inNutshell- Project Report

inNutshell Enterprise aims at benefiting people who work in the news industry who wish to categorize their content and publish it. Categorization helps users navigate or browse through collections, Web sites or search results. By grouping too many discrete items into understandable categories, users can quickly eliminate what is irrelevant or not interesting, and just pay attention to what matters most.

Our enterprise offers an additional feature for our client to summarize huge articles to provide the short glimpse of it to accomplish readers to dive deep into the article. For Summarization there are 2 approaches Extractive Summarization and Abstractive Summarization. Out of which we have used Extractive Summarization. This will be beneficial for both the users as well as the editors working for a News Industry.

We have implemented 6 models in our project for news classification and calculated their accuracy and performance metrics. For news summarization we have use 1 algorithm that is text extraction.

Project Goals

The Goal of our project is:

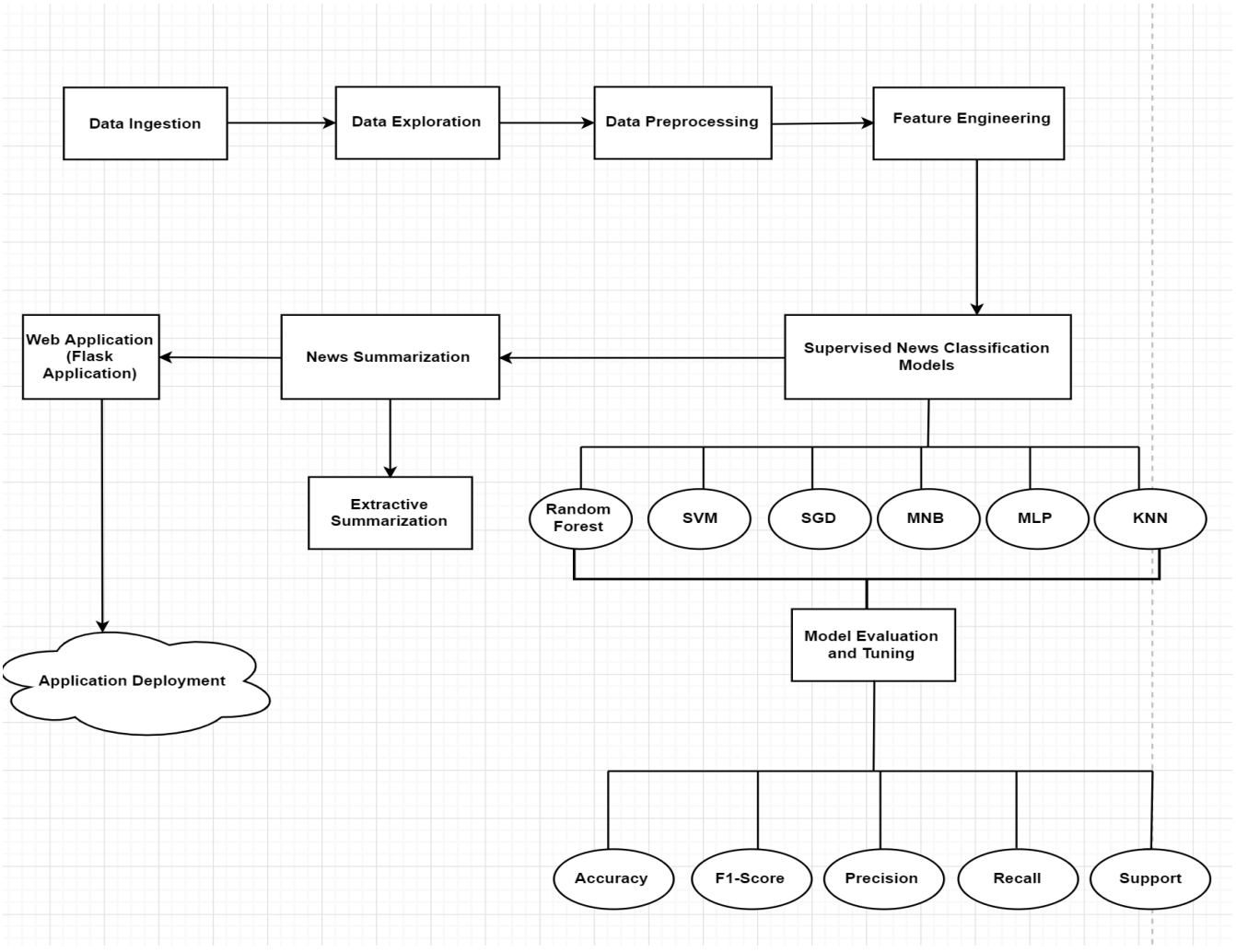
1) **News Classification:**

   For news classification we have used multiclass classification concept. Multiclass classification is a popular problem in supervised machine learning. In our dataset the classes used are different categories of news. The goal is to categorize the user's input news into category.

