ST451 Project: Analysis from a Bayesian's perspective

39290

1 Introduction

Classical machine learning has been applied in most of the areas in the machine learning world, while Bayesian learning can also be powerful since it provides another perspective of observing data and parametrising models. A straightforward application is that it provides a natural and principled way of combining prior information with data, within a solid decision theoretical framework. You can incorporate past information about a parameter and form a prior distribution for future analysis. And parameters become random instead of static under the Bayesian setting, this helps measure the uncertainty of the fitted model and mitigate over-fitting. In this project, I will first conduct EDA(Exploratory Data Analysis)[Tuk+77] for two individual data sets, followed by modelling from both frequentist and Bayesian's perspective and compare the results. Some advanced models such as Gaussian processes [Ras03] will be deployed in the first task, and K-means[HW79], Gaussian Mixture Models, Birch[ZRL96] will be used in the second task.

2 Task Formulation

NASA Airfoil Self-Noise Analysis[1]
There are 6 variables in this data set, the goal of this problem is to predict the Sound Pressure Level given the other 5 features.
Features:

- frequency:Frequency, in Hertz
- angle: Angle of attack(degress)
- chord: Chord length(m)
- Free-stream velocity
- thickness: Suction side displacement thickness

Targets:

- pressure: Sound pressure level

To evaluate the target variable as accurately as possible, we may use full of the features, or we should shrink some of them by adding regularisation term. Also the feature function

$$y = f(\mathbf{x}) + \epsilon$$

is worth exploring in this task. I will first try linear models and then Gaussian processes regression to assess the goodness of fit of models.

Facebook Live Sellers in Thailand[2]
There are 7050 records in the data and each record is featured by 12 attributes. Each instance carries information of sale that is posted on Facebook. By excluding the some trivial attributes such as ID and type, there are 9 integer variables left.

[1] `https://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise`:The NASA data set comprises different size NACA 0012 airfoils at various wind tunnel speeds and angles of attack

[2] `https://archive.ics.uci.edu/ml/datasets/Facebook+Live+Sellers+in+Thailand`:From UCI repository

Since we do not have labels for this task, we are required to explore the underlying clusters each sample belongs to. The task can be modelled as:

