

EXPERIMENT NO- 10

AIM: To develop social media text analytics models for improving existing product/ service by analyzing customer 's reviews/comments.

RESOURCES REQUIRED: Windows/MAC/Linux O.S, Compatible version of Python.

THEORY:

Text mining is the process of examining large collections of text and converting the unstructured text data into structured data for further analysis like visualization and model building. In this article, We will utilize the power of text mining to do an in-depth analysis of customer reviews on an e-commerce clothing site.

Customer reviews are a great source of “Voice of customer” and could offer tremendous insights into what customers like and dislike about a product or service. For the e-commerce business, customer reviews are very critical, since existing reviews heavily influence buying decision of new customers in the absence of the actual look and feel of the product to be purchased.

How to use text analytics to improve CX

Step 1: create dashboards to see the data that matters the most

Step 2: analyze data to avoid losing customers

Step 3: get insights on what your customer thinks about specific areas

Step 4: discover hidden trends the moment they arise

Step 5: understand what your customers think about support staff

A strong customer analytics process helps you understand who your customers are, how they behave, and how satisfied they are with your company or product experience. Customer analytics are as important as financial reports in gauging the health of your business. Custom analytics dashboards and reports can track team performance, identify bottlenecks before a process breaks, and measure customer satisfaction (CSAT) on an ongoing basis. This data can help you build efficiencies, understand customer needs and behaviors, and resolve problems faster — or eliminate them altogether.

What is review analysis?

Review analysis is the process of transforming unstructured review data to structured data that can be used to guide decision-making.

A few of the primary uses include:

- Product feature ideas: scanning product reviews for sentiment on desired features
- Roadmap prioritization: determining what the dev team should focus on first
- Bug tracking: scanning new reviews in real-time
- Customer care ratings: determining what customer service departments are delivering good service.

Product reviews are one of the most comprehensive and useful sources of insider info available, anywhere. These often have gold nuggets that can guide development, troubleshoot new initiatives, and improve customer experience. They provide a helpful benchmark to compare your offerings to the competition, showing both your strong points and areas in which, you need to close the gap. Think free customer research, there before you've even thought to ask.

CONCLUSION: Hence, we have successfully developed social media text analytics models for improving existing product/ service

Problem Statement

This is a dataset containing 60,000 Stack Overflow questions from 2016-2020. Questions are classified into three categories:

- HQ: High-quality posts without a single edit.
- LQ_EDIT: Low-quality posts with a negative score, and multiple community edits. However, they still remain open after those changes.
- LQ_CLOSE: Low-quality posts that were closed by the community without a single edit.

Aim - To build a classifier model on training data(train.csv) and validate its performance on validation data(valid.csv)

Importing Libraries

```
In [7]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
import random

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
import sklearn.metrics

import matplotlib.pyplot as plt
import seaborn as sns
import plotly
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.figure_factory as ff
```

Data Preparation

```
In [8]: df_train = pd.read_csv("train.csv")
df_train.head()
```

```
Out[8]:
```

	Id	Title	Body	Tags	CreationDate	Y
0	34552656	Java: Repeat Task Every Random Seconds	<p>I'm already familiar with repeating tasks e...	<java> <repeat>	2016-01-01 00:21:59	LQ_CLOSE
1	34553034	Why are Java Optionals immutable?	<p>I'd like to understand why Java 8 Optionals...	<java> <optional>	2016-01-01 02:03:20	HQ
2	34553174	Text Overlay Image with Darkened Opacity React...	<p>I am attempting to overlay a title over an ...	<javascript> <image> <overlay> <react-native> <opa...	2016-01-01 02:48:24	HQ
3	34553318	Why ternary operator in swift is so picky?	<p>The question is very simple, but I just cou...	<swift> <operators> <whitespace> <ternary-operato...	2016-01-01 03:30:17	HQ
4	34553755	hide/show fab with scale animation	<p>I'm using custom floatingactionmenu. I need...	<android> <material-design> <floating-action-but...	2016-01-01 05:21:48	HQ

```
In [9]: df_train.count()
```

```
Out[9]: Id          45000
Title          45000
Body           45000
Tags           45000
CreationDate   45000
Y              45000
dtype: int64
```

