CSCI 6364 Machine Learning – Assignment 2

Abde Manaaf Ghadiali – G29583342

Disclaimer – Much of the Information for Question 1 and 2 has been excluded from the report due to the report's length. Please refer to the Jupyter Notebook which is present along with this report in the zip file.

Question 1: Binary Classification with Custom Naive Bayes

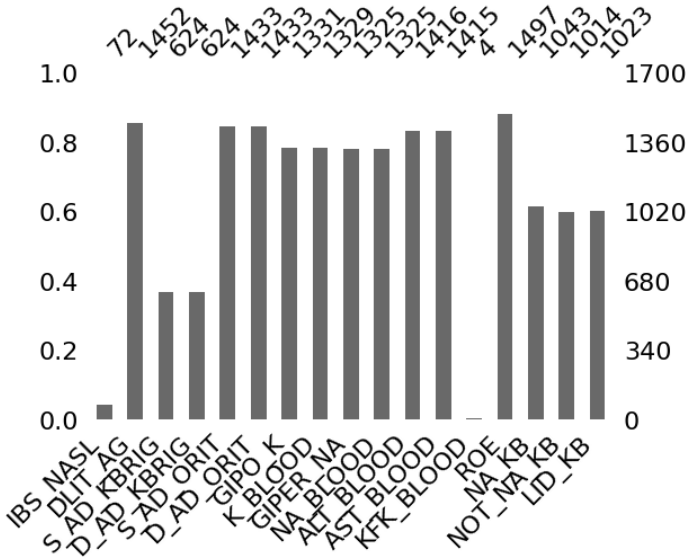
Objective

This assignment aims to construct a binary classification model employing the **Naive Bayes algorithm**. The goal is to utilize the **Myocardial Infarction Complications** dataset to forecast **Chronic Heart Failure** occurrences in patients, leveraging diverse features.

Data Source: https://archive.ics.uci.edu/dataset/579/myocardial+infarction+complications

Data Loading and Preprocessing

1. The data is fetched using the provided code snippet from the website after installing the **ucimlrepo** library via pip. Utilizing this library, we extract the dataset (ID = 579), obtaining our features, target, and metadata for column descriptions.
2. We exclude the 'ZSN_A' feature from the independent feature list, as it essentially duplicates the target 'ZSN' feature with different binning. This leaves us with **110 features and 1700 records**.
3. Now looking at the data, we realize it has a lot of **null values**. To handle these null values, we approach it in 2 steps. Firstly, we see all the features with null values and then decide on a **threshold** (here its 0.1 i.e., any feature with more than 10% null values) and drop those features from our input data whose null values percentage is greater than the threshold. Below are the features that are dropped:



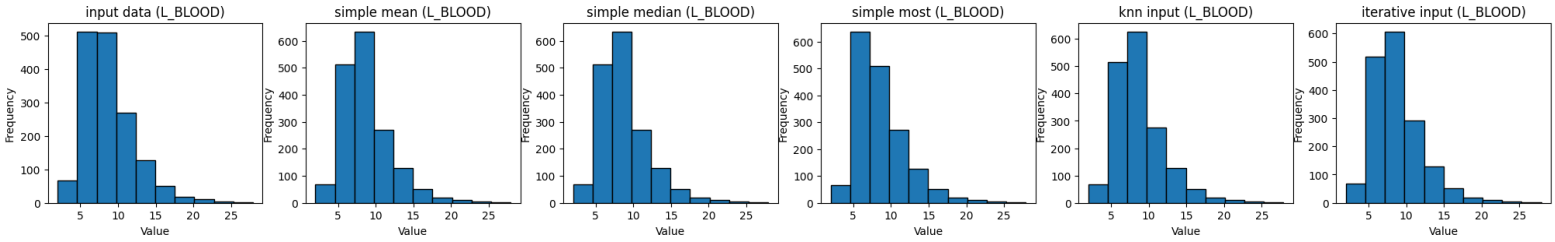
4. After dropping the above features, we are left with
5. For the remaining features with null values, we apply 5 methods of imputation and save each dataset into a dictionary:

   a. **Simple Imputer with Mean Strategy**
   b. **Simple Imputer with Median Strategy**
   c. **Simple Imputer with Most Frequent Strategy**
   d. **KNN Imputer with 25 neighbors**
   e. **Iterative Imputer with Bayesian Ridge Estimator**
6. Following this, we standardize each of our datasets using three scalers and one transformer:
   a. **Standard Scaler with Power Transformer**
   b. **Min Max Scaler with Power Transformer**
   c. **Robust Scaler with Power Transformer**
7. Lastly, we split the datasets into train and test (with test ratio = 0.2, since our number of records is very small) and stratify on the target feature to get an approx. same distribution for train and test.