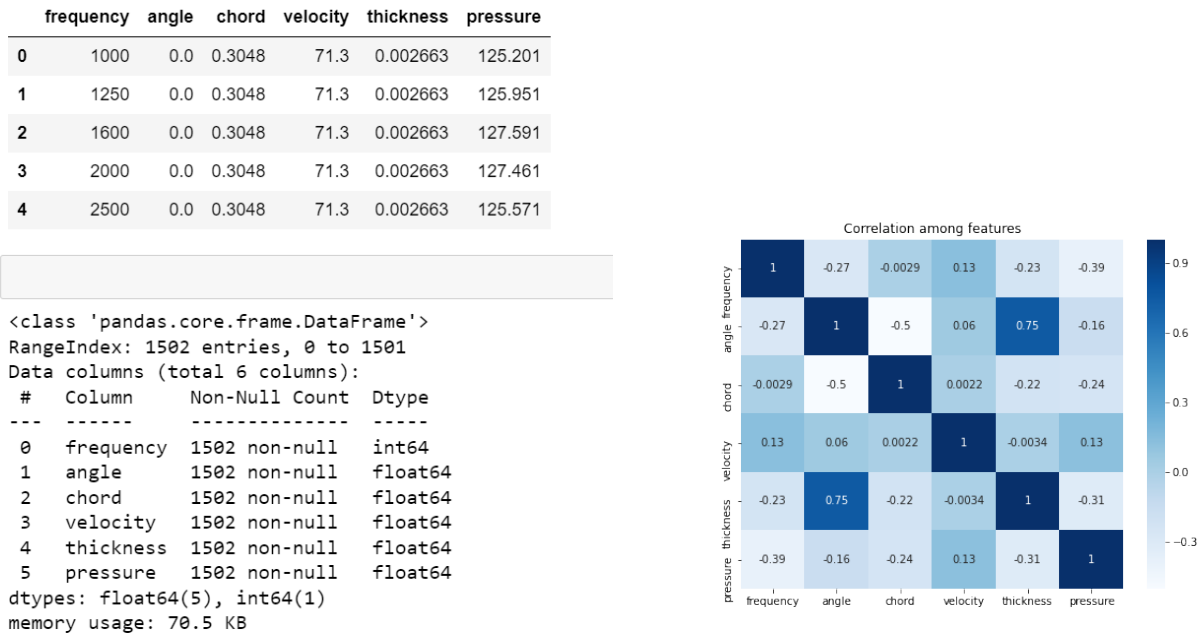
$$f(x_i|\theta_k),\ k = 1, \ldots, K$$

where θ_k denotes the cluster/group k and $f(x_i|\theta_k)$ is the distribution of samples that belong to this cluster. Since the original data usually has high dimensions, dimensionality reduction such as PCA(Principal Component Analysis) will be conducted before the clustering algorithms and is more convenient for visualisation. I will carry out three unsupervised learning algorithms: K-means, GMMs(Gaussian Mixture Models) and Birch here and compare their performances.

3 NASA Airfoil Self-Noise Analysis

3.1 An overview through EDA

Before proceeding to the target prediction, we need to first obtain some basic information about the data. Here I show the first few entries of the data along with the data type of each variable. Since all variables are quantitative, I will also give a correlation map among them.



	frequency	angle	chord	velocity	thickness	pressure
0	1000	0.0	0.3048	71.3	0.002663	125.201
1	1250	0.0	0.3048	71.3	0.002663	125.951
2	1600	0.0	0.3048	71.3	0.002663	127.591
3	2000	0.0	0.3048	71.3	0.002663	127.461
4	2500	0.0	0.3048	71.3	0.002663	125.571