Checking for nulls in training dataset.

```
In [10]: df_train.isna().sum().sum()
```

```
Out[10]: 0
```

```
In [11]: df_val = pd.read_csv("valid.csv")
df_val.head()
```

Out[11]:

	Id	Title	Body	Tags	CreationDate	Y
0	34552974	How to get all the child records from differen...	I am having 4 different tables like \r\nselect...	<sql><sql-server>	2016-01-01 01:44:52	LQ_EDIT
1	34554721	Retrieve all except some data of the another t...	I have two table m_master and tbl_appointment\...	<php> <mysql><sql> <codeigniter> <mysqli>	2016-01-01 08:43:50	LQ_EDIT
2	34555135	Pandas: read_html	<p>I'm trying to extract US states from wiki U...	<python> <pandas>	2016-01-01 09:55:22	HQ
3	34555448	Reader Always gimme NULL	I'm so new to C#, I wanna make an application ...	<sql-server> <c#-4.0>	2016-01-01 10:43:45	LQ_EDIT
4	34555752	php rearrange array elements based on condition	basically i have this array:\r\n\r\narray(...	<php>	2016-01-01 11:34:09	LQ_EDIT

In [12]: `df_val.count()`

```
Out[12]: Id          15000
Title        15000
Body         15000
Tags         15000
CreationDate 15000
Y            15000
dtype: int64
```

Checking for nulls in validation dataset.

In [13]: `df_val.isna().sum().sum()`

Out[13]: 0

Data Preprocessing

Dropping ID, Tags and CreationDate.

At the moment we shall only focus on text and its category("Y") we can reserve Tags and CreationDate for another way to re-visit this problem.

```
In [14]: df_train = df_train.drop(['Id', 'Tags', 'CreationDate'], axis=1)
df_train['Y'] = df_train['Y'].map({'LQ_CLOSE':0, 'LQ_EDIT': 1, 'HQ':2})
df_train.head()
```

Out[14]:

	Title	Body	Y
0	Java: Repeat Task Every Random Seconds	<p>I'm already familiar with repeating tasks e...	0
1	Why are Java Optionals immutable?	<p>I'd like to understand why Java 8 Optionals...	2
2	Text Overlay Image with Darkened Opacity React...	<p>I am attempting to overlay a title over an ...	2
3	Why ternary operator in swift is so picky?	<p>The question is very simple, but I just cou...	2
4	hide/show fab with scale animation	<p>I'm using custom floatingactionmenu. I need...	2

```
In [15]: df_val = df_val.drop(['Id', 'Tags', 'CreationDate'], axis=1)
df_val['Y'] = df_val['Y'].map({'LQ_CLOSE':0, 'LQ_EDIT': 1, 'HQ':2})
df_val.head()
```

Out[15]:

	Title	Body	Y
0	How to get all the child records from differen...	I am having 4 different tables like \r\nselect...	1
1	Retrieve all except some data of the another t...	I have two table m_master and tbl_appointment...	1
2	Pandas: read_html	<p>I'm trying to extract US states from wiki U...	2
3	Reader Always gimme NULL	I'm so new to C#, I wanna make an application ...	1
4	php rearrange array elements based on condition	basically i have this array:\r\n\r\n array(...	1

We join the title and the body of the provided text data to leverage both during classification process.

```
In [16]: df_train['concat_text'] = df_train['Title'] + ' ' + df_train['Body']
df_train = df_train.drop(['Title', 'Body'], axis=1)
df_train.head()
```

Out[16]:

	Y	concat_text
0	0	Java: Repeat Task Every Random Seconds <p>I'm ...
1	2	Why are Java Optionals immutable? <p>I'd like ...
2	2	Text Overlay Image with Darkened Opacity React...
3	2	Why ternary operator in swift is so picky? <p>...
4	2	hide/show fab with scale animation <p>I'm usin...

```
In [17]: df_val['concat_text'] = df_val['Title'] + ' ' + df_val['Body']
df_val = df_val.drop(['Title', 'Body'], axis=1)
df_val.head()
```

```
Out[17]:
```

	Y	concat_text
0	1	How to get all the child records from differen...
1	1	Retrieve all except some data of the another t...
2	2	Pandas: read_html <p>I'm trying to extract US ...
3	1	Reader Always gimme NULL I'm so new to C#, I w...
4	1	php rearrange array elements based on conditio...