2) **News Summarization:**

   For news summarization we have used Natural Language Processing technique i.e. Extractive Text Summarization. The main aim of this module is to summarize the content of the news. To fetch the latest news from bbc.com in real-time and show it on the web page in a summarized format. This allows the user to stay updated with the current news in less time.

Pipeline Design

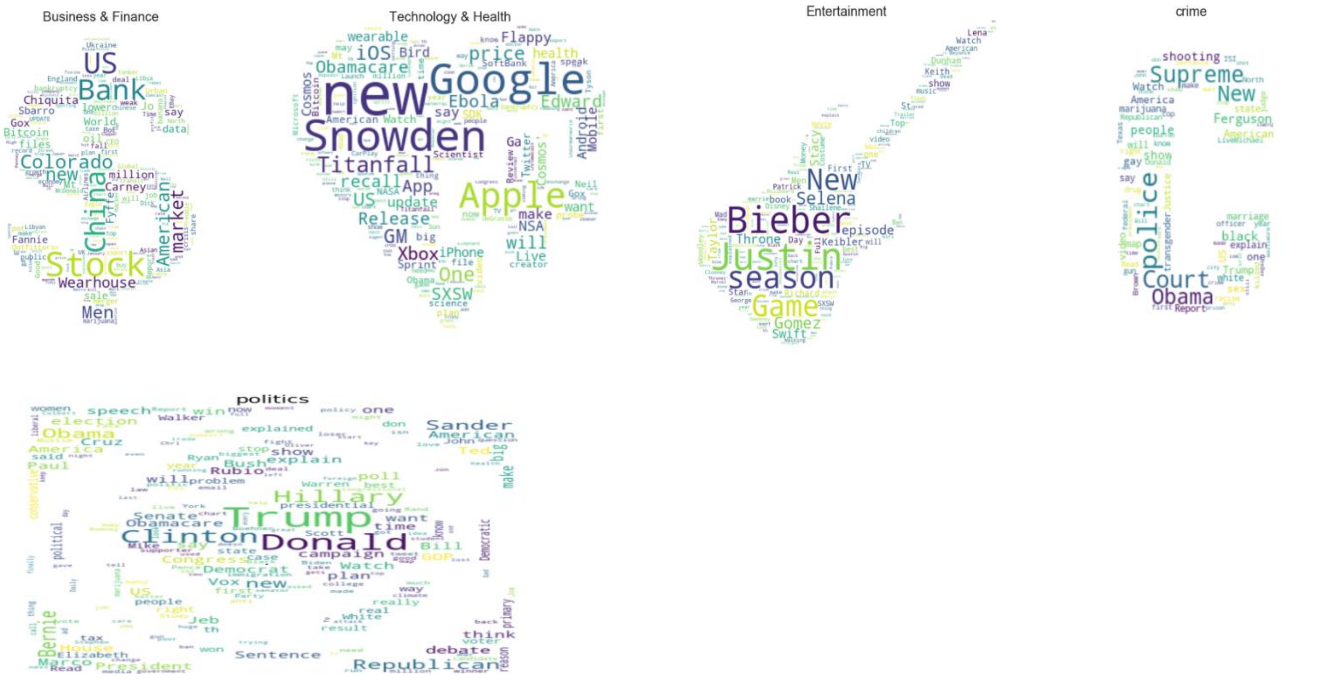


Implementation details

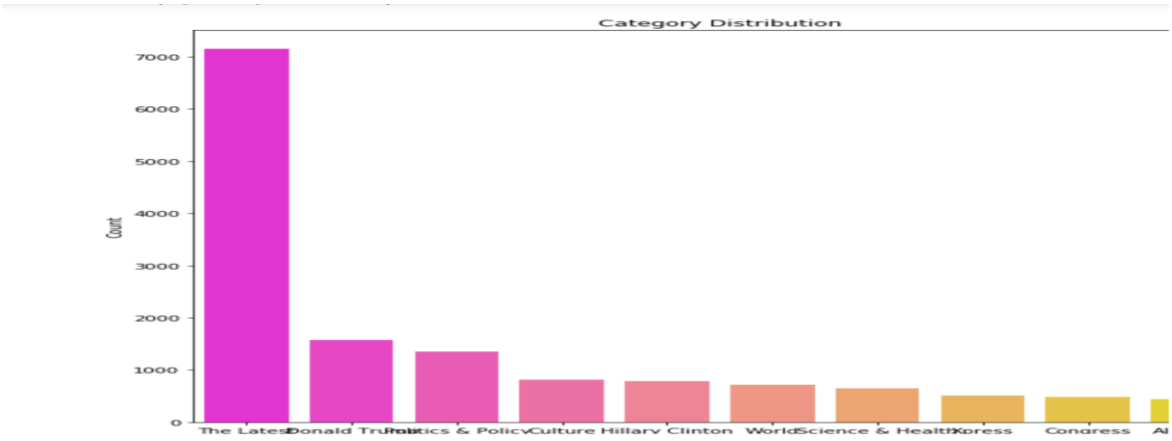
1) Data Ingestion and Data Exploration

Data ingestion is the process of obtaining and importing data for immediate use or storage in a database. To ingest something is to "take something in or absorb something." We had dataset of around 23k which consisted columns of title and category.