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1502 entries, 0 to 1501
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   frequency  1502 non-null   int64
 1   angle      1502 non-null   float64
 2   chord      1502 non-null   float64
 3   velocity   1502 non-null   float64
 4   thickness  1502 non-null   float64
 5   pressure   1502 non-null   float64
dtypes: float64(5), int64(1)
memory usage: 70.5 KB
```

(a) How data looks like

(b) Correlation among variables

Figure 1: Basic data information

We can find from the Fig1b that the main relation of features with target is **frequency|pressure**, which represents the frequency domain noise spectrum. Thickness seems influential to some extent as well, while velocity and chord are not that correlated to the pressure. Later I will also conduct prediction based on this finding.

Fig2 shows that all features contain lots of localised peaks in the density except pressure, frequency, angle, chord, thickness are all heavily skewed in distribution. The data distribution is imbalanced .We have a lot more data for lower angle as opposed to higher ones, less data for medium level of chord. Whilst angle is probably pre-adjusted, chord & velocity are probably not. In regard of relation to pressure, it seems low frequency is populated where frequency<5kHz. This phenomenon also applies for angle variable. Chord and velocity are kind of random and no clear patterns appear.

The data has skewed features with multiple peaks which suggests nonlinearity. Furthermore, velocity and chord take only a few certain values and do not show underlying pattern, we can then assess the prediction by dividing into different categories of velocity&chord.

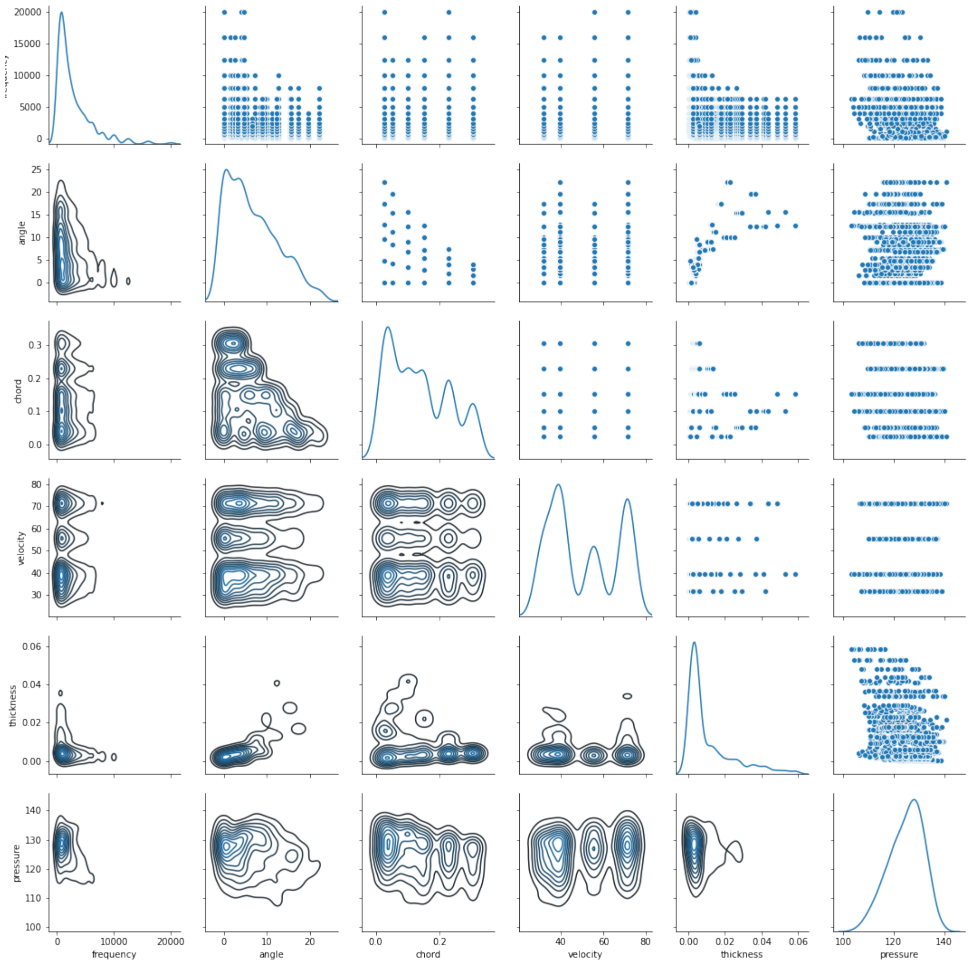


Figure 2: Pair plot of variables

3.2 Prediction of Target

