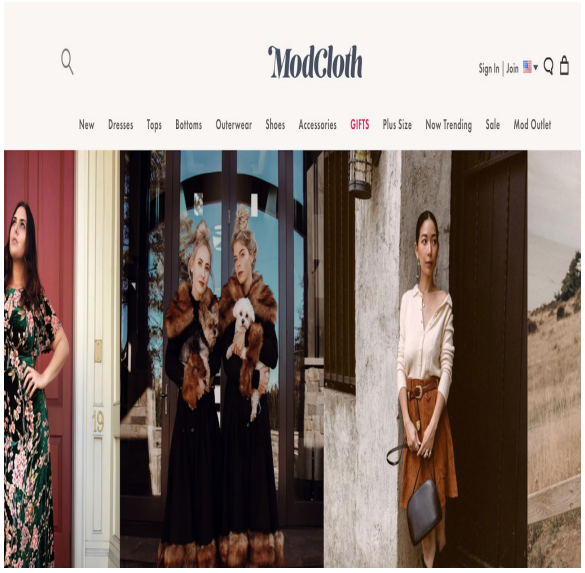
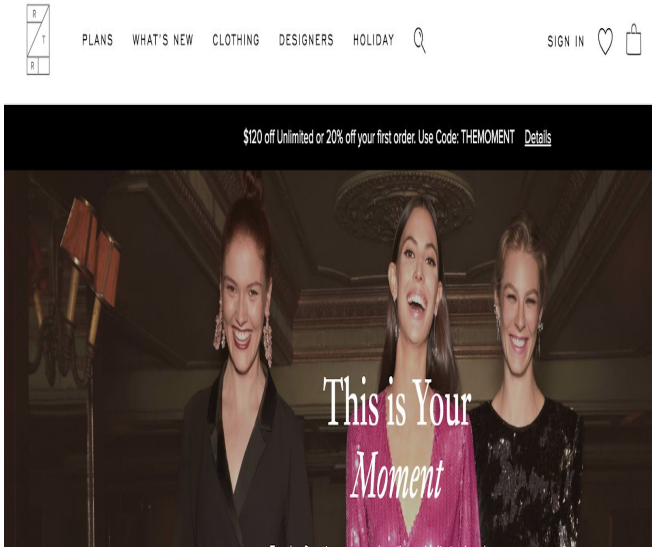
Clothing Fit





TEAM: ANALYTICS TRIAD

Members:

1. Priyanka Raju (013839865)
2. Ashna Gupta (013742040)
3. Kesha Shah (013534352)

Github Link: https://github.com/Kesha0796/AnalyticsTriad

Chapter 1: Introduction

In this digital era, with the growth of online fashion industry and varying availability of sizes, it becomes difficult for the customers to figure out their appropriate sizes. Due to which there might be the possibility of placing a wrong sized order and which will in turn lead to the process of returns or exchanges of the purchased item. It also becomes difficult for the retailers and stores to handle return or exchange requests of all the customers. This problem can be addressed by building a machine learning model which could recommend the size to the customers based on their preferences, their order history or according to similarity of their physical measurements with the other users.

Users are generally asked to enter their feedback in the form of details like their height, weight, body type such as athlete, straight and narrow and categories of the item, cup size, hips, bust size and many more. This history of details can in turn be used for predicting the fit of the customer and then further recommend the size to the users. Also based on the category or type of cloth they want to purchase they would be suggested size. As people prefer to order dress with small size, while jumpsuit of large size. Also if user has no past history of buying clothes from that specific website than size would be recommended on the basis of similarity of body physique with other customers.

For our project, we are using 2 dataset called ModCloth and RenttheRunWay which are collected as a feedback from the customers of 2 different shopping websites. Our goal is to merge both the datasets so as to obtain varied features from both the set and perform preprocessing on data such as handle missing values, removing duplicates and incorrect data, etc. After preprocessing, different classification algorithms are applied used to train dataset and build a model to predict fit i.e. "fit","small" or "large" for the customers. Based on the results obtained after fit prediction, using this information we can infer size the size of the customers.

