (1 - B)X_t$$

where B is the backward operation.

For $MA(q)$, it is a model on forecasting past errors. Here is the formula:

$$X_t = \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q}$$

where $q =$ the number of lags, θ are the parameters.

Since we only constructed the ARIMA model for hobby category, we simply choose the order $p = 4, q = 2$ by looking at the PACF and ACF graphs.

For SARIMA construction on the other two categories (Food and Household) and the three states (CA, TX, WI), we added the parameter

- P: Seasonal autoregressive order
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period.

by conducting a Grid Search around 256 models of different parameter groups and pick the best parameter group with the smallest AIC value.

Gaussian Processes Unlike ARIMA/SARIMA models which assume linear dependencies for predictions, Gaussian Processes use Bayesian inference to make forecasts, which are suitable for datasets that have complex dependencies and complex latent dynamics. As Bishop stated in his work (1) that in a Bayesian model, for a linear regression model of the form $y = w^T \phi(x)$ in which w is a vector of parameter and $\phi(x)$ is a vector of fixed nonlinear basis function, we use past data to construct a prior distribution over function $y(x, w)$. By inputting the training dataset, we can then evaluate the posterior distribution over w with a full Bayesian inference to average over all possible values of a function. Thus, we are able to use the posterior distribution D over regression functions to compute the predictive distribution $p(y_*|x_*, D)$ for new input x_*.

A Gaussian Process is a collection of random variables such that any finite subset has consistent Gaussian multivariate statistics. It can be fully specified by a mean function $m(x)$ and a covariance function $K(x, x')$ and we write $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$:

$f(x_1), ..., f(x_N) \sim \mathcal{N}(\mu, K)$, where $\mu_i = m(x_i)$, $K_{ij} = k(x_i, x_j)$, $x_1 ... x_N$ as inputs

To do forecasting using GP model, we need to choose the appropriate kernel because it determines the overall shape of the function that the model is trying to learn. The main signatures of data we want to capture with our choice of kernels are: $1)$ increasing trend $2)$ some form of periodicity. Thus, two main kernels we are using are exponential squared (RBF) kernel and periodic kernel. Inspired by Rassmussen and Williams' work (2), we construct our kernels and fit our models in two approaches.

Design 1: We break our kernel into four components: $1)$ a long time trend featured by RBF kernel, $2)$ seasonal variations captured by periodic kernel, $3)$ irregularities using RQ kernel, $4)$ some noise simply using white kernel.

Design 2: We are interested in the seasonal variations thereby using a combination of periodic kernels (yearly, monthly, quarterly) with addition and multiplication.

As for the hyper-parameter tuning, we notice that length scale l determines the length of the "wiggles" in our function, which is a free parameter; the periodicity p is the distance between repetitions of the function, and they are fixed with estimates of $52, 12, 4$ representing yearly, quarterly, monthly

Figure 1: Aggregated Weekly Sales Per State and Per Category

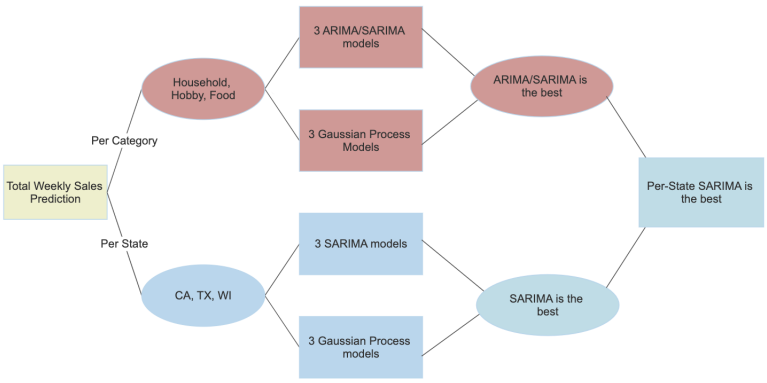


Figure 2: Method to predict the total sales by comparing models in both category and state

seasonality respectively. Finally, we compare the result of RMSE on the test data and the predicted plot to determine the best kernel. We will explore the actual implementation with detailed examples in the next section.

3 Experimental Evaluation

3.1 Data

We are using the dataset of Walmart Daily Sales(3) and there are 30490 pieces of sales information of different goods in three states (California, Texas, Wisconsin) and in three categories (Food, Hobby, Household). There are no missing values in this dataset.