To assess the goodness of fit of models, I will produce plots with real target values as x-axis and prediction values as y-axis, it can then be quite straightforward to observe the performance. Closer the curve is to the line $y = x$, the better model predicts. I will also show the prediction by dividing into various velocity&chord pairs, even if they are quantitative but only a few certain values are exposed in the data.

Ordinary Linear Regression Suppose the features have linear relationship with the target and are determined by some parameters β, and we obtain the optimal parameters by minimising the residual sum of squares(This also maximises the likelihood):

$$Y = X\beta + \epsilon$$

$$\beta_{MLE} = (X^T X)^{-1} X^T Y$$

$$Y_{test} = X_{test}\beta_{MLE}$$

Bayesian Ridge Regression Instead of outputting the target as a certain value, Bayesian regression allows to produce prediction from a distribution.

$$Y_{test} \sim N(Y^*, \Sigma_Y)$$

The model will return the mean of prediction along with its covariance matrix, here I will use the mean of each sample prediction to draw the plot.

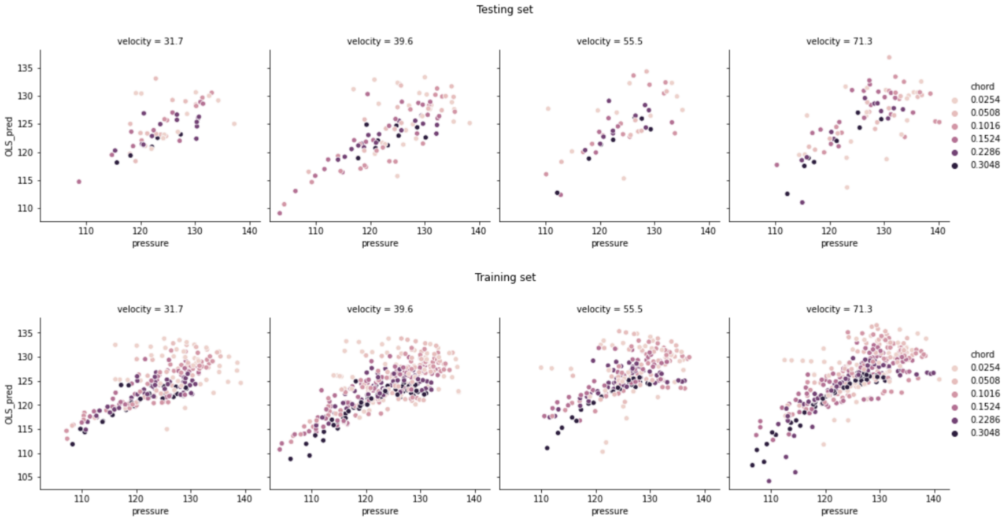


Figure 3: OLS prediction, the model seems to perform similarly on both training and testing sets. More accurate predictions are spotted on samples with larger chord value, while velocity does not show observable pattern.

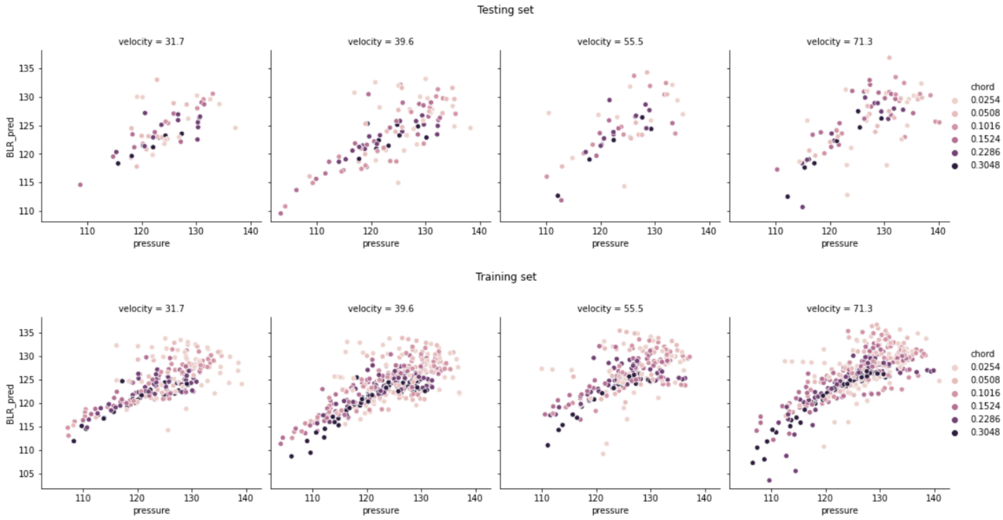


Figure 4: BRR prediction, the prediction of Y^* is almost identical to the ones in the OLS

Gaussian Processes Regression For a regression problem

$$Y = f(X) + \epsilon$$

Gaussian Processes assume the function space are normally distributed:

$$\pi(f) \sim GP(f_0, K)$$

where f is a Gaussian process with mean f_0 and covariance kernel K, then

$$[f(x_1), \ldots, f(x_n)]^T \sim N(f_0, K)$$

$f_0 = [f_0(x_1), \ldots, f_0(x_n)]$, K is the n x n matrix with elements $K(x_i, x_j)$. Intuitively, we do not want to find out what $f(X)$ is, instead the kernel matrix determines. Reversely, we can build the appropriate kernels first and then do the prediction without knowing exact $f(X)$. And after observing the data, the covariance kernel will be updated accordingly based on the Bayes theorem. It is also very important for Gaussian processes to select hyperparameters but I do not go details here.