Chapter 2: System design and Implementation

1. **Algorithms Used:**
   Fit Prediction:

   1. Logistic Regression: As the goal of this problem is to predict fit i.e. small, fit or large which is classifier problem,we have used Logistic regression which provides very efficient solution to categorical problems. Logistic Regression assigns observations to discrete set of classes and uses sigmoid function to return probability value. We have used "lbfgs" and "saga" solver for Parameter tuning."Saga" gives efficient results for large datasets and multinomial class.

   2. Naive Bayes: Naive Bayes Classifier algorithm is used for predicting fit, small and large for our problem statement. This algorithm is based on the probabilistic model which is then used for classification. It Converges quickly compared to other algorithms hence its performance compared to other algorithms are quite fast and easier.There are three types of Naive Bayes models present which are Gaussian, Multinomial and Bernoulli.

Gaussian model deals with data containing features in continuous form, Multinomial model deals with data having discrete values and Bernoulli deals with data having binary or boolean values. Since our data set mostly contains features which are continuous so we have used Gaussian Naive Bayes Model.

3. Large Margin Nearest Neighbor: Large Margin Nearest Neighbor is an extension of K-NN were the similarities between the neighbors are calculated based on the mahalanobis distance metrics. This algorithm creates a larger margin between the items that are similar to those that are not similar. This technique supposedly performs better in comparison to that of the default K-NN. On applying LMNN to the training set, we are projecting the points to newer space/dimensions(which is called as metric learning) and the transformed data is then applied onto to any classification model to obtain better accuracy and prediction. For this problem statement of predicting the fit/size, it seems appropriate to apply this algorithm and evaluate its outcome.

Size Inference:

1. K-Nearest Neighbor: To infer the size of a customer, we can do a decision based prediction for a particular instance(user) or similarity based prediction. We have chosen to start with a simple approach using the K-NN based on the analysis on our dataset where the size inference can be obtained based on the fit feedback of other customers. In order to implement this idea, we will be using only the data of the customers whose size is "fit" to train the model, which could help in determining the size for those instances which are "large" or "small".

2. **Technologies and Tools:** We have use Python Language to implement our project. The tools used in this aspect are Jupyter Notebook, Google Colab and libraries used are SkLearn, Pandas, Numpy, Catergorical_Encoders.

3. **System design and architecture:**

Dataset Merging: Have merged the two datasets to gives us more insights on the varying features of the customers. The columns with same attributes are scaled to the match and different features are added as new columns.

ModCloth Data

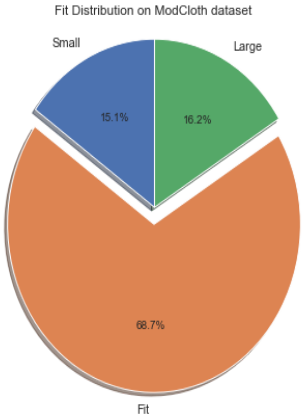
	Count of missing values	Percentage of missing values	Data Type
bra size	6018	7.27	float64
bust	70936	85.68	object
category	0	0.00	object
cup size	6255	7.56	object
fit	0	0.00	object
height	1107	1.34	object
hips	26726	32.28	float64
item_id	0	0.00	int64
length	35	0.04	object
quality	68	0.08	float64
review_summary	6725	8.12	object
review_text	6725	8.12	object
shoe size	54875	66.28	float64
shoe width	64183	77.53	object
size	0	0.00	int64
user_id	0	0.00	int64
user_name	0	0.00	object
waist	79908	96.52	float64
fit_to_numeric	0	0.00	int64
height_cms	1107	1.34	float64

RenttheRunway Data

	Count of missing values	Percentage of missing values	Data Type
age	960	0.50	float64
body type	14637	7.60	object
bust size	18411	9.56	object
category	0	0.00	object
fit	0	0.00	object
height	677	0.35	object
item_id	0	0.00	int64
rating	82	0.04	float64
rented for	10	0.01	object
review_date	0	0.00	object
review_summary	0	0.00	object
review_text	0	0.00	object
size	0	0.00	int64
user_id	0	0.00	int64
weight	29982	15.57	object