We have 1941 days of sales in this dataset both per category and per state, thus we decide to aggregate the daily sales to weekly sales, which truncates 1941 data points to 278 data points. Since the first week is not a whole week, thus the aggregated weekly sales is very lower than the other sales, which is meaningful to drop this outlier. In total, there are 277 weekly sales for each category and for each state. In Figure 1, we can see Food Category has higher sales than Household and Hobby, and CA state has the highest sales among the three states.

3.2 Methodology

To predict the total sales, we decide to train the first 267 weeks and test the last 10 weeks. As Figure 2 shows, we conducted the Drill-Down analysis per category and per states and compared ARIMA, SARIMA, and Gaussian Process to see which model has the smallest RMSE. After choosing the best model per category and per state for Gaussian Process and ARIMA/SARIMA, we predict the total sales by adding the three categories and the three states up. Then we compare the RMSE and MAE to see whether category drill-down analysis or state drill-down analysis is better and whether ARIMA/SARIMA or Gaussian Process is better in the best Drill-Down Analysis (either category or state).

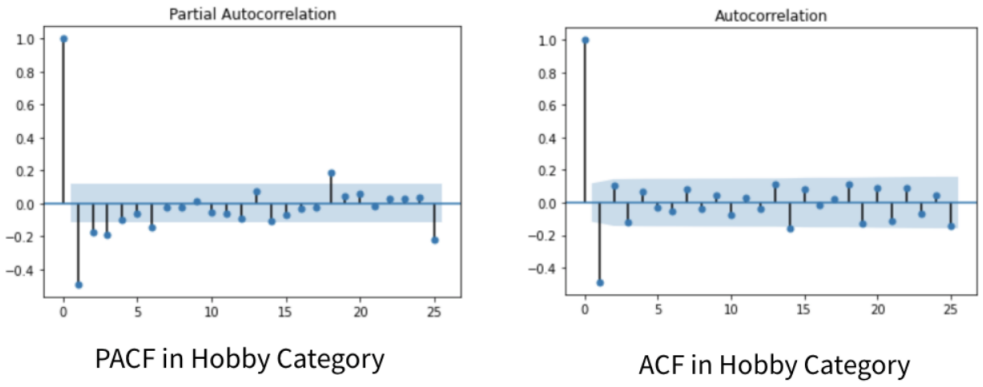


Figure 3: Aggregated Weekly Sales Per State and Per Category

	ADF	ADF after 1st differencing	ADF after seasonalize
Food	0.166	<0.001	<0.001
Hobby	0.527	<0.001	0.0672
Household	0.664	<0.001	<0.001

	ADF	ADF after 1st differencing	ADF after seasonalize
CA	0.281	<0.001	<0.001
TX	0.058	<0.001	<0.001
WI	0.849	<0.001	<0.001

Figure 4: Aggregated Weekly Sales Per State and Per Category

ARIMA and SARIMA As stated in algorithm, we conducted the ADF test in Figure 4 to see if we should difference or seasonlize our dataset by comparing the P-values. As we can see all states and categories should be 1st order differencing because it has the P-values smaller than 0.001 and Hobby does not follow the seasonal pattern. Thus, for Hobby category, we look at the ACF and PACF in Figure 3 and decide to choose ARIMA$(4, 1, 2)$.

For SARIMA, as shown in Figure 5, we construct Grid Search of a range of 256 parameter groups for each state (CA, TX, WI) and each category (Food and Household) and choose the one with the smallest AIC per state and per category.

	Final SARIMA Parameter	AIC	BIC
Food Category	(3, 1, 2) (3, 1, 1, 4)	5736.837	5772.895
Household Category	(4, 1, 2) (0, 1, 2, 4)	5190.344	5222.796
CA State	(1, 1, 0) (3, 1, 1, 12)	5256.916	5278.140
TX State	(3, 1, 2) (0, 1, 1, 12)	4963.094	4987.855
WI State	(3, 1, 2) (1, 0, 0, 4)	5176.640	5201.724

Figure 5: Final SARIMA for Seasonality Pattern

Choice of Kernels	Exact Form	RMSE on Test Data
long_term_trend+ seasonal+ irregularities + noise	RBF(L=255) + RBF(L=0.9)* Period(P=52) + RQ(alpha=1, L=0.99) + white(noise=0.01)	9920.38
kernel_yearly	RBF(L=265) * Period(P=49.3)	7070.43
kernel_yearly + kernel_quarterly	RBF(L=273) * Period(P=49) + RBF(L=0.2) * Period(P=10)	8917.87
kernel_yearly * kernel_quarterly	RBF(L=204) * Period(P=53.2) * RBF(L=204) * Period(P=12.5)	4536.47
kernel_yearly + kernel_monthly	RBF(L=273) * Period(P=49) + RBF(L=0.3) * Period(P=6)	8917.90