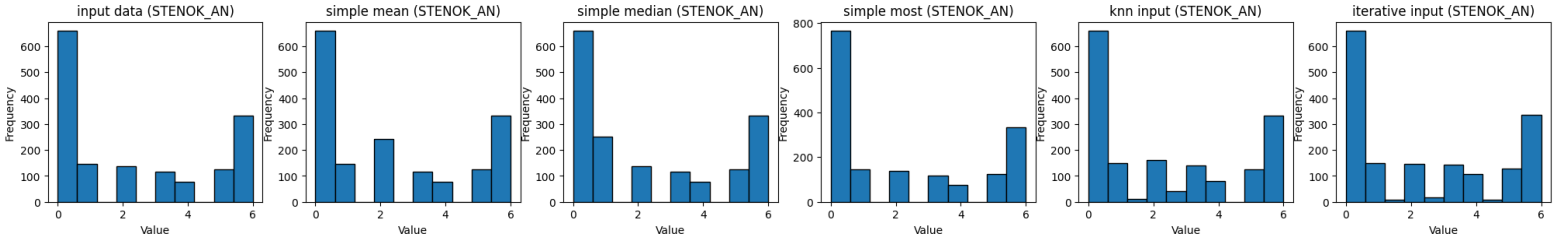
Data Visualization

1. Looking at the data, we realize that we have three types of features: **Continuous (4), Categorical (15) and Binary (72)** with a dominance of Binary features.
2. Following this, we plot **Histograms** for all the feature set side by side, to see how our imputation changed the raw dataset.

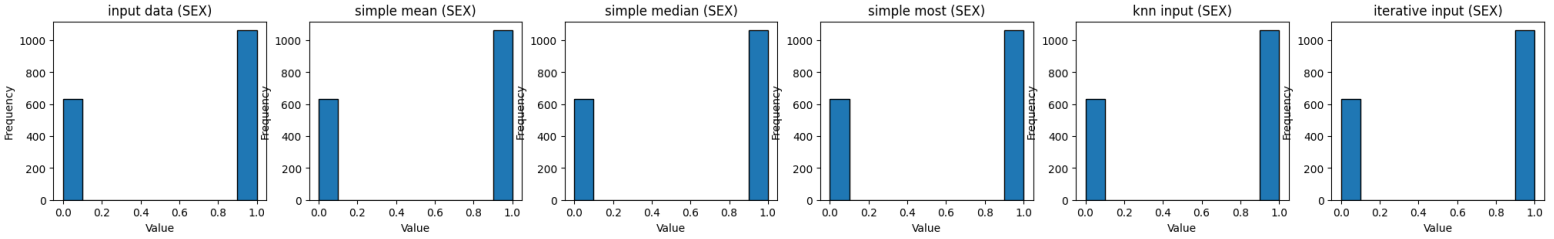
Continuous Feature (Histogram)



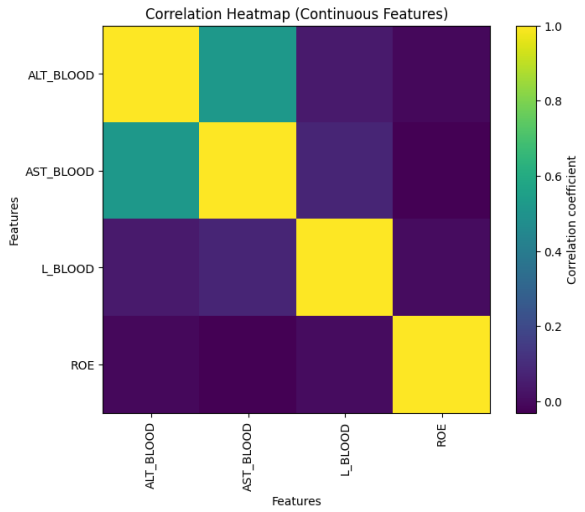
Categorical Feature (Histogram)