Word Cloud:



Category Distribution before Feature Engineering



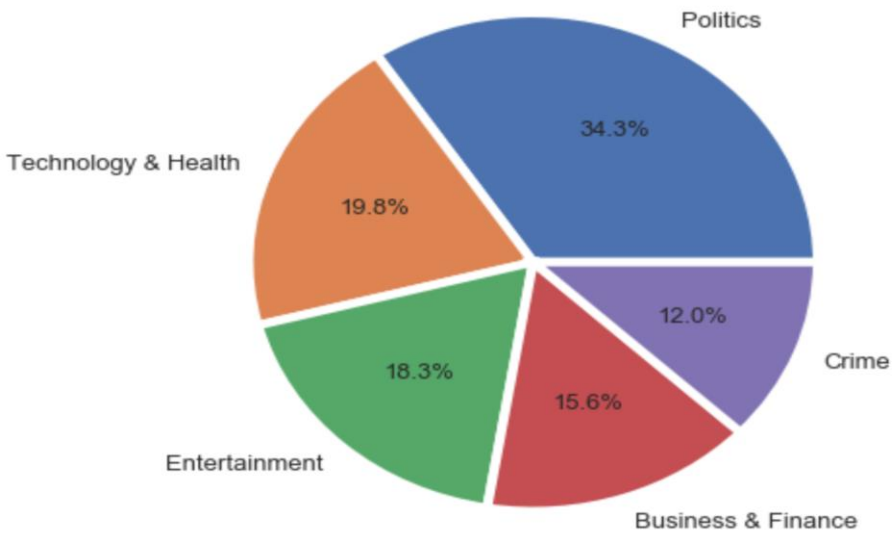
2) Data Preprocessing

In data Preprocessing we removed the rows containing missing values in title, categories and description. Removed the html tags from the description. Cleaned the data further by removing the Unicode error.

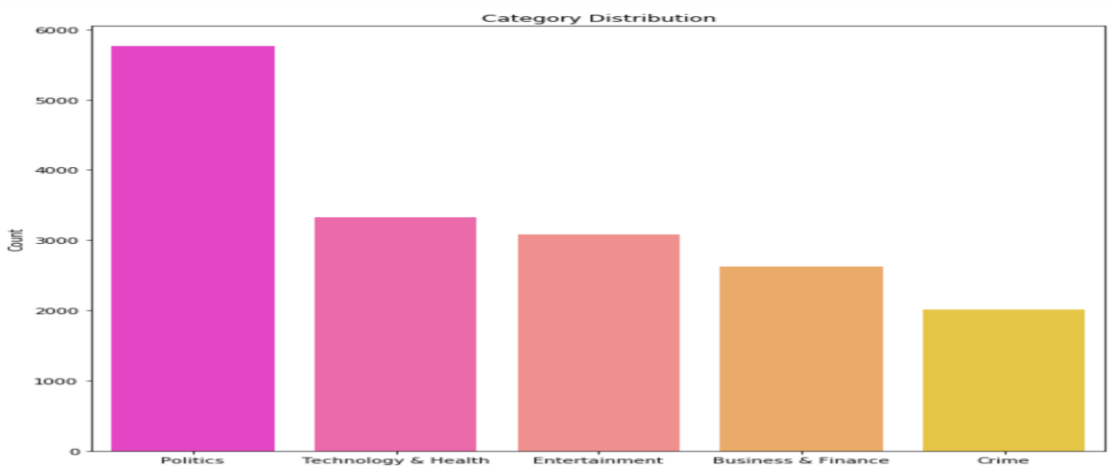
3) Feature Engineering

For feature engineering we created clusters for categories. Initially there were around 170 categories which cannot be considered as core categories. They were sub categories of some main categories. So we manually created 5 clusters and added all the related categories in those 5 categories.