Figure 6: Different Choices of Kernels and RMSE on TX State Test Data

Gaussian Processes Following the kernel designs stated in Algorithm Section, we construct our kernels and provide initial estimates of hyper-parameters l and p. Notice that we treat length scale as free parameter and treat periodicity as fixed to ensure the interpretability of our kernels and models. To implement this, we use scikit-learn library in Python, and in its implementation, the "fit" method maximizes the marginal log likelihood of the GP model, which is of great help for hyper-parameter tuning.

Take Texas (TX state)'s data as an example. We use two approaches to design our kernels and try 5 kernels in total as shown in Figure 6. The first kernel is constructed by Design 1, while the other four are constructed by Design 2 using a combination of periodic kernels under the addition and multiplication operation. By computing the RMSE on the test data of last 10-week's sales, we find that the 4th row of this table, a yearly kernel multiplying a quarterly kernel, has a relatively better performance with the smallest RMSE. We repeat the same procedure with other two states and three categories, and the best kernels we could find for each of them are shown in Figure 8, which will be further explained in the Result section.

3.3 Results

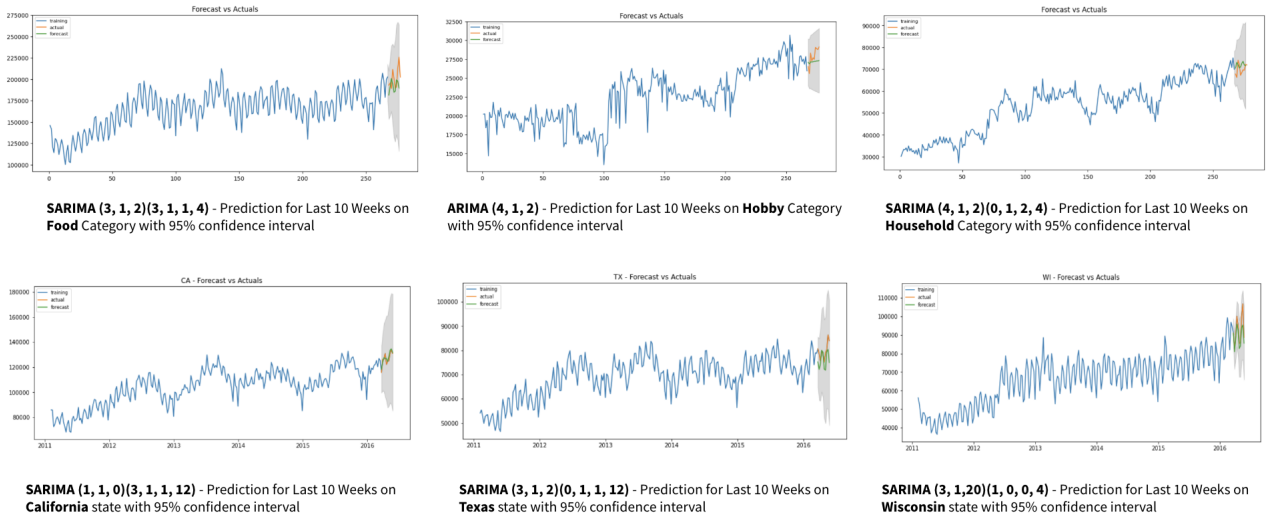
ARIMA and SARIMA We constructed the model with parameters that we explored in Methodology and compare the predicted values with the actual sales for the last 10 weeks. In Figure 7, we can see all models are well fitted into the 95% confidence interval. To be notice, Hobby Category without seasonality tends to predict the last 10 weeks' sales in a flat mean line.

Gaussian Process By constructing our kernels using Design 1 and Design 2, we compare predictions with actual weekly sales on the test data, and choose the kernel that leads to the smallest RMSE. Figure 8 demonstrates the final GP model predictions on last 10-week sales. We can see that most of them show a seasonality of 50, which approximates to one year. The only exception is for Hobby category. Its kernel has large length scale ($L = 278$) and large periodicity ($P = 225$), which is consistent with what we find in ARIMA/SARIMA model that there's no obvious seasonality for Hobby Category.