Merged Data

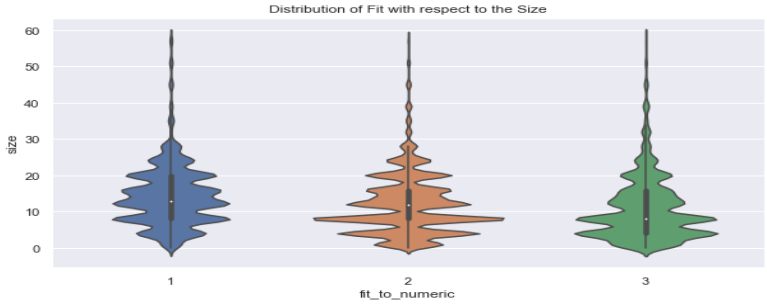
	Count	Percentage	Data Type
bra size	198562	72.12	float64
bust	263480	95.69	object
category	0	0.00	object
cup size	198799	72.20	object
fit	0	0.00	object
height	1784	0.65	float64
hips	219270	79.64	float64
item_id	0	0.00	int64
length	192579	69.94	object
quality	150	0.05	float64
review_summary	6725	2.44	object
review_text	6725	2.44	object
shoe size	247419	89.86	float64
shoe width	256727	93.24	object
size	0	0.00	int64
user_id	0	0.00	int64
user_name	192544	69.93	object
waist	272452	98.95	float64
age	83750	30.42	float64
body type	97427	35.39	object
bust size	101201	36.76	object
rented for	82800	30.07	object
review_date	82790	30.07	object
weight	112772	40.96	object



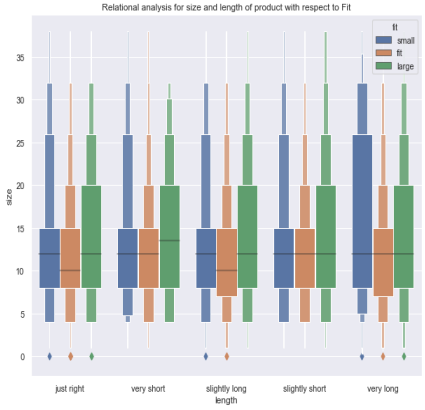
4. **Visualization:** The tools used for exploratory analysis are Matplotlib and Seaborn.

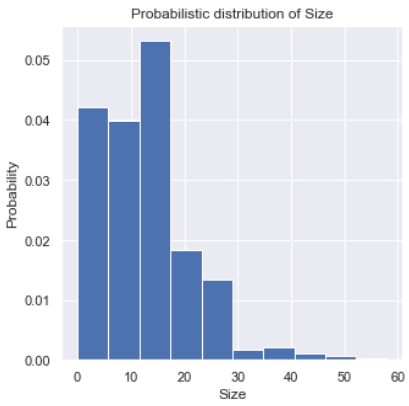
Overall distribution of the target attribute 'Fit' on one of the datasets: It shows from the chart that 70% of the data consists of the fit feedback to be of 'Fit' class.(Pie Chart shown above)

Relational analysis between the features 'Fit' and 'Size': We can analyse from the below distribution that the fit feedback provided by the customers have mostly purchased for varying sizes between 0 to 30.



Analysis of size and length attribute with respect to 'Fit' & Probabilistic distribution of 'size' on overall dataset(Renttherunway):





Feature Correlation Map for ModCloth data

Chapter 3: Experiments & Proof of concept Evaluation

1) Dataset

We have used the dataset of ModCloth.com and RentTheRunWay.com. The datasets provide information on the customer and item measurements, their fit feedback, categories of the items purchased, ratings on the quality of the product and reviews given by the customers.
The statistics of the dataset is represented is tabulated as:

	Modcloth	Renttherunway
Number of Users	47,958	105,508
Number of Items	1,378	5,850
Total	82,790	192,544
Number of Features	18	15

The two datasets with their respective features and their descriptions are as follows:

a) ModCloth: modcloth_final_data.json (Size 40.6 MB):

Feature Description:

item_id: unique product id, waist: waist measurement of customer, size: the standardized size of the product, quality: rating for the product, cup size: cup size measurement of customer, hips: hip measurement of customer, bra size: bra size of customer, category: the category of the product, bust: bust measurement of customer, height: height of the customer, length: feedback on the length of the product, fit: fit feedback, user_id: a unique id for the customer, shoe size: shoe size of the customer, shoe width: shoe width of the customer, review_text: review of customer,

review_summary: review summary

b) RentTheRunWay: renttherunway_final_data.json (123 MB)
Feature Description:
item_id: unique product id, weight: weight measurement of customer, rented for: purpose clothing was rented for, body type: body type of customer, review_text: review given by the customer, review_summary: summary of the review, size: the standardized size of the product, rating: rating for the product, age: age of the customer, category: the category of the product, bust size: bust measurement of customer, height: height of the customer, fit: fit feedback, user_id: a unique id for the customer, review_date: date of review.