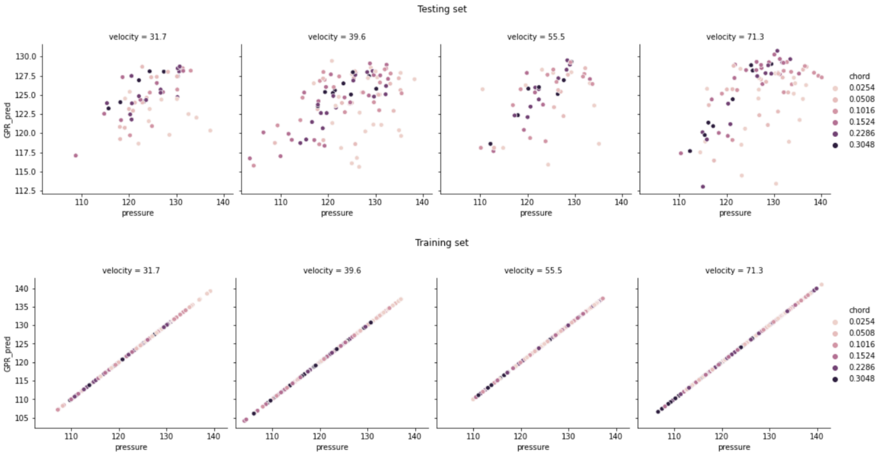


Figure 5: GPR with RationalQuadratic kernel, the model overfits: it perform perfectly on training data but poorly on testing data. The similar result is obtained here that prediction on samples with larger chord values are more accurate.

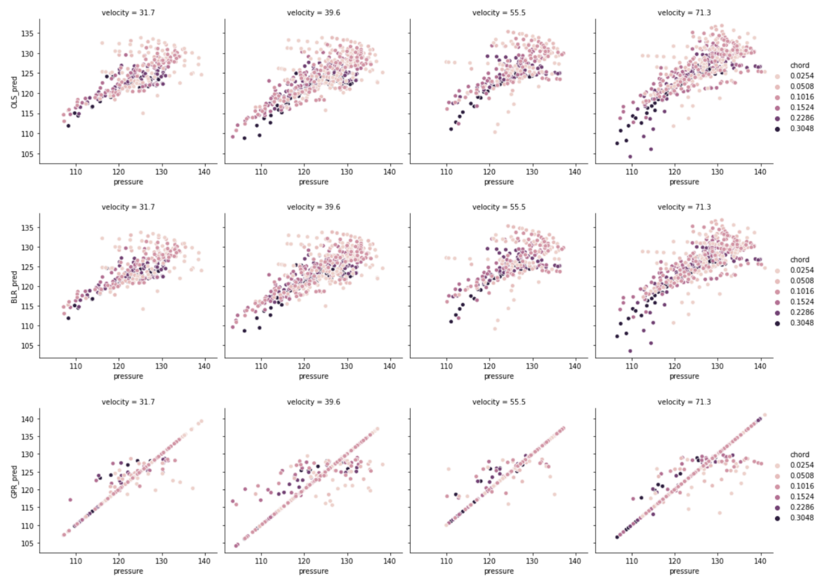


Figure 6: Prediction of three models on full data set

Conclusion

- The two linear models do not distinguish the other, while Bayesian model is from a probabilistic perspective. If we take the mode of Bayesian model it is almost identical to the MLE estimate from Ordinary linear regression.
- Overfitting of Gaussian Processes Regression may be caused by various reasons, since it is sensitive to the data and the kernel. The data set using here is not large enough for Gaussian Processes.
- All three models tend to perform a lot better on data points with larger chord values.

4 Facebook Live Sellers in Thailand

4.1 Data overview and cleaning

Observe from Fig7a, there are 10 attributes in the original data frame, with 9 integer-based variables and 1 categorical variable. It is not hard to find that:

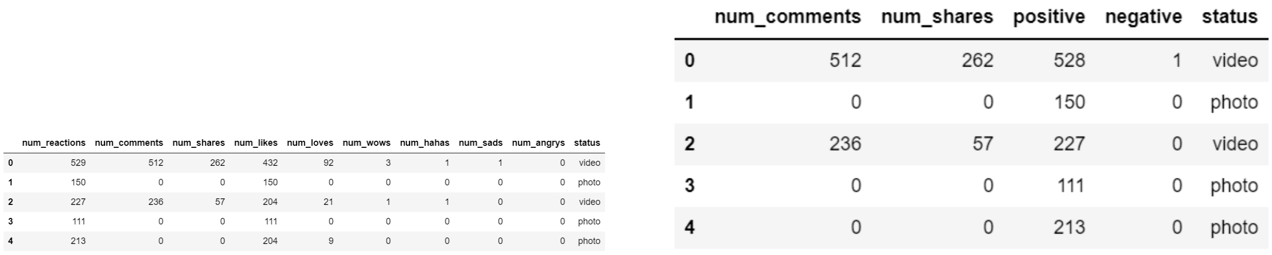
$$\#reactions = \#likes + \#loves + \#wows + \#hahas + \#sads + \#angrys$$

To avoid overlapping, we can then create another two variables based on the sentiment of reactions:

$$\#positive = \#likes + \#loves + \#wows + \#hahas$$

$$\#negative = \#sads + \#angrys$$

Although the data is cleaned and rearanged, there is also one categorical variable **status**, the next step is to check whether it is important for later clustering.



	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys	status
0	529	512	262	432	92	3	1	1	0	video
1	150	0	0	150	0	0	0	0	0	photo