Category Percentage:



Category Distribution after Feature Engineering:



4) Supervised News Classification Model

We have implemented total 6 models for Multiclass News Classification. The 6 models are:

a) Random Forest
b) Support Vector Machine (SVM)
c) Stochastic Gradient Descent (SGD)
d) Multinomial Naïve Bayes (MNB)
e) multilayer perceptron (MLP)
f) K-Nearest Neighbours (KNN)

5) Model Evaluation and tuning

We performed hyperparameter tuning on all the models to increase the accuracy of the model. Also calculated the performance metrics for all the models to check which model is considered as best.

6) News Summarization

For the purpose of News Summarization, we used Extractive Text Summarization. First, we converted the paragraphs into sentences, then we removed the stop-words and did stemming using porter-stemmer, followed by Tokenization, and lastly, we evaluated the weighted occurrence frequency of each word and finally we substituted the words with their weighted frequency which gave us the resultant as the summarized news.

7) Web Application

inNutShell - News Categorizer and Summarizer

Categorize your news and read it on go

Categorize your news and read it on go

CLASSIFY

BACK

Looking for latest news in Summarized Format?? CLICK ME!!

Categorize your news and read it on go

Spanish election: Socialists battle to stop right-wing surge

CLASSIFY

POLITICS

BACK

Summarized news

Sri Lanka bombings: PM Wickremesinghe says he was 'out of the loop'

He said Hashim led the attack on the popular tourist hotel, accompanied by a second bomber identified as "Ilham". Rights activists say the Ahmadi community, as foreign Muslims, could face reprisal attacks. The Sri Lankan government has said it will search schools, in one of a series of announcements designed to address public anxiety in the aftermath of the attacks. Two of the bombers are said to have been the sons of spice trader Mohammad Yusuf Ibrahim, one of Sri Lanka's richest men. Mr Ibrahim was detained and questioned after the attacks. of his sons was reportedly the bomber at the city's Shangri-La hotel - alongside Hashim, according to President Sirisena.

http://www.bbc.com/news/world-asia-48070552

Sri Lanka attacks: Sister of 'ringleader' deplores attack

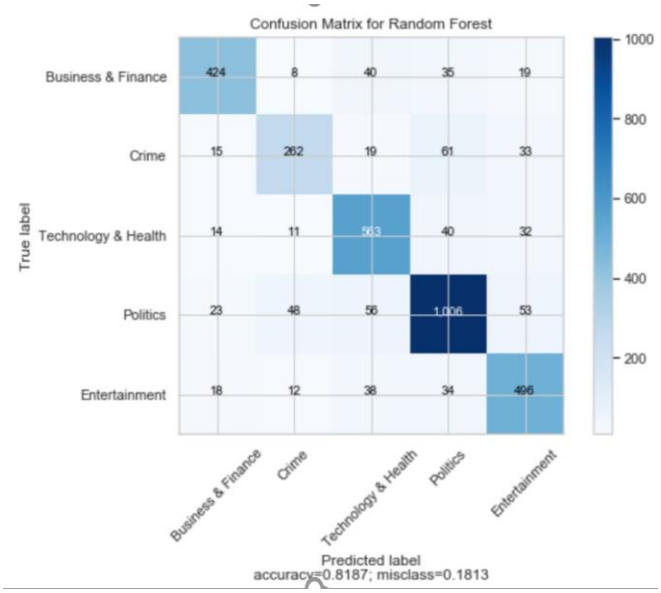
She is clearly unhappy with the attention that she is getting. She is the youngest of five siblings and Mr Hashim, believed to be around 40, the eldest. "I strongly deplore what he has done. Even if he is my brother, I cannot accept this. I don't care about him any more. "We had a

