Problem 1

a.

After brief exploratory analysis, I understood that the stock prices are listed with week-wise data. For each week, stock price is mentioned. It might be an average stock price of the week or the stock price at the day of the week.

When log transformation of data is done the variation or the variance of the data and range decreases.

The intial range of stock price in the training data is around 23 to 199.

When log-transformation is done, the range varies from 3.2 to 5.299.

the log transformation of time was done after converting the data to timestamp values which also exhibited the same trend of decrease in range, variance.

b. Cross-Validation of Time Series data was conducted by using TimeSeries Split of the time series data. Because in TimeSeries Method of sklearn.model_selection, successive training sets are supersets of those that come before them.

c. I used GaussianRegressor to fit the model, which gave very good results of accuracy 0.983.1 or 98.31 initially. To fit the model, I did the log-transformation of the data after converting date strings to timestamp values.

I came across the research paper of prediction of Stock Prices where some kernels are mentioned.

I used the Matern Class Co-variance kernel and Rational Quadratic Co-variance kernels in sklearn as a combination of both.

Mattern Class Co-variance is defined as

$$K(t,t') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}(t-t')}{\rho}\right)^{\nu}$$

Rational Quadratic Co-Variance

$$k_{RQ}(t-t') = \left(1+\frac{(t-t')^2}{2\alpha l^2}\right)^{-\alpha}$$

The k_{RQ} co-variance is the SE co-variance function in the limit $\alpha \to \infty$.

Additionaly, optimizer function can be defined or mentioned.

Also, alpha, the value added to the diagonal of the kernel matrix during

fitting can be mentioned so that calculated values form a positive definite matrix. All the target values can be normalized by using normalizing = true. This is done in case input is Standardized Input.

Problem 2

For the problem 2, data given in data_midterm_problem2.csv. is very variable. The data has values of X_1 to X_{13} defined multiple times for a given number of customers (visitors). Hence, the data cannot be modeled easily. Also the data is very specific, that it gives values of X_1 to X_{13} for 27, 28, 29 and so on for each number of unique visitors. Hence model cannot be built easily on such a data.

I tried to apply linear regression, Ridge Regression, Ridge Classification, Logistic Regression, Random Forest Classifier, AdaBoost Classifier, Gaussian Process Regressor, SVM Classifier, SVM Regression (SVR)

Gaussian Naive Bayesian Classifier.

All the above mentioned models provided results around 10 – 26% accuracy on the training data and 2 – 10 % accuracy on test data which was created from training data provided. Hence I would tell the boss that data is too complex or detailed for to make any inference on the data by any machine learning model so as to predict visitors or make any use of data. I would suggest the boss that data has to made less detailed or complex by providing the features

data for a range of unique visitors like 0–5, 5–10, ... 80–85, 85–90. So that the overall data becomes less complex to make prediction of hourly web traffic for the model to learn and predict the visitors based on the features.

If data has unique row for each number of visitor like 23 etc then also overall data becomes less complex for the model to learn and use the inference to predict the unique visitors based on inference.

Problem 3

Given $y_i = a e^{-be^{-cx_i}}$

$$P(y/x, a, b, c) = P(y_1, y_2, y_3 \ldots y_n)$$

$$= P(Y_1 = y_1, Y_2 = y_2 \ldots Y_n = y_n)$$

$$= P(y_1) \cdot P(y_2) \ldots P(y_n)$$

$\because P(x,y) = P(x) \cdot P(y)$ when x, y are indep variables.

$$P(y/x, a, b, c) = \prod_{i=1}^{N} y_i$$

$$= \prod_{i=1}^{N} \left(a e^{-be^{-cx_i}}\right)$$

b. Optimisation of parameters can be done as.

b. $a, b, c = \text{argmax}_{a,b,c} \left(\prod_{i=1}^{N} y_i\right)$.

This is the estimate for the parameters a, b, c.

c. $\beta \sim p(\cdot)$

$\beta \sim N(\mu, \Sigma)$

$$\beta = \text{argmax}_{\beta} \left(\prod_{i=1}^{N} y_i\right)$$

where

$\beta \sim N(\mu, \Sigma)$ and

where $y_i = ae^{-be^{-cx_i}} = \text{function}(a, b, c)$

d. $\text{Loss Function}(\beta) = \text{argmax}_{\beta} \left(\prod_{i=1}^{N} (y_i)\right) + \lambda \|\beta\|_2$

where β is a function of a, b, c.

and $y_i = ae^{-be^{-cx_i}}$

e. $\text{Loss Function}(\beta) = \sum_{i=1}^{N} (y_i - \beta x_i)^2$

β is a function of a, b, c

and $y_i = ae^{-be^{-cx_i}}$

$\text{Loss Function}(a, b, c) = \sum_{i=1}^{N} (y_i - fn(a,b,c) \cdot x_i)^2$

Another approach,

Based on the graph provided, the model that fits to data can be Logistic Regression Model.

$$P(y|x,a,b,c) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

For logistic regression, we already have Loss Function defined.

a.

$$\langle a,b,c \rangle = \underset{a,b,c}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1+e^{-y_i \beta x_i})$$

$$\beta = fn(a,b,c)$$

b.

$$L(\beta) = \sum_{i=1}^{n} \log(1+e^{-y_i \beta x_i})$$

where $\beta \sim N(\mu, \Sigma)$

$\beta \sim P(\cdot)$.