Defining a custom text cleaning function

This function does the following -

- Standardizes text by lowercasing all of it.
- Uses regex to get rid of tags, special characters, etc.

```
In [18]: def cleanse(concat_text):
          concat_text = concat_text.lower()
          concat_text = re.sub(r'^([a-zA-Z]\s)', '', concat_text)
          return concat_text
```

```
In [19]: df_train['cleaned_text'] = df_train['concat_text'].apply(cleanse)
```

```
In [20]: df_val['cleaned_text'] = df_val['concat_text'].apply(cleanse)
```

Shuffles dataframe

```
In [21]: df_train = df_train.sample(frac=1).reset_index(drop=True)
          df_val = df_val.sample(frac=1).reset_index(drop=True)
```

Assigning X and Y values

For both training and validation datasets -

- X Variable - Cleaned text
- Y Variable - Y

```
In [22]: X_train = df_train['cleaned_text']
          Y_train = df_train['Y'].values
```

```
In [23]: X_val = df_val['cleaned_text']
          Y_val = df_val['Y'].values
```

Vectorization of data

Vectorizer : TF-IDF Vectorizer

TF-IDF stands for “Term Frequency — Inverse Document Frequency”. This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus.

We load the TF-IDF vectorizer, first we fit it on training dataset and then transform it on validation dataset.

```
In [24]: vectorizer = TfidfVectorizer()  
X_train = vectorizer.fit_transform(X_train)  
X_val = vectorizer.transform(X_val)
```

Checking sizes of training and validation datasets

```
In [25]: print("Shape of training dataset size :",df_train.shape)  
print("Shape of validation dataset size :",df_val.shape)
```

```
Shape of training dataset size : (45000, 3)  
Shape of validation dataset size : (15000, 3)
```

ML Modelling

We approach the problem by building ML Models from simple to complex models on training set and then assess them on the basis of their scores on validation dataset.

Multinomial Naive Bayes

```
In [26]: nb_classifier = MultinomialNB()  
nb_classifier.fit(X_train, Y_train)
```

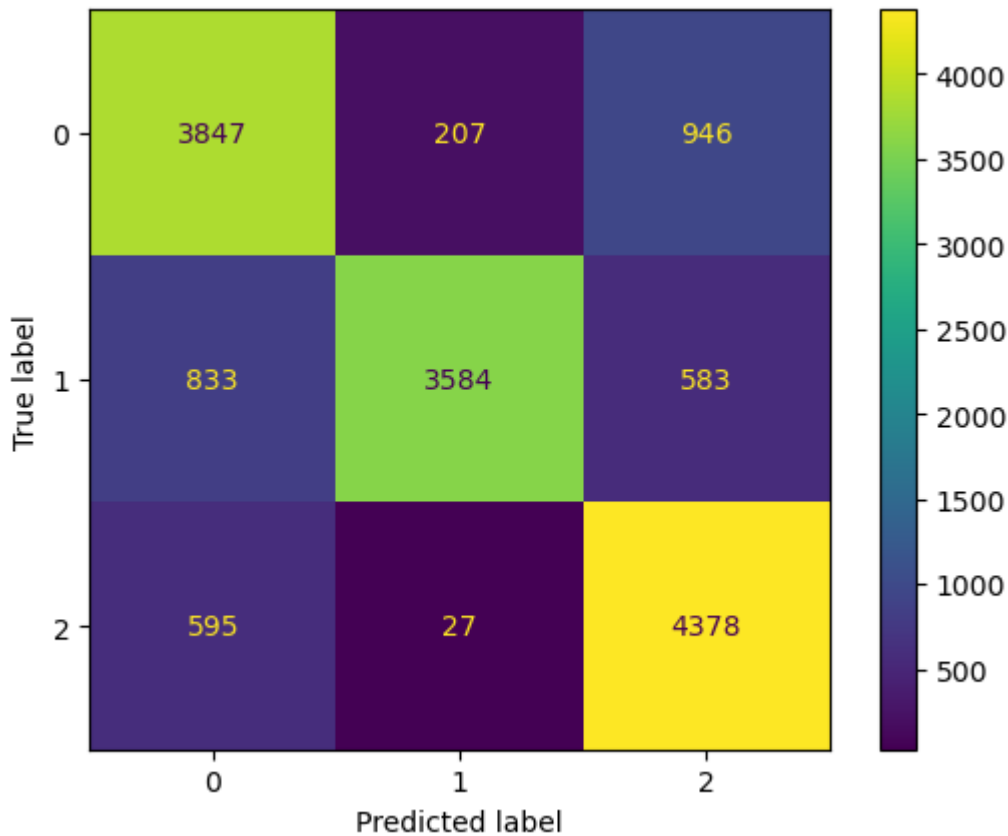
```
Out[26]: MultinomialNB()
```

```
In [27]: nb_val_accuracy = nb_classifier.score(X_val, Y_val)  
print(f"Validation Accuracy of Naive Bayes Classifier is: {(nb_val_accu
```

```
Validation Accuracy of Naive Bayes Classifier is: 78.73%
```

```
In [28]: sklearn.metrics.ConfusionMatrixDisplay.from_estimator(nb_classifier, X_val,
```

```
Out[28]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fcf  
8af5a460>
```