A further analysis on the hyper-parameters l and p can be shown with the example of TX state in Figure 6. We observe that for Design 1 kernel, although its four components are supposed to capture the signatures of our data, its performance is not the best among our choices. Its seasonal kernel (RBF*periodic) has a very small value of length scale ($L = 0.9$), resulting a more volatile function. Indeed, the fitted GP plot tends to overfit on the train data, leading to a bad performance on test data. For Design 2 kernels, the RMSE outcomes show that multiplication of a yearly kernel ($P = 53.2$) and a quarterly kernel ($P = 12.5$) outperforms than other combinations, such as the addition of two periodic kernels, or simply one periodic kernel.

3.4 Discussion

There are some pros and cons for ARIMA models and Gaussian Process models.



SARIMA (3, 1, 2)(3, 1, 1, 4) - Prediction for Last 10 Weeks on **Food** Category with 95% confidence interval

ARIMA (4, 1, 2) - Prediction for Last 10 Weeks on **Hobby** Category with 95% confidence interval

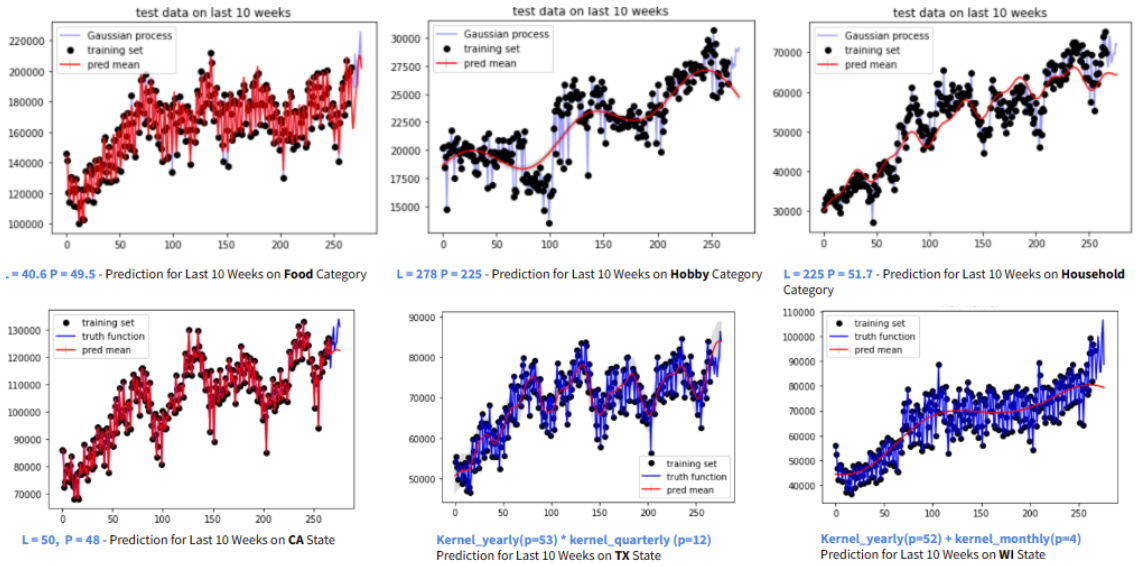
SARIMA (4, 1, 2)(0, 1, 2, 4) - Prediction for Last 10 Weeks on **Household** Category with 95% confidence interval

SARIMA (1, 1, 0)(3, 1, 1, 12) - Prediction for Last 10 Weeks on **California** state with 95% confidence interval

SARIMA (3, 1, 2)(0, 1, 1, 12) - Prediction for Last 10 Weeks on **Texas** state with 95% confidence interval

SARIMA (3, 1,20)(1, 0, 0, 4) - Prediction for Last 10 Weeks on **Wisconsin** state with 95% confidence interval

Figure 7: Final ARIMA and SARIMA Models Predicts the Last 10 Weeks Sales with 95% CI



. = 40.6 P = 49.5 - Prediction for Last 10 Weeks on **Food** Category

L = 278 P = 225 - Prediction for Last 10 Weeks on **Hobby** Category

L = 225 P = 51.7 - Prediction for Last 10 Weeks on **Household** Category

L = 50, P = 48 - Prediction for Last 10 Weeks on **CA** State

Kernel_yearly(p=53) * kernel_quarterly (p=12)
Prediction for Last 10 Weeks on **TX** State

Kernel_yearly(p=52) + kernel_monthly(p=4)
Prediction for Last 10 Weeks on **WI** State

Figure 8: Final Gaussian Process Model Predictions on the Last 10 Weeks Sales

Aggregate Weekly Sales	ARIMA/SARIMA		Gaussian Process	
	RMSE	MAE	RMSE	MAE
Per Category	0.123	0.129	0.243	0.207
Per State	0.0829	0.0702	0.269	0.236

Figure 9: RMSE (normalized by 100,000) on total weekly prediction of last 10 weeks

ARIMA / SARIMA ARIMA and SARIMA works best both in Per state and Per category analysis. The RMSE is 0.123 for category analysis and 0.0829 for state analysis in Figure 9.