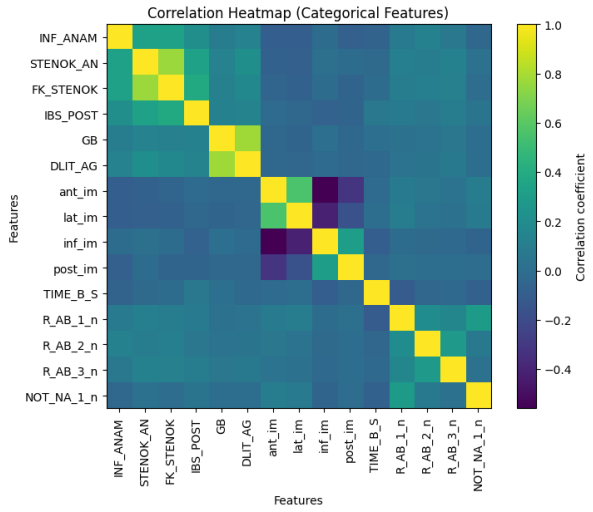


Binary Feature (Histogram)



3. We also plot the **Correlation Heat Map** to see if there are any correlated features in the data. Here we see most of the features have almost zero correlation with each other.





4. Lastly, we see the distribution of our target feature to check if the data is balanced or imbalanced. Upon inspection we realize the distribution of class is: **0 – 1306 Records (77%)** and **1 – 394 Records (23%)**

Model Development

1. We start the model development process by first selecting the important features. We employ 3 different **feature selection** methods and then apply **PCA (dimensionality reduction)** to reduce the feature space even more. We apply feature selection because keeping all the features some of which might not be proving useful in the final model prediction and just increasing the error.
2. Our feature selection process first removes the **highly correlated features** (at a **threshold of 0.8**) (using **Pearson's Correlation Coefficient**).
3. Next, we use **Point Bi-Serial Correlation** to calculate Coefficients between features and target to see if any of the features are correlated to the target. We also get the **P-Value Statistic** based on which we filter out the features and keep only those features whose P-Value is less than **0.05**.

	Data Subset	# of Features Before	# of Features After
0	standard_simple_mean_input_data_0.1	93	19
1	standard_simple_median_input_data_0.1	93	24
2	standard_simple_most_frequent_input_data_0.1	93	24
3	standard_knn_input_data_0.1	93	21
4	standard_iterative_input_data_0.1	93	21
5	min_max_simple_mean_input_data_0.1	93	19
6	min_max_simple_median_input_data_0.1	93	24
7	min_max_simple_most_frequent_input_data_0.1	93	24
8	min_max_knn_input_data_0.1	93	21
9	min_max_iterative_input_data_0.1	93	21
10	robust_simple_mean_input_data_0.1	93	19
11	robust_simple_median_input_data_0.1	93	24
12	robust_simple_most_frequent_input_data_0.1	93	24
13	robust_knn_input_data_0.1	93	21
14	robust_iterative_input_data_0.1	93	21

4. After dropping Correlated Features, we apply two methods of feature selection: **Recursive Feature Elimination (RFE)** and **Sequential Feature Selection (Forward) (SFS)**. The estimator for both is **Random Forest Classifier** and we selected the best 15 Features.
5. On those 15 features we apply **PCA (Dimensionality Reduction)** to further reduce our feature space to only the most important uncorrelated principal components. This gives us our final dataset with **10 features** in each data subset for model building.
6. The last step before model building is **balancing the data**. Since we know our target feature is **imbalanced**, we need to employ some sampling techniques to balance out the data by creating synthetic data or removing records from our data. We tried two balancing techniques: **SVMSMOTE (for Over Sampling)** and **Repeated Edited Nearest Neighbors (RENN) (for Under Sampling)**.
7. Finally for the model building process we have **15 data subsets**. We apply **Gaussian Naïve Bayes Classifier** model from **Scikit-Learn** Library on all our data subsets.