Analysis of models

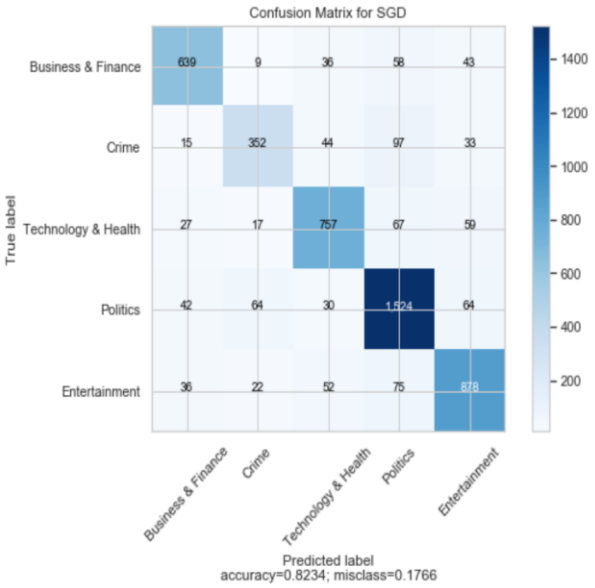
MODELS	ACCURACY	PRECISION	RECALL	F1-SCORE	SUPPORT
RANDOM FOREST	81.95	0.82	0.82	0.82	3360
SVM	80.88	0.81	0.81	0.81	5040
SGD	82.34	0.82	0.82	0.82	5040
MNB	80.55	0.81	0.81	0.81	5040
MLP	79.00	0.79	0.79	0.79	5040
KNN	78.27	0.8	0.79	0.79	3360