Logistic Regression

```
In [29]: lr_classifier = LogisticRegression(C=1.)
lr_classifier.fit(X_train, Y_train)
```

```
/home/slowgamer/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_logistic.py:814: ConvergenceWarning:
```

```
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
https://scikit-learn.org/stable/modules/preprocessing.html
```

```
Please also refer to the documentation for alternative solver options:
```

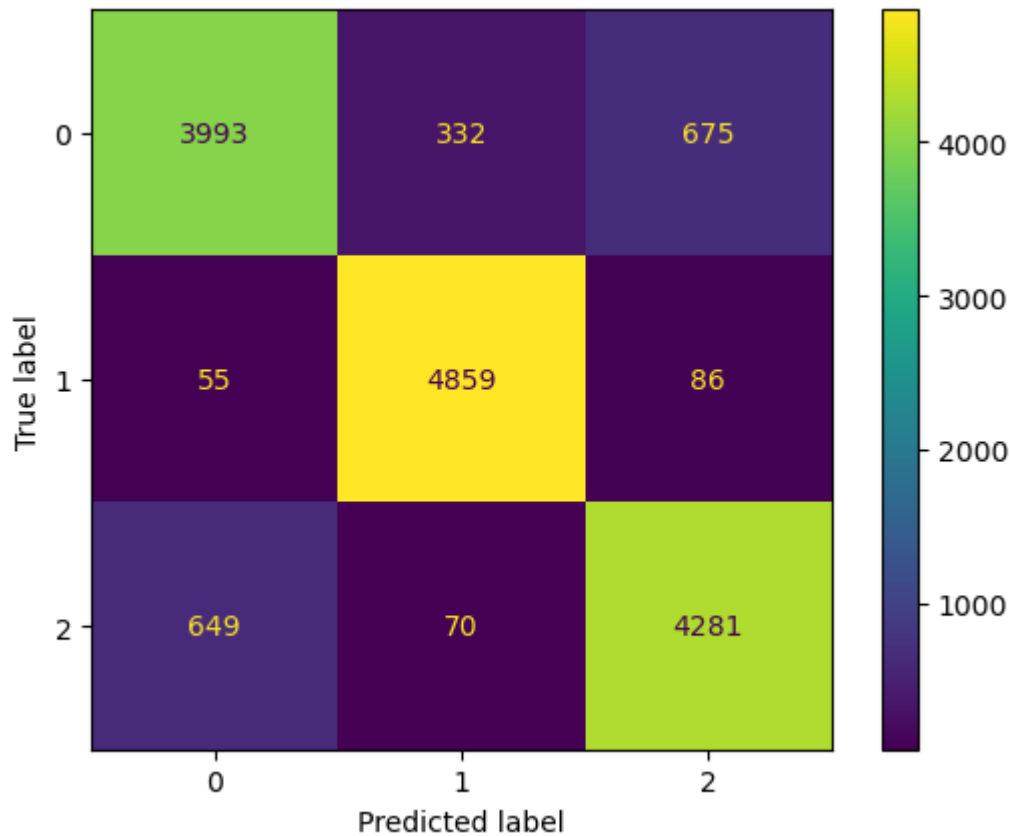
```
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
```

```
Out[29]: LogisticRegression()
```

```
In [30]: lr_val_accuracy = lr_classifier.score(X_val, Y_val)
print(f"Validation Accuracy of Logistic Regression Classifier is: {(lr_val_accuracy)}")
Validation Accuracy of Logistic Regression Classifier is: 87.55%
```

```
In [31]: sklearn.metrics.ConfusionMatrixDisplay.from_estimator(lr_classifier, X_val, Y_val)
```

```
Out[31]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fcf85911790>
```