2	227	236	57	204	21	1	1	0	0	video
3	111	0	0	111	0	0	0	0	0	photo
4	213	0	0	204	9	0	0	0	0	photo

(a) Original data

	num_comments	num_shares	positive	negative	status
0	512	262	528	1	video
1	0	0	150	0	photo
2	236	57	227	0	video
3	0	0	111	0	photo
4	0	0	213	0	photo

(b) Cleaned data

Figure 7: Basic data information

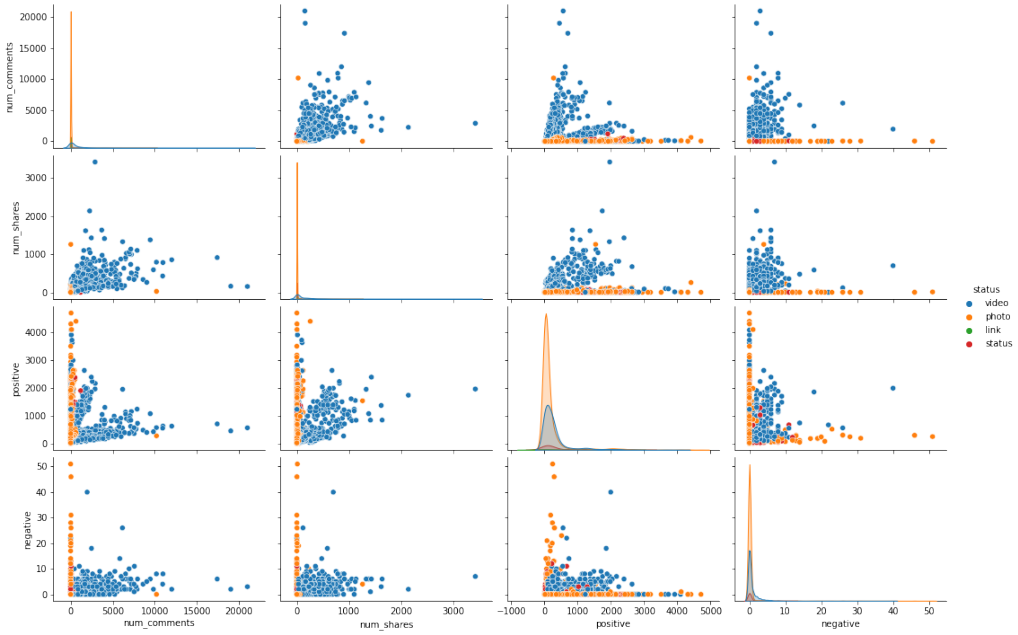
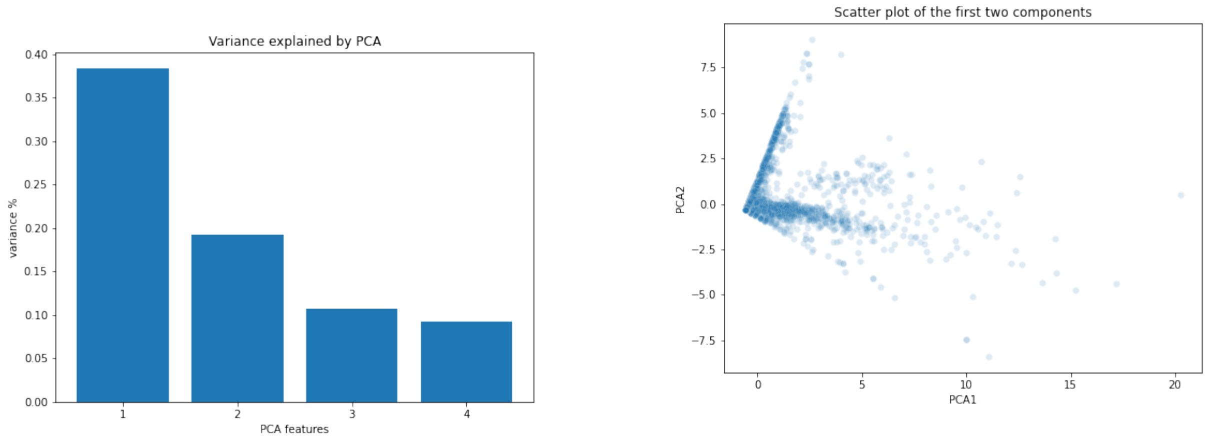


Figure 8: Pair plot of the data

Similarly to what I have done in the first task, here I produce a pair-wise relation map among features, categorised by **status**. From the above plot status has no direct relation to segmenting reactions, we can drop it and move on with other features.

4.2 Dimensionality reduction

Because the original data has 4 features, I will first perform dimensionality reduction technique by PCA(Principal Component Analysis) to capture the largest amount of variance of the data, it is also more feasible for visualisation. Fig9a shows that the first two components explain the majority of the variance in our data. Hence I will use two principal components for visualisation. As the two principal components shown in Fig9b which indicates that there are at least two distinguishable clusters. This factoid tells us that the observations in the dataset can be grouped.



(a) Variance explanation of PCA

(b) Distribution of data on the first two principal components

Figure 9: PCA analysis

4.3 Clustering

4.3.1 K-means

K-means is vastly applied in many clustering problems and prove its usefulness. this method aims at partitioning n observations into K clusters in which each observation belongs to the cluster based on the distance. Before implementing this algorithm, we need to decide how many clusters that will be used. There are two popular methods for selecting the number of clusters.

Elbow method