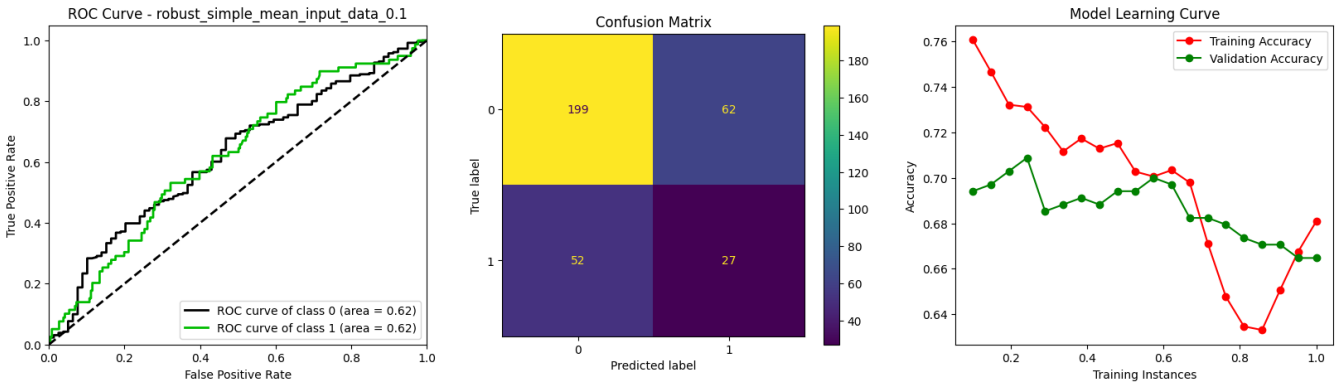
Performance Analysis

Below is the Final **Model Evaluation Metric** Table along with the **ROC Curve**, **Training Vs Test Accuracy Curve**, and the **Confusion Matrix**.

Table for RFE Selected Features on Gaussian Naïve Bayes Classifier:

Data Subset	*Train Accuracy*	*Test Accuracy*	*ROC*	*Precision*	*Recall*	*F-1 Score*
Standard *Simple Mean*	0.681	0.665	0.619	0.303	0.342	0.321
Standard *Simple Median*	0.686	0.674	0.603	0.278	0.253	0.265
Standard *Simple Most Frequent*	0.686	0.674	0.603	0.278	0.253	0.265
Standard *KNN*	0.674	0.700	0.590	0.283	0.190	0.227
Standard *Iterative*	0.674	0.703	0.604	0.323	0.253	0.284
Min Max Simple Mean	0.681	0.665	0.619	0.303	0.342	0.321
Min Max Simple Median	0.686	0.674	0.603	0.278	0.253	0.265
Min Max Simple Most Frequent	0.686	0.674	0.603	0.278	0.253	0.265
Min Max KNN	0.674	0.700	0.590	0.283	0.190	0.227
Min Max Iterative	0.674	0.703	0.604	0.323	0.253	0.284
Robust Simple Mean	0.681	0.665	0.619	0.303	0.342	0.321
Robust Simple Median	0.686	0.674	0.603	0.278	0.253	0.265
Robust Simple Most Frequent	0.686	0.674	0.603	0.278	0.253	0.265
Robust KNN	0.674	0.700	0.590	0.283	0.190	0.227
Robust Iterative	0.674	0.703	0.604	0.323	0.253	0.284

ROC Curve, Confusion Matrix, and Model Learning Curve



Reflection and Insights

1. **Consistency in Performance**: Across various preprocessing methods and scaling techniques (Standard, Min Max, Robust), there is a remarkable consistency in both train and test accuracies, ROC, precision, recall, and F-1 scores. This consistency suggests that the choice of preprocessing method may have a limited impact on the model's performance.
2. **Impact of Imputation Strategy**: Notably, the choice of imputation strategy (Mean, Median, Most Frequent) does not significantly affect the model's performance, as evidenced by the similarity in performance metrics across different imputation methods.
3. **Generalization Capability**: The model's ability to achieve comparable performance on both training and test datasets suggests that it can generalize well to unseen data. This is crucial for real-world applications where the model needs to make accurate predictions on new, previously unseen instances.
4. **Handling Imbalanced Data**: Despite the imbalance in the target feature, the model demonstrates the capability to predict chronic heart failure occurrences effectively. While there may be some room for improvement in metrics related to the minority class, the model's ability to maintain consistent performance across different subsets indicates its efficacy in handling imbalanced data.
5. **Room for Improvement**: While the model demonstrates promising predictive power, there is always room for improvement. Further exploration of feature engineering techniques, model selection, and hyperparameter tuning could potentially enhance the model's performance and predictive power. Additionally, incorporating domain knowledge or exploring alternative machine learning algorithms may lead to better predictive outcomes.