2) Data Preprocessing

Our chosen dataset consists of attribute types comprising both numerical and categorical data types. The data preprocessing steps involves cleaning of data, With this kind of combinatory data types, requires a good amount of data preprocessing for the given purpose and analysis as below:

a) Data exploration/visualization:
- i) Handling of missing values by removing those instances for the purpose of exploratory analysis.
- ii) Recoding of existing features. Ex. 'height' attribute is converted to centimeters from feet(inches), 'weight' is modified to be represented as numeric in terms of 'lbs' along with modifying the data type to float.
- iii) Target variable 'Fit' is represented as class(categorical) and a new feature is created which holds the numerical representation of 'Fit'.

b) Merging of dataset:
- i) Handling of missing values by imputing with the mean/median for some of the attributes like age/height/weight.
- ii) Rating attribute of Renttherunner is represented in the scale of 1 to 10. Modcloth quality/rating is represented in the scale of 1 to 5. So when merging the datasets, one of the rating scale needs to be scaled with respect to the other dataset.
- iii) Some of the attributes which have 90+ % of missing values are dropped/removed from the dataset. Ex. 'bust','waist'.

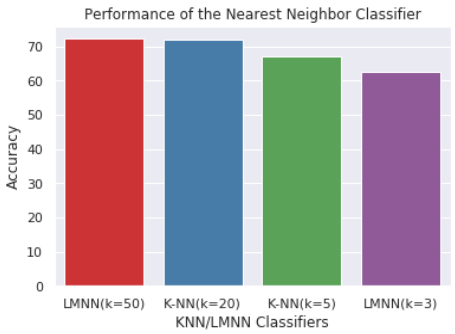
c) Prediction of Fit and Inferring size:
- i) Converting categorical features to numerical using different types of encoders based on the algorithms used to build the model on the merged dataset.

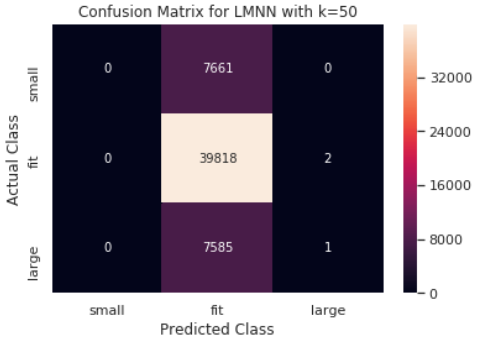
3) Methodology

We are following the standard pipeline of Data mining/ Machine Learning methodologies where the merged and preprocessed dataset is split into train and test set with sampling in the ratio of 80% on the train set and 20% on the test set.
Hyper parameter tuning of the algorithm is implemented using the GridSearchCV along with cross validation technique provided by the sklearn library which performs a brute force approach in giving the best parameters.
Once the training model is complete, this learned model is then applied onto the test set to make the necessary predictions. Evaluation in terms of precision and accuracy is measured.

4) Algorithm Evaluation and Analysis

a) **Large Margin Nearest Neighbor**: With parameter tuning of k= 50(for both LMNN and KNN) resulted in an **accuracy of 72.31%.** Implemented the following algorithms with respective hyperparameter tuning:
1. Baseline KNN model with n_neighbors=5 (accuracy = 67.03 %)
2. KNN with n_neigbors=20 (accuracy = 67.03 %)
3. Default LMNN with n_neigbors=3 (accuracy=72.09%))
4. LMNN with n_neighbors=50 (accuracy=72.31%)





Classification Report for the best model:

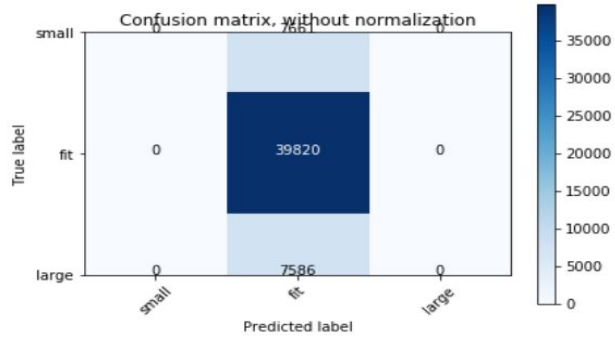
```
Classification Report
              precision    recall  f1-score   support

       small       0.00      0.00      0.00      7661
         fit       0.72      1.00      0.84     39820
       large       0.33      0.00      0.00      7586

    accuracy                           0.72     55067
```

b) **Logistic Regression Model:** This results are for solver "saga" and max_iter value =2000. By default iter value is 100, but as this is huge dataset, we needed to increase the iter value to 2000 for accurate results. The accuracy for Logistic Model with "saga" is 0.72311. We also used solver as "lbfgs" by which we were able to obtain accuracy of 0.73221.

Confusion Matrix:



Classification Report for Logistic Model:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	7661
2	0.72	1.00	0.84	39820
3	0.00	0.00	0.00	7586
accuracy			0.72	55067
macro avg	0.24	0.33	0.28	55067
weighted avg	0.52	0.72	0.61	55067

c) <u>Naive Bayes Algorithm Model:</u>

<u>Gaussian Naive Bayes Algorithm</u>: There are two parameters present in this model priors which specifies the prior probabilities of class whose default values are None denoting it sets the probabilities in accordance of data. The other parameter that needs to be set is var_smoothing whose default value is set to 1e-09.

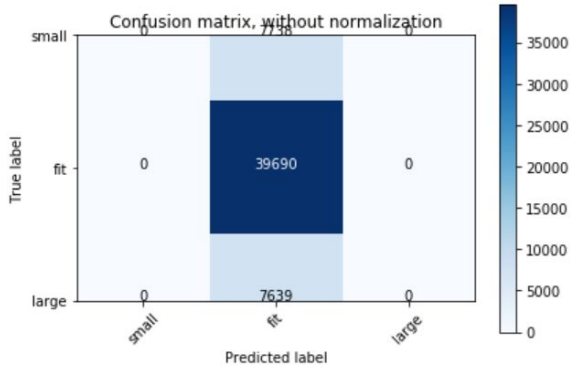
After changing the values of these parameters we obtained maximum accuracy of 72.25% with priors set to None and var_smoothing set to 1e-09. In this prior probabilities are calculated according to the data and hence obtained maximum accuracy.

We used different values for prior parameter which involved using different prior probabilities for the three classes:

1. model = GaussianNB(priors=None , var_smoothing=1e-09)
   Accuracy = 0.720758348920406
2. model = GaussianNB(priors=[0.6,0.2,0.2] , var_smoothing=1e-09)
   Accuracy = 0.14051973051010588
3. model = GaussianNB(priors=[0.2,0.2,0.6] , var_smoothing=1e-09)
   Accuracy = 0.13872192056948807
4. model = GaussianNB(priors=[0.2,0.6,0.2],var_smoothing=1e-09)
   Accuracy = 0.720758348920406

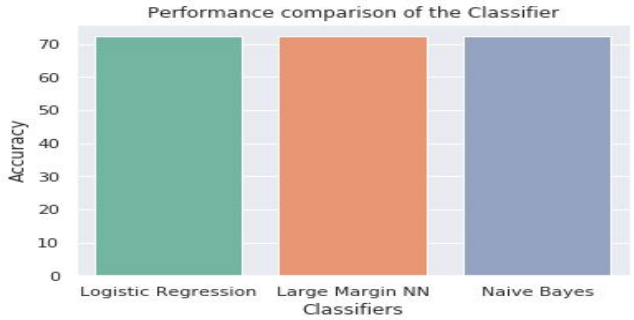
The results obtained from this experiment was not better as compared to the results obtained by setting priors to its default value.

<u>Confusion Matrix:</u>



Classification Report for Naive Bayes

	precision	recall	f1-score	support
1	0.00	0.00	0.00	7738
2	0.72	1.00	0.84	39690
3	0.00	0.00	0.00	7639
accuracy			0.72	55067
macro avg	0.24	0.33	0.28	55067
weighted avg	0.52	0.72	0.60	55067



After performing prediction fit using Naive Bayes classification, Logistic Regression and Large Margin NN Classifiers, we were able to achieve an accuracy of 72%.