The Elbow method focus on the inertia as a function of the number of clusters. The location of an Elbow is usually considered as an indicator of the appropriate number of clusters because it means that adding another cluster does not improve much better the partition.

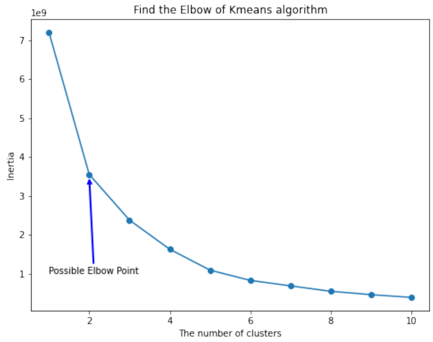


Figure 10: Elbow method, which suggests 2 clusters

Silhouette method The Elbow method is not always clear and sometimes it is hard to identify the location of the Elbow. Silhouette method offers as an alternative. This method measures the quality of a clustering and determines how well each point lies within its cluster.

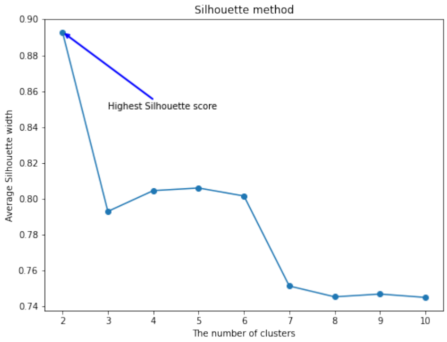


Figure 11: Silhouette method suggests 2 clusters, which has the highest average Silhouette value

Visualisation

Based on previous analysis, I will conduct the K-means algorithm with 2 clusters. Here two principal components are used as x-y axes, the third feature is the label of which cluster each sample belongs to. Now compare with the Fig8, it looks very similar to the **Positive vs Comments** plot, we can hence interpret the clusters as:

- Label 0: Positive trend, fewer comments
- Label 1: Negative trend, more comments

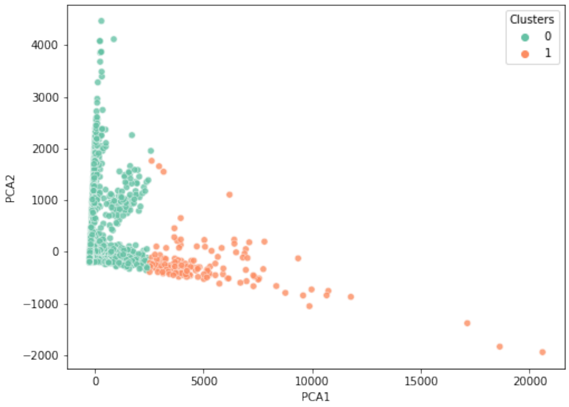
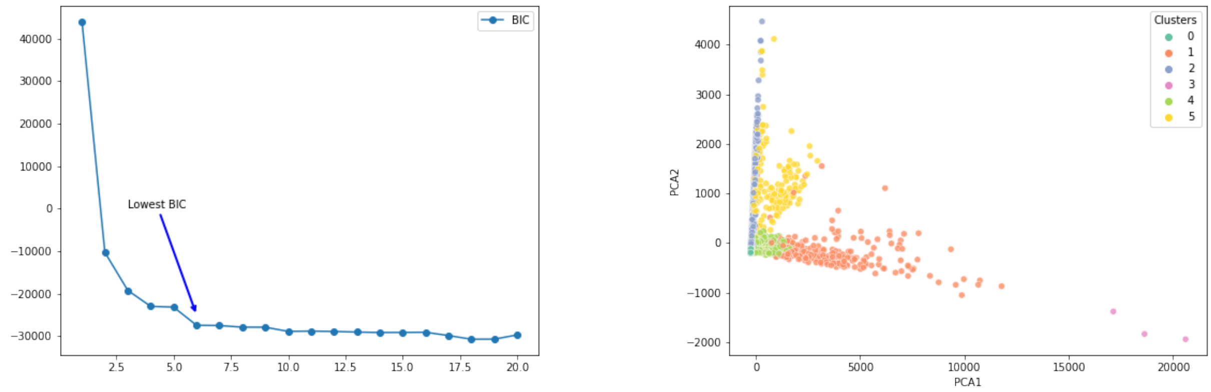


Figure 12: K-means clustering visualisation

4.3.2 GMMs(Gaussian Mixture Models)

Different from K-means which is distance-based, GMMs serves as a distribution-based model. It assumes that there are a certain number of Gaussian distributions, and each of them represent a cluster. Therefore, a GMM tends to divide the data points following the same distribution together. Instead of classifying samples to certain groups deterministically, Gaussian Mixture Models use soft allocation of individuals to clusters, which means each sample is assigned a K-sized vector of probabilities. The i-th element in the vector represents the probability of belonging to the i-th cluster.