Question 2: Emotion Analysis using SVM and K-Means

Objective

Develop an understanding of emotion recognition in text using **Support Vector Machines (SVM)** for classification and **K-Means clustering** for pattern discovery The goal is to utilize the **Emotion Lines** dataset to predict the different emotions based on the sentence utterance by a speaker (in text format).

Data Source: https://blackboard.gwu.edu/bbcswebdav/pid-13804467-dt-content-rid-126822844_2/xid-126822844_2

Data Preparation

1. The Emotion Lines Data in its Raw format was JSON list of lists of dictionaries. We have 3 such JSON files for training, validation (dev) and test. Each dictionary had four key-value pairs with the keys being, speaker, utterance, emotion, annotation. We convert each list of dictionaries to a Pandas Dataframe and concatenate these data frames to create one Dataframe for each train, dev, test.

	speaker	utterance	emotion	annotation
0	Chandler	also I was the point person on my company□s tr...	neutral	4100000
1	The Interviewer	You must□ve had your hands full.	neutral	5000000
2	Chandler	That I did. That I did.	neutral	5000000
3	The Interviewer	So let□s talk a little bit about your duties.	neutral	5000000
4	Chandler	My duties? All right.	surprise	2000030
...	...	...	...	...
10556	Chandler	You or me?	neutral	3000011
10557	Ross	I got it. Uh, Joey, women don't have Adam's ap...	non-neutral	2100011
10558	Joey	You guys are messing with me, right?	surprise	0000050
10559	All	Yeah.	neutral	4000010
10560	Joey	That was a good one. For a second there, I was...	non-neutral	1200020

2. We see that our Raw data consists of 10516 Training Records, 1178 Validation Records and 2764 Testing Records. Also, there are a few Hexadecimal Values (mainly ‘) which have not been decoded properly. We replace those hexadecimal values with the appropriate ASCII value.

	speaker	utterance	emotion	annotation	processed_text
0	Chandler	also I was the point person on my company□s tr...	neutral	4100000	also i was the point person on my company's tr...
1	The Interviewer	You must□ve had your hands full.	neutral	5000000	you must have had your hands full.
2	Chandler	That I did. That I did.	neutral	5000000	that i did. that i did.
3	The Interviewer	So let□s talk a little bit about your duties.	neutral	5000000	so let us talk a little bit about your duties.
4	Chandler	My duties? All right.	surprise	2000030	my duties? all right.
...	...	...	...	...	...
10556	Chandler	You or me?	neutral	3000011	you or me?
10557	Ross	I got it. Uh, Joey, women don't have Adam's ap...	non-neutral	2100011	i got it. uh, joey, women do not have adam's a...
10558	Joey	You guys are messing with me, right?	surprise	0000050	you guys are messing with me, right?
10559	All	Yeah.	neutral	4000010	yeah.
10560	Joey	That was a good one. For a second there, I was...	non-neutral	1200020	that was a good one. for a second there, i was...

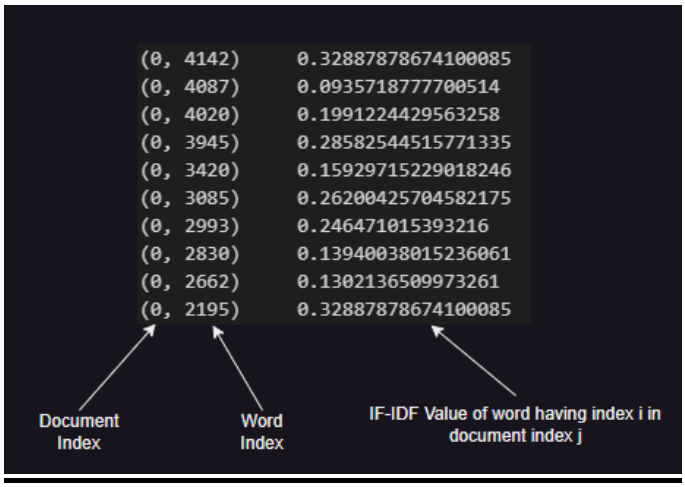
3. Next, we convert all the strings to Lower Case and fix any Contractions present in the text. Following this we tokenize each sentence into Word Token which breaks the sentences into words.