Confusion Matrix

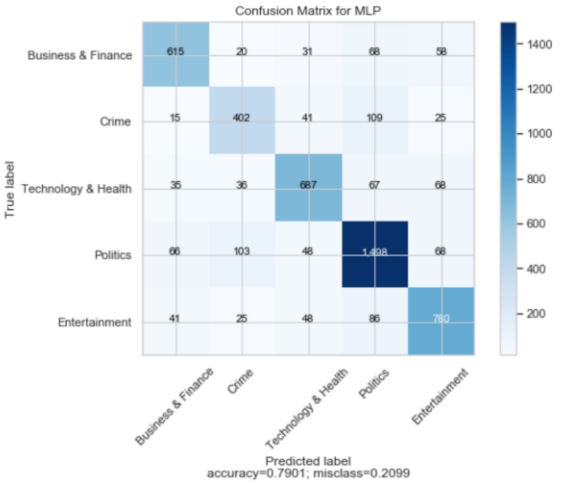
Random Forest



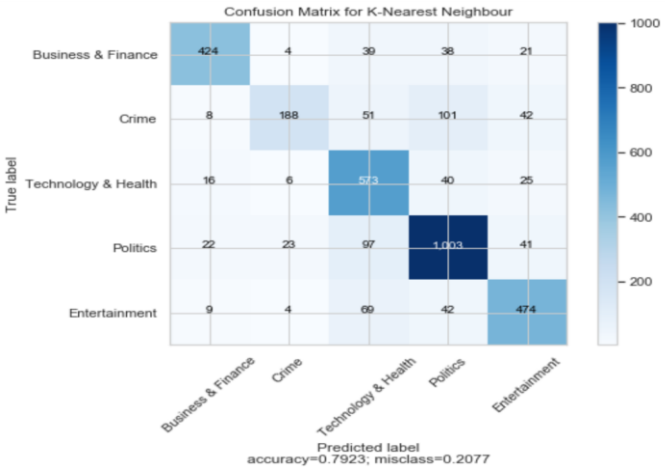
Stochastic Gradient Descent



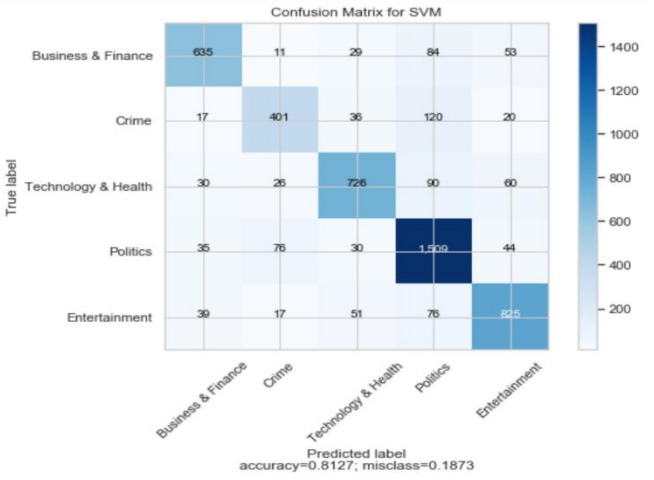
Multilayer Perceptron



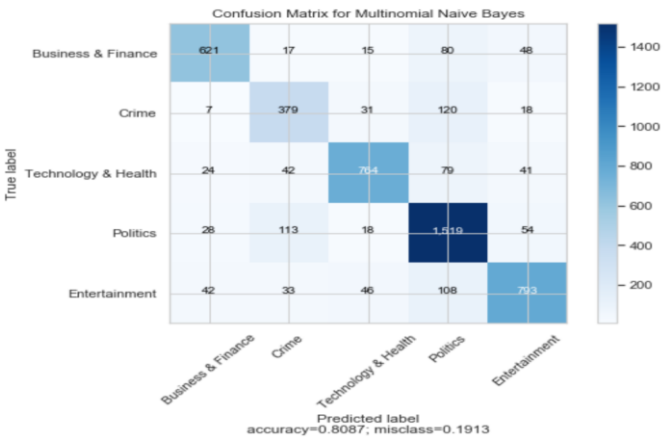
K-Nearest Neighbour



Support Vector Machines



Multinomial Naïve Bayes



Details on how to run the model

In order to get accurate results, we compared different classification models namely- Random Forest, SGD, SVM, DT, MLP, and MNB.

Our Approach towards running these models was as follows-

1. Loading the Dataset

```
In [1]:
import numpy as np
import pandas as pd

from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import CountVectorizer
```

```
In [2]:
news = pd.read_csv('newdatasetfinal.csv')
news.head()
```

Out[2]:

	title	NewCategory
0	Bitcoin is down 60 percent this year. Here's w...	Business & Finance
1	6 health problems marijuana could treat better...	Crime
2	9 charts that explain the history of global we...	Business & Finance
3	Remember when legal marijuana was going to sen...	Crime
4	Obamacare succeeded for one simple reason: it'...	Technology & Health

2. Vectorization

Vectorization

```
In [5]:
# convert data to vectors
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(news['title'])

y = news['NewCategory']
```

3. Splitting the Dataset into Training and Testing and then Training the model

Training SGD Classifier

```
In [7]:
sgd = SGDClassifier(n_jobs=-1, penalty='l2', n_iter=500, random_state=1234)

# hyperparameters for tuning
sgd_grid = [{'loss': ['hinge', 'log', 'squared_hinge'],
             'alpha': [0.0001, 0.0001, 0.00001]}]

# grid search with cross validation
sgd_search = GridSearchCV(estimator=sgd, param_grid=sgd_grid, cv=10, refit=True)
sgd_search.fit(X_train, y_train)
```

4. Tuning the Hyperparameters

Hyperparameter Tuning

[20]:
```
sgd = SGDClassifier(n_jobs=-1, penalty='elasticnet', n_iter=500, random_state=1234 )

# hyperparameters for tuning
sgd_grid = [{'loss': ['hinge', 'log', 'squared_hinge'],
             'alpha': [0.0001, 0.0001, 0.00001]}]

# grid search with cross validation
sgd_search = GridSearchCV(estimator=sgd, param_grid=sgd_grid, cv=50, refit=True)
sgd_search.fit(X_train, y_train)
```

5. Evaluating the performance of each model

Performance Metric

1. Accuracy

In [27]:
```
# Train a new classifier using the best parameters found by the grid search
print("Accuracy of SGD Classifier model:", sgd_search.best_estimator_.score(X_test, y_test)*100)
```

```
Accuracy of SGD Classifier model: 82.34126984126983
```

2. F1 Score

In [25]:
```
from sklearn.metrics import f1_score
print("F1 Score for all the categories:")
f1_score(y_test, y_pred, average=None)
```

```
F1 Score for all the categories:
```

Out[25]:
```
array([0.82772021, 0.70049751, 0.82015168, 0.85980254, 0.82056075])
```

3. Confusion Matrix

In [26]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

Out[26]:
```
array([[ 639,    9,   36,   58,   43],
       [  15,  352,   44,   97,   33],
       [  27,   17,  757,   67,   59],
       [  42,   64,   30, 1524,   64],
       [  36,   22,   52,   75,  878]], dtype=int64)
```

4. Precision-Recall

In [29]:
```
from sklearn.metrics import classification_report
```