KNN Classifier

```
In [32]: knn_classifier = KNeighborsClassifier()
knn_classifier.fit(X_train, Y_train)
```

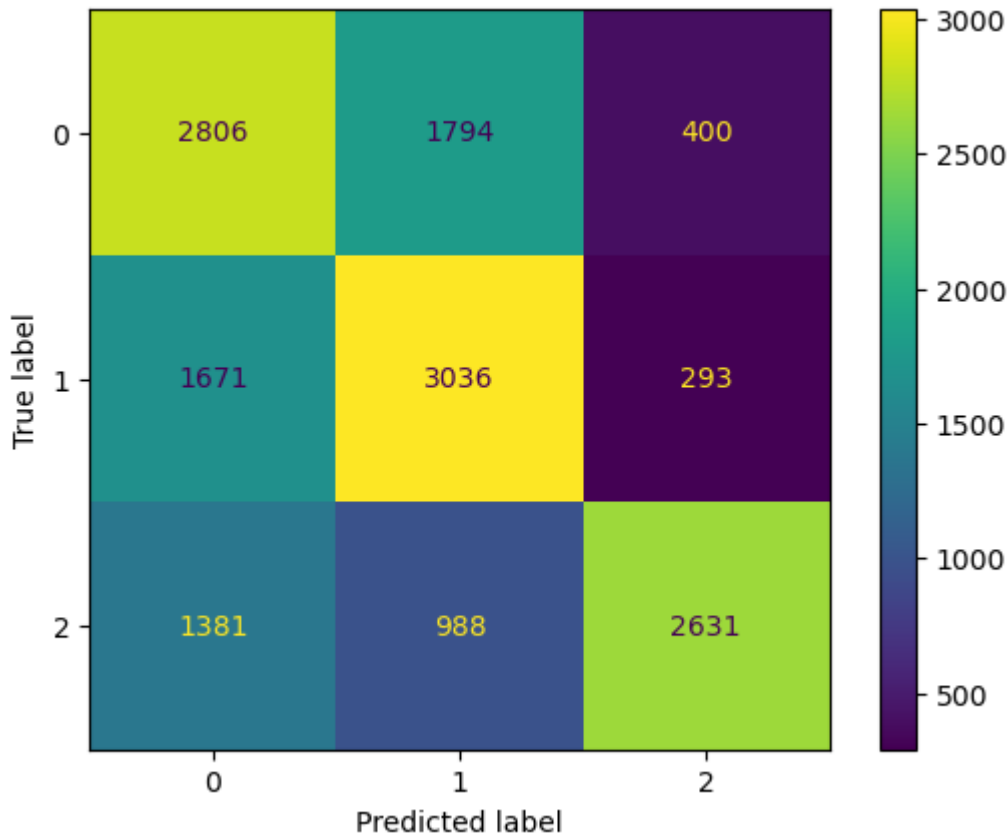
```
Out[32]: KNeighborsClassifier()
```

```
In [33]: knn_val_accuracy = knn_classifier.score(X_val, Y_val)
print(f"Validation Accuracy of KNN Clf. is: {(knn_val_accuracy)*100:.2f}%")
```