	speaker	utterance	emotion	annotation	processed_text
0	Chandler	also I was the point person on my company□s tr...	neutral	4100000	[also, i, was, the, point, person, on, my, com...
1	The Interviewer	You must□ve had your hands full.	neutral	5000000	[you, must, have, had, your, hands, full]
2	Chandler	That I did. That I did.	neutral	5000000	[that, i, did, that, i, did]
3	The Interviewer	So let□s talk a little bit about your duties.	neutral	5000000	[so, let, us, talk, a, little, bit, about, you...
4	Chandler	My duties? All right.	surprise	2000030	[my, duties, all, right]
...	...	...	...	...	...
10556	Chandler	You or me?	neutral	3000011	[you, or, me]
10557	Ross	I got it. Uh, Joey, women don't have Adam's ap...	non-neutral	2100011	[i, got, it, uh, joey, women, do, not, have, a...
10558	Joey	You guys are messing with me, right?	surprise	0000050	[you, guys, are, messing, with, me, right]
10559	All	Yeah.	neutral	4000010	[yeah]
10560	Joey	That was a good one. For a second there, I was...	non-neutral	1200020	[that, was, a, good, one, for, a, second, ther...

4. From here, we divide the dataset into two streams, one where we remove the stop words and the other where we do not remove the stop words. On both the data features, we apply Lemmatization technique (with Parts of Speech Tagging) to transform the words into its root form.

	speaker	utterance	emotion	annotation	processed_text	stop_words_removed_text
0	Chandler	also I was the point person on my company□s tr...	neutral	4100000	[also, i, be, the, point, person, on, my, comp...	[also, point, person, company, transition, kl,...
1	The Interviewer	You must□ve had your hands full.	neutral	5000000	[you, must, have, have, your, hand, full]	[must, hand, full]
2	Chandler	That I did. That I did.	neutral	5000000	[that, i, do, that, i, do]	NaN
3	The Interviewer	So let□s talk a little bit about your duties.	neutral	5000000	[so, let, us, talk, a, little, bit, about, you...	[let, us, talk, little, bit, duty]
4	Chandler	My duties? All right.	surprise	2000030	[my, duty, all, right]	[duty, right]
...	...	...	...	...	...	...
10556	Chandler	You or me?	neutral	3000011	[you, or, me]	NaN
10557	Ross	I got it. Uh, Joey, women don't have Adam's ap...	non-neutral	2100011	[i, get, it, uh, joey, woman, do, not, have, a...	[get, uh, joey, woman, adam, apple]
10558	Joey	You guys are messing with me, right?	surprise	0000050	[you, guy, be, mess, with, me, right]	[guy, mess, right]
10559	All	Yeah.	neutral	4000010	[yeah]	[yeah]
10560	Joey	That was a good one. For a second there, I was...	non-neutral	1200020	[that, be, a, good, one, for, a, second, there...	[good, one, second, like, whoa]

5. Lastly, we employ TF-IDF Vectorization to get the statistic of a word in the text relative to all the text. We apply TF-IDF Vectorization fit method to the training data and transform the validation and test data. Along with TF-IDF on the Text data, we apply Label Encoder on the target data to get numeric encoding of the emotions.