Similarly to K-means, GMMs also need to specify the number of clusters in advance, BIC is usually applied to find the optimal number of clusters. Fig13a indicates 6 clusters for the GMM, while Fig13 shows how samples are grouped. It seems GMM over-fits the data for some extent.



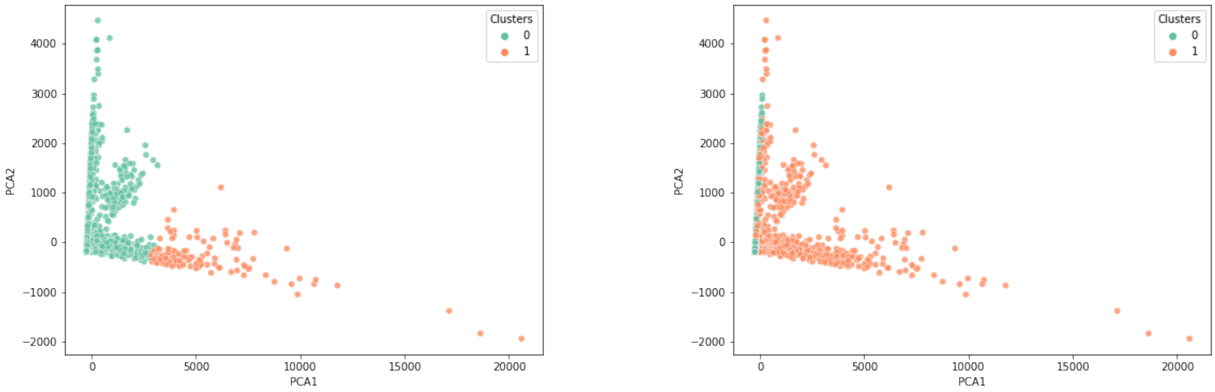
(a) BIC against the number of clusters

(b) GMMs with suggested number of clusters

Figure 13: GMMs model

4.3.3 Birch(Balanced Iterative Reducing and Clustering hierarchies)

Unlike algorithms such as K-means which is not feasible for large scale data set, when conducting clustering on large data sets, BIRCH proves it to be very efficient. Moreover, BIRCH is very useful because of its easy implementation. This algorithm is based on the clustering features tree. Additionally, this algorithm applies a tree-structured summary to create clusters. Here I give a plot of using Birch model with 2 clusters(same as suggested in K-means) to group the data, we can find that it performs slightly better than K-means model, there is a clear boundary between the two clusters. Meanwhile, a GMM with 2 clusters is also given here for comparison: It does not give as good results as the other two models.



(a) Birch with 2 clusters

(b) GMMs with 2 clusters

Figure 14: Comparison of two models

Conclusion:

- 2 clusters seem the optimal choice for this task. K-means and Birch have similar performance on the problem while GMM has poor results.
- Comments to positive responses are much higher than comments to negative.

References

[Tuk+77] John W Tukey et al. *Exploratory data analysis*. Vol. 2. Reading, MA, 1977.

[HW79] John A Hartigan and Manchek A Wong. “Algorithm AS 136: A k-means clustering algorithm”. In: *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[ZRL96] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. “BIRCH: an efficient data clustering method for very large databases”. In: *SIGMOD ’96*. 1996.

[Ras03] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Summer school on machine learning*. Springer. 2003, pp. 63–71.