Validation Accuracy of KNN Clf. is: 56.49%

```
In [34]: sklearn.metrics.ConfusionMatrixDisplay.from_estimator(knn_classifier, X_val)
```

```
Out[34]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fcf850fe1f0>
```



Support Vector Classifier

Note - The cells for SVC have been commented because it takes too long to run.

```
In [35]: #svc_classifier = SVC(kernel = 'linear')
#svc_classifier.fit(X_train, Y_train)

In [36]: #print(f"Validation Accuracy of Support Vector Classifier is: {(svc_class

In [37]: #sklearn.metrics.ConfusionMatrixDisplay.from_estimator(svc_classifier, X_
```

Decision Tree Classifier

```
In [38]: dt_classifier = DecisionTreeClassifier()
dt_classifier.fit(X_train, Y_train)

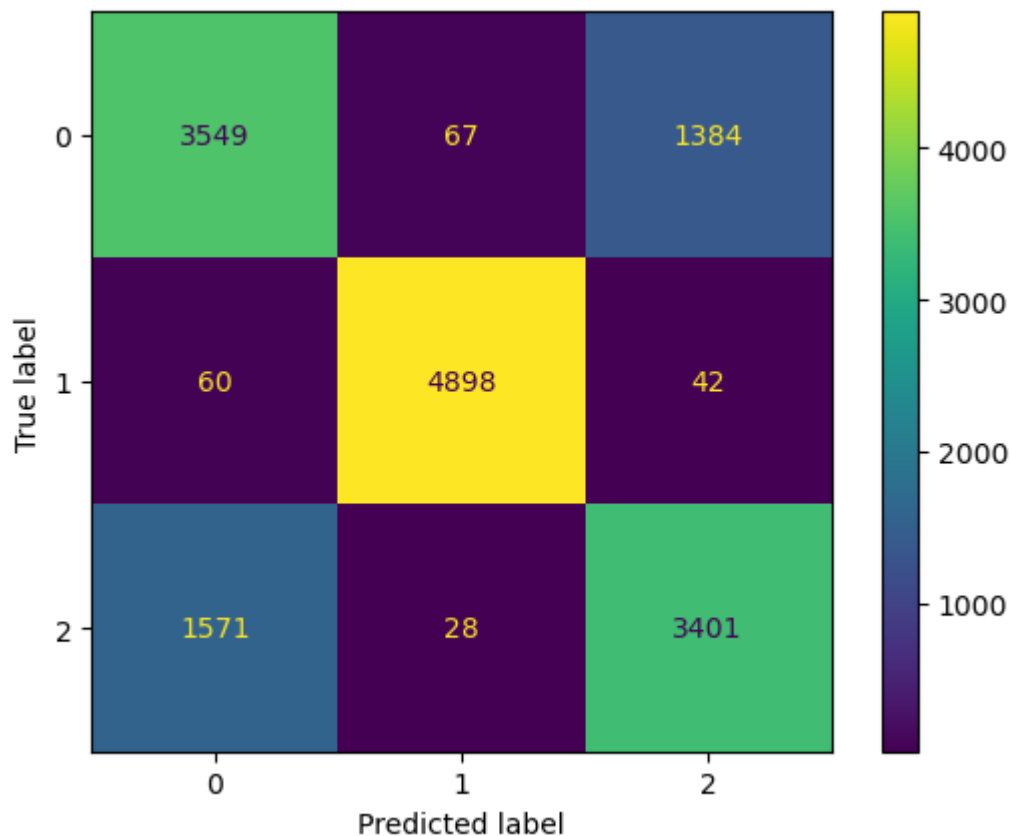
Out[38]: DecisionTreeClassifier()

In [39]: dt_val_accuracy = dt_classifier.score(X_val, Y_val)
print(f"Validation Accuracy of Decision Tree Clf. is: {(dt_val_accuracy)*

Validation Accuracy of Decision Tree Clf. is: 78.99%

In [40]: sklearn.metrics.ConfusionMatrixDisplay.from_estimator(dt_classifier, X_va

Out[40]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fcf
8af58b50>
```