SVM for Emotion Classification

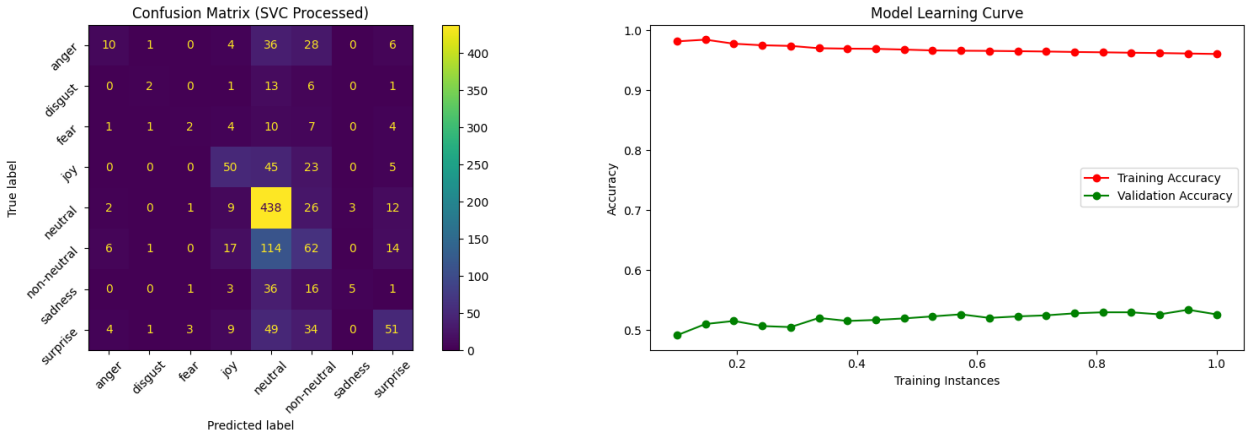
1. After we have completed the TF-IDF Vectorization, we start building the standard default SVM Model. Initially we built two models. One for token with stop words removed and for text without stop words removed. This will be our baseline model.

Data Subset	Cross Validation Train Accuracy	Cross Validation Test Accuracy
Data With Stop Words	0.9627	0.5587
Data Without Stop Words	0.9404	0.5398

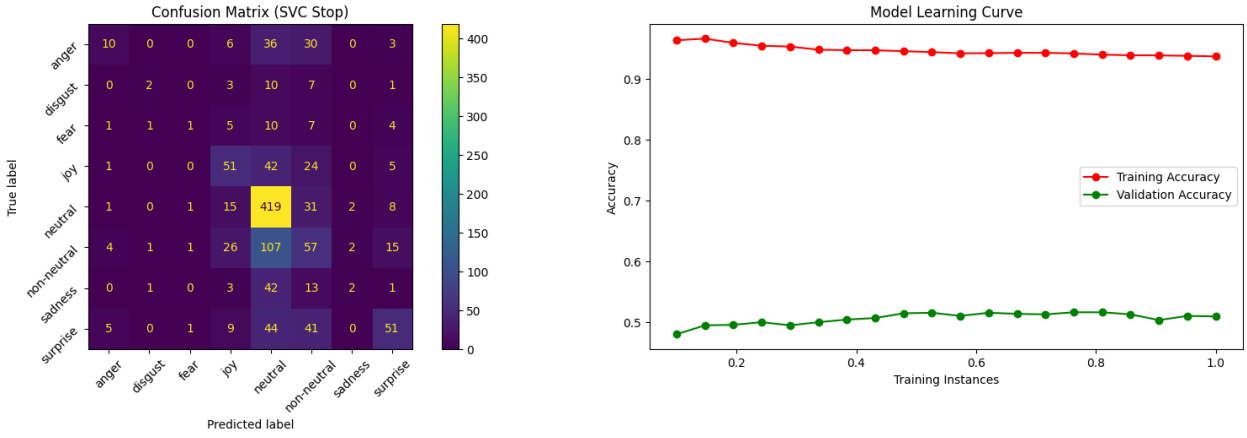
2. After getting the Baseline, we employ Grid Search to find the best parameters for SVC and build out two models based on the best parameters to get out final model. Since our target is imbalanced, we add class weights as a parameter to balance the weights of each class.
3. After Executing Grid Search, we realize that the Base Line is the best SVM Model as the Grid Search also provided with the same parameters for SVM.

Data Subset	Train Accuracy	Validation Accuracy	Test Accuracy
Data With Stop Words	0.960	0.526	0.567
Data Without Stop Words	0.937	0.510	0.560