$\beta = \underset{\text{ction}}{\text{fanc}}(a,b,c)$

c.

$$\underset{\text{Loss Function}}{L(\beta)} = \sum_{i=1}^{n} \log(1+e^{-y_i \beta x_i}) + \lambda \|\beta\|_2$$

where β is a function of a,b,c

and $y_i = a e^{-be^{kx_i}}$

e. $L(\beta) = \Sigma(y_i - \beta x_i)^2$

β = for a, b, c

$y_i = a e^{-be^{-cx_i}}$

Problem 4:-

For the problem 4, initially I tried to reduce the dimensions of the data using PCA, but did not get any output. I reduced dimensions to 2-features, 5-features, 10-features but did not get any fruitful results when I tried to plot the data using matplot lib.

Then I understood the image is hidden in the data and given that the image pixel has to be of form rgb values which range from [0, 255], whatever I did was pointless.

Hence, I converted the existing data using the MinMaxScalar such that each pixel in the data would have a value between 0 to 255.

Then I used imshow method to get the image from the data.

I was astonished to see that I got a blurred image that actually looked like a pattern. I do hope this image is the expected output.

Problem 5 :-

The linear regression model follows :-

$$\beta_{OLS} = \underset{\beta}{\operatorname{argmin}} \sum_i (y_i - \beta^T x_i)^2$$

The Bayesian linear regression model

$$y_i \sim N(\beta^T x_i, \sigma^2)$$

$$\beta \sim P(\cdot)$$

for some prior distribution P.

Max of posteriori for β.

$$\beta_{MAP} = \underset{\beta}{\operatorname{argmax}} \prod_{i=1}^{N} N(y_i | \beta^T x_i, \sigma^2) \; P(\beta | \cdot)$$

For a Bayesian Linear Regression Model,

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Normalization}}$$

or

$$\text{Posterior} \propto \text{Likelihood} * \text{Prior}$$

$$\beta_{MAP} = \frac{\text{Likelihood} * \text{Prior}}{\text{Normalization Constant}}$$

$\therefore$ Posteriori $= \beta_{MAP}$.

$$\beta_{MAP} = \frac{\beta_{MAP} * \text{Prior}}{\text{Normalization Constant}}$$

$\therefore$ In our case $\hat{\beta}_{MAP} = \hat{\beta}_{OLS}$

that likelihood $= \hat{\beta}_{MAP}$

Prior $p(\beta|\circ) =$ Normalization Constant

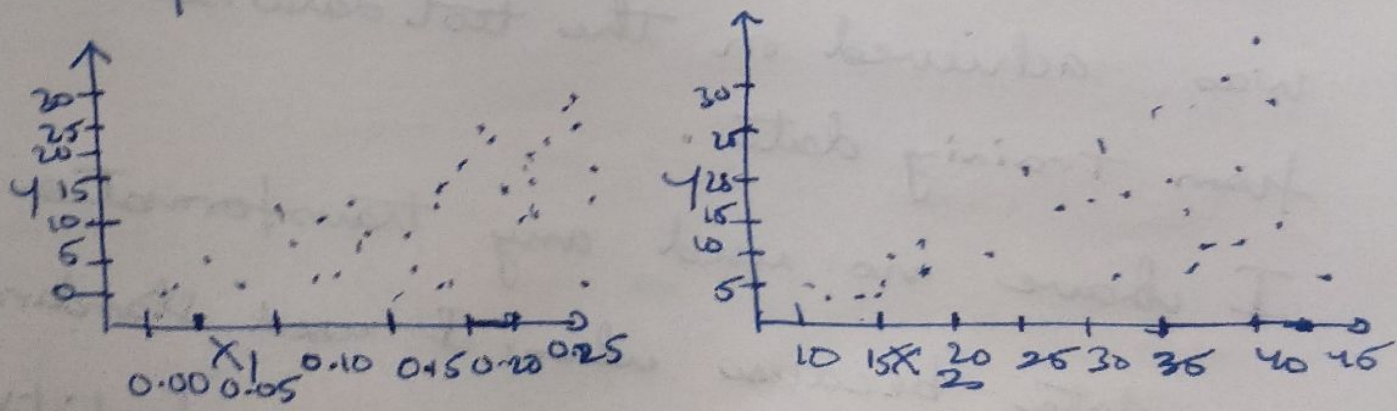
Prior $p(\beta|\circ) \propto 1$

Prior $p(\beta|\circ) \propto c$

Problem 6:-

Based on the data provided in training data I plotted the graph y vs x1 and y vs x2.

Based on the observation of the graph y vs X1 and y and X2. Both of the graphs had points plotted where increase in X1 resulted in average increase in y and increase in X2 resulted in y can be inferred.



Hence based on these graphs, I concluded that, Linear Regress Model would be best fit to the data given.

Since the data so less, that is we have only 25 words in the training data, I decieded to train the model with entire data and get the 'y' values for the test dataset in data_test midterm_problem6.csv.

The accuracy of the model when entire dataset is used is 77.8-1.

When lesser data is used to train the model, 70-1. 77-1. accuracy was achieved on the test data set derived from training data.

I have no used any transformation on data because when I used standard Scalar(), I got negative values as predictions to data of the test data.