Random Forest Classifier

```
In [41]: rf_classifier = RandomForestClassifier()
         rf_classifier.fit(X_train, Y_train)
```

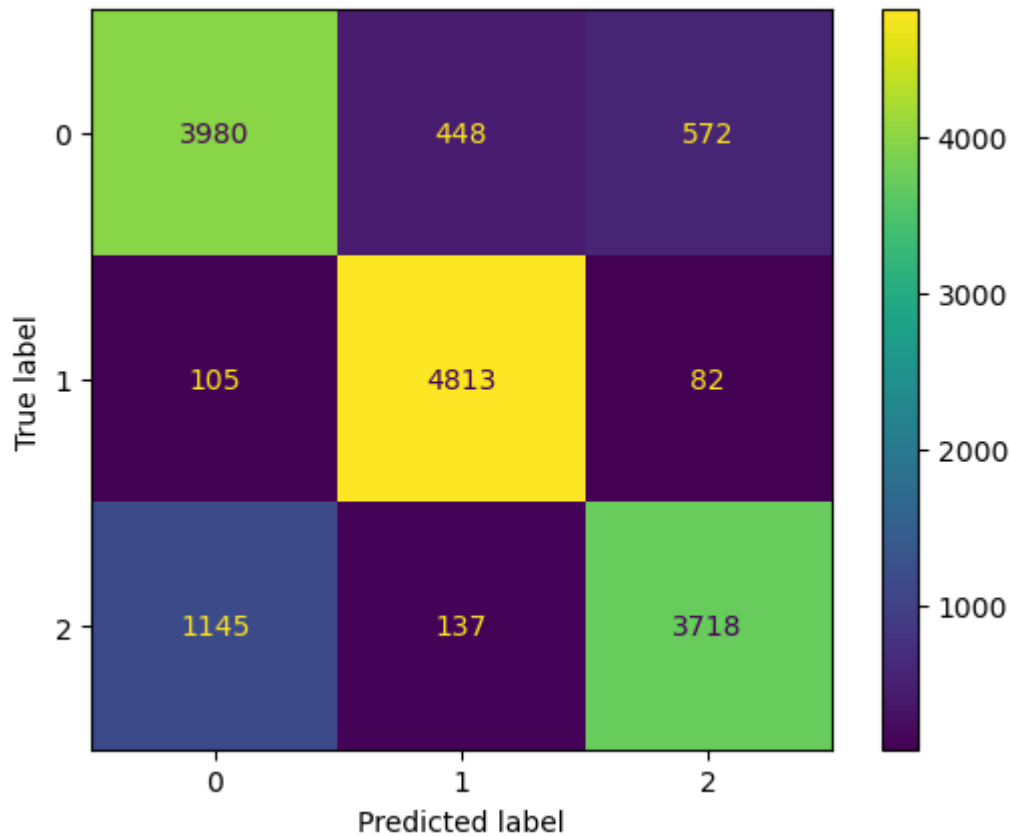
```
Out[41]: RandomForestClassifier()
```

```
In [42]: rf_val_accuracy = rf_classifier.score(X_val, Y_val)
         print(f"Validation Accuracy of Random Forest Classifier is: {(rf_val_accu
```

Validation Accuracy of Random Forest Classifier is: 83.41%

```
In [43]: sklearn.metrics.ConfusionMatrixDisplay.from_estimator(rf_classifier, X_val
```

```
Out[43]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fcf
         85ae73a0>
```



MLP Classifier

```
In [44]: mlp_classifier = MLPClassifier()
mlp_classifier.fit(X_train, Y_train)
```

/home/slowgamer/anaconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.py:699: UserWarning:

Training interrupted by user.