Chapter 4: Discussion and Conclusion

Working on this project provided an outline as to how online shopping for apparel can be done more conveniently and can be more satisfying for the customers. Rather than going to the stores and buying , it would save a lot of time of customers. Fit prediction and size inference would help to get the perfect size of clothes for the customers.

Large Margin Nearest Neighbor a metric learning approach technique was suggested to be used as part of solving this problem fit determination[1]. Though we could not implement the complete algorithm as mentioned in the paper, we were able to implement it partially using the library provided by PyLMNN. Few other algorithms like Logistic Regression, Naive Bayes and KNN were used to analyse the machine learning problem. Tried implementing SVM for fit prediction but resulted in taking a lot of time for modelling and training.

Inference of the size of a product for a specific user involved some challenges. Firstly while working on merged data set involved a large set of values in the data set to be missing. So training model on such data set was not useful. So we used RentTheRunWay as the data set for inferring the size which involved a bit of less values to be missing. Second challenge faced was there were no users with multiple transactions which would help in inferring the specific size measurements. Only buying one type of category by a user would not act as a good data set for training and informing perfect size . So for this purpose we used Knn Algorithm that would group all the nearest neighbors(customers) together and therefore size could be inferred on similarity basis.

Implementing this project helped to get a better understanding as to how exactly large amount of data sets are analyzed. The methodology as to how exactly Machine learning and Data Mining Algorithms can be applied on such huge amount of data sets and further how exactly they can be evaluated.

Chapter 5: Project plan and Task distribution

We planned the project workflow distribution such that all of us could complete the given task on time. We divided tasks in such a way that we had minimal dependency on each other and yet merge our portion of task at the end with ease. The task was divided and implemented successfully in the following way:
Ashna Gupta: Predicted fit using Naive Bayes Algorithm. Studied about different Naive Baye Algorithms present and which would be suitable for our data set. Experimented with the algorithms using various parameters and correspondingly evaluated them. Data preprocessing steps were performed on the merged data set.
Priyanka Raju:

a) Performed Exploration and Visualization on both the Dataset (Fit-Prediction-Data-Visualization-and-Exploration.ipynb)
b) Applied all the preprocessing techniques needed for this project pre and post merging. Tried out various categorical encoding techniques (Fit-Prediction-Preprocessing-And-Merging-Dataset.ipynb)
c) Merging of Dataset (Fit-Prediction-Preprocessing-And-Merging-Dataset.ipynb)
d) Implemented KNN and Metric Learning Large Margin Nearest Neighbor algorithms (Fit_Prediction_ML_Algorithm_Implementation.ipynb)

Kesha Shah: Implemented SVM algorithm and Logistic Regression to predict fit with various different parameters and tried this on different tools because was facing problem at some places. Did preprocessing for my algorithm.

We used Renttherunway dataset and initially thought of using tree based algorithm to infer the size of the customers. Eventually we used KNN classifier to recommend size to the users.

References

[1] Misra, Rishabh, Mengting Wan, and Julian McAuley. "Decomposing fit semantics for product size recommendation in metric spaces." Proceedings of the 12th ACM Conference on RecommenderSystems . ACM, 2018. [2] http://cseweb.ucsd.edu/~jmcauley/datasets.html#clothing_fit
[3]https://github.com/rishabhmisra/Product-Catalog-Size-Recommendation-Framework
[4]https://www.kaggle.com/rmisra/clothing-fit-dataset-for-size-recommendation
[5]https://towardsdatascience.com/demystifying-confusion-matrix-confusion-9e82201592fd
[6]https://intellipaat.com/blog/what-is-logistic-regression/
[7]https://towardsdatascience.com/logistic-regression-using-python-sklearn-numpy-mnist-handwriting-re
[8]cognition-matplotlib-a6b31e2b166a
[9]https://stackoverflow.com/questions/38640109/logistic-regression-python-solvers-defintions
[10]https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02
[11]https://pypi.org/project/PyLMNN/ [12]https://scikit-learn.org/
[13]https://towardsdatascience.com/importance-of-distance-metrics-in-machine-learning-modelling-e51395ffe60d
[14]http://contrib.scikit-learn.org/metric-learn/introduction.html
[15]https://seaborn.pydata.org/index.html
[17]https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn
[18]https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
[19]https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html