Pros

- Only past time series dataset is sufficient to fit ARIMA/SARIMA models
- It is appropriate to predict short-term predictions, so the predictions for the 10 weeks is appropriate

Cons

- It is hard to find some special points, eg. it cannot catch some peak points in Food category
- When predicting long term data points, it will approach to the mean which may not capture the trend
- For Hobby, ARIMA is hard to find the seaonal pattern

Gaussian Processes GP models have higher RMSE than ARIMA/SARIMA models both in per state and per category analysis. The RMSE (divided by 100,000) for category analysis is 0.243, while for state analysis, it's 0.246.

Pros

- GPs are flexible and can be used to model a wide range of functions, and it doesn't assume linear dependencies for datasets.
- As nonparametric models, GPs can be optimized using gradient descent to maximize marginal log likelihood for hyper-parameter tuning.

Cons

- GPs require a good choice of kernel function and initial estimates of its hyper-parameters, which can be difficult to determine in practice, eg. a very small length scale can cause overfitting, leading to a volatile function and bad performance on test data;

One way of resolving the overfitting issue is to enforce to use large length scale by setting its lower bound higher; another potential problem arises as there are some other latent dynamics ignored, such as econmoic conditions, marketing and promotional activities.

4 Conclusions and Future Work

For category, it is hard to find seasonality for Hobby goods sales, because people tend to buy their hobby goods randomly or depending on their wage level. However, Food and Household appears to follow seasonality better in SARIMA than GP-based model.

For states, they all show a yearly ($P \approx 52$) seasonality from both models. For the model performance, each state follows similar results that SARIMA have a better performance than GP models.

We also make some comparisons between these two models. In general within our dataset, we find that for SARIMA, the mean of the forecast is more precise, which follows the true points almost perfectly. For GP models, the choice of kernel function and hyper-parameter tuning have large impacts on prediction accuracy. Sometimes, including seasonality in GP model may not lead to a significant improvement on performance. When sales in certain weeks change greatly, the inclusion of a seasonal kernel with inappropriate hyper-parameter can actually increase prediction error. As for the computational complexity, SARIMA models are easier to implement, while GP models are more time-consuming, with a cost of $O(N^3)$.

For future work, we have a calendar dataset corresponding to festival and holidays. We will implement these special dates into our account to predict sales in special days. For example, customers will tend to buy more in Black Friday, Thanksgiving, Christmas season, etc. The calendar dataset also includes variables called SNAP - Supplement Nutrition Assistance Program provided by US federal government, serves as an incentive to buy more food products; It has a monthly pattern eg. first 10

days for every month for CA state. Thus there may be a high sales in first 10 days for each month in California.

Besides two models we implement for this project, there are some other robust models we could use to make sales prediction using our dataset. To incorporate special day analysis, we can try Facebook Prophet, which is based on a decomposable, additive model that allows it to capture patterns in the data such as trends, seasonality, and holidays/user-defined dates. LSTM networks are effective models as well that use memory cells to retain information from previous time steps, and capture patterns in the data over long period of time.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[3] S. M. v. Addison Howard, inversion, "M5 forecasting - accuracy," 2020. [Online]. Available: https://kaggle.com/competitions/m5-forecasting-accuracy

A Student Contribution

Meiyu(Emily) Li:
Coding: Data Preview, Data Preprocessing for aggregate weekly data per category, ARIMA and SARIMA on per category analysis, Gaussian Process on per category analysis

Writeup: Data Description as well as Algorithm, Methodology, Results, Discussions, Conclusion about the ARIMA and SARIMA part

Zehui (Bella) Gu:
Coding: Data Preprocessing for aggregate weekly data per state, ARIMA/SARIMA on per state analysis, Gausssian Process on per state analysis

Writeup: Introduction, Problem Formulation, GP model part of Algorithm, Methodology, Result, Discussion, Conclusion Section, as well as model comparison.