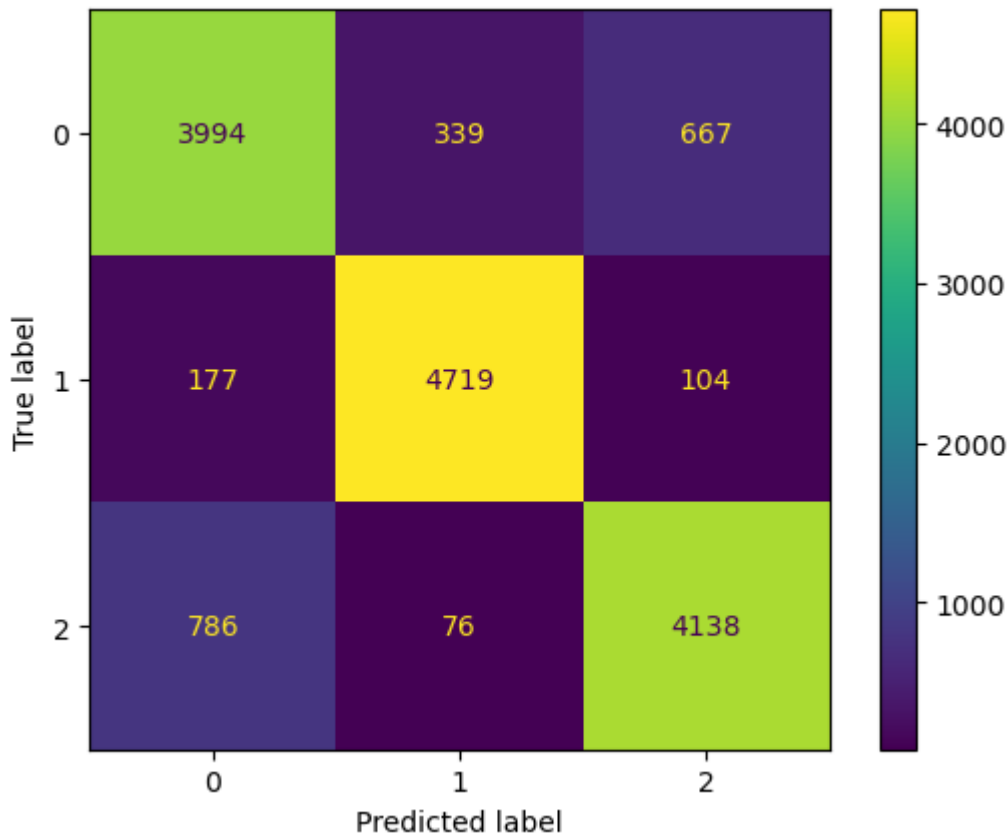
```
Out[44]: MLPClassifier()
```

```
In [45]: mlp_val_accuracy = mlp_classifier.score(X_val, Y_val)
print(f"Validation Accuracy of MLP Classifier is: {(mlp_val_accuracy)*100}%")
```

Validation Accuracy of MLP Classifier is: 85.67%

```
In [46]: sklearn.metrics.ConfusionMatrixDisplay.from_estimator(mlp_classifier, X_val, Y_val)
```

```
Out[46]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fcf
d6e925b0>
```



Scoring ML Model Performance

List of Validation Accuracies

```
In [47]: print(f"Validation Accuracy of Naive Bayes Classifier is: {(nb_val_accuracy)*100:.2f}%")
print(f"Validation Accuracy of Logistic Regression is: {(lr_val_accuracy)*100:.2f}%")
print(f"Validation Accuracy of KNN Clf. is: {(knn_val_accuracy)*100:.2f}%")
print(f"Validation Accuracy of Decision Tree Clf. is: {(dt_val_accuracy)*100:.2f}%")
print(f"Validation Accuracy of Random Forest Classifier is: {(rf_val_accuracy)*100:.2f}%")
print(f"Validation Accuracy of MLP Classifier is: {(mlp_val_accuracy)*100:.2f}%")
```

Validation Accuracy of Naive Bayes Classifier is: 78.73%
 Validation Accuracy of Logistic Regression is: 87.55%
 Validation Accuracy of KNN Clf. is: 56.49%
 Validation Accuracy of Decision Tree Clf. is: 78.99%
 Validation Accuracy of Random Forest Classifier is: 83.41%
 Validation Accuracy of MLP Classifier is: 85.67%

Conclusion

The following models performed the best -

1. Logistic Regression - 87.51%
2. MLP Classifier - 85.67%
3. Random Forest Classifier - 83.16%