Confusion Matrix and Accuracy Plot for Data with Stop Words

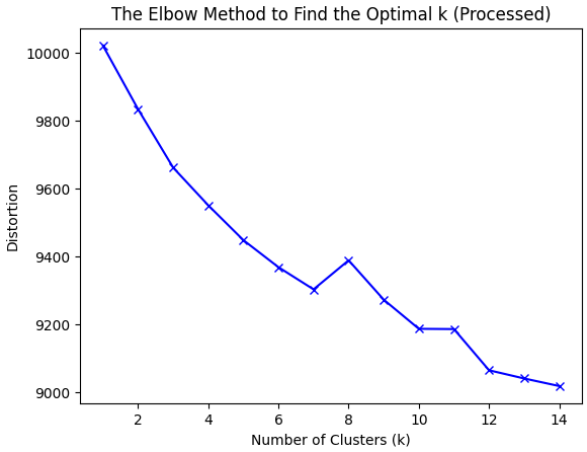


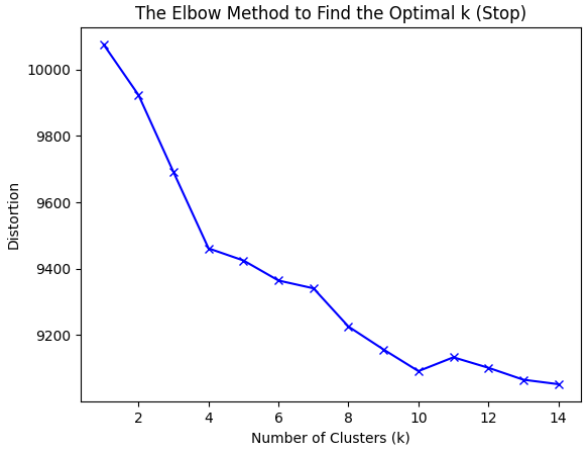
Confusion Matrix and Accuracy Plot for Data without Stop Words



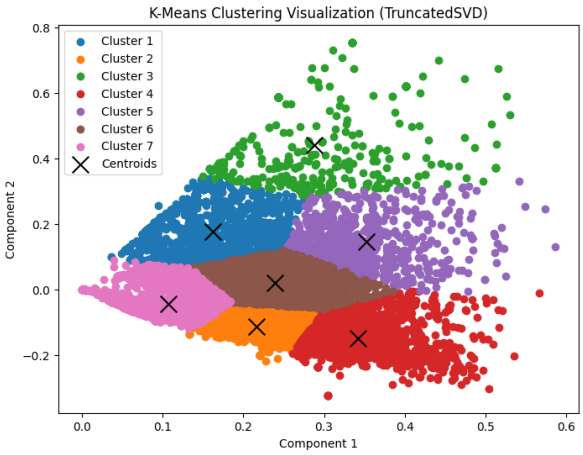
K-Means Clustering

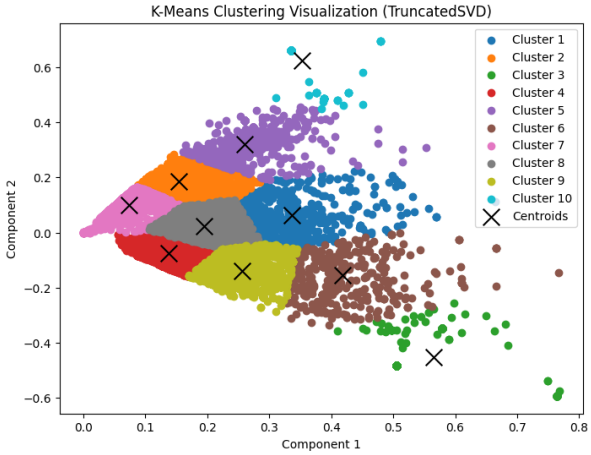
1. We assume twice the number target classes as the maximum number of clusters and iterate over the range to create the K-Means Model. We then save the inertia_ attribute of the model in a list. This becomes our distortions as inertia_ is nothing but sum of squared distances of samples to their closest cluster center.
2. We apply this approach on both our data set and plot the graph of these distortions to get the optimal number of clusters using the elbow method.





3. Once we have the optimal number of clusters, we apply Truncated SVD on the input data sets to reduce the feature space of the data. We get the 2 principal components and run a K-Means Model on this data with the optimal number of clusters.
4. We then plot the 2 principal components and the cluster mapping with the centroid of each cluster.





Model Insights

1. From the SVM Model Stats and the K-Means distortions and clusters, we can infer that Unsupervised Learning is visually performing better than SVM model. Our K-Means can detect the 7 different clusters for the different emotions.
2. Secondly Our SVM Model is Overfitting the Data as the Training Accuracy is wat more than the Validation or Testing Accuracy.
3. Lastly, we can say that SVM model might not be the best fit model for this task as it cannot detect the non-linearity of the data which is in contrast to the K-Means Clustering method which is able to capture the different clusters.