**AI-Based Product Review Categorisation**

**Problem Understanding**

The objective of this feature is to help team prioritise improvements in the products quality and services.

This is important in e-commerce platform as it enables improvement in products and services, it improve the customer experience, and boosts the rating and retention, and also gives as an information in which part of the service the company needs to improve.

**Design Approach**

**Input data –**

Delivery rating

Quality rating

Customer Service rating

Overall rating

**Output data –**

Product review

**High level pipeline steps –**

Importing the libraries

Importing the dataset

Encoding the Dependent Variable

Splitting the dataset into the Training set and Test set

Feature Scaling

Training the K-NN model on the Training set

Predicting the Test set results

Evaluating Classification Models Performance

Predicting a new test result

**Python Code Implementation**

## Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

## Importing the dataset

dataset = pd.read\_csv('Dataset.csv')

X = dataset.iloc[:,1:].values

y = dataset.iloc[:,0].values

## Encoding the Dependent Variable

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

y = le.fit\_transform(y)

## Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)

## Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

## Training the K-NN model on the Training set

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors = 5)

classifier.fit(X\_train, y\_train)

## Predicting the Test set results

y\_pred = classifier.predict(X\_test)

## Evaluating Classification Models Performance

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

‘’’ [[TN, FP],

[FN, TP]] ‘’’

accuracy\_score(y\_test, y\_pred)

## Predicting a new results

print(classifier.predict(sc.transform([[6,7,7,8]])))

if classifier.predict(sc.transform([[6,7,7,8]])) == “0”:

print(“delivery-issue”)

else if if classifier.predict(sc.transform([[6,7,7,8]])) == “1”:

print(“quality-issue”)

else if classifier.predict(sc.transform([[6,7,7,8]])) == “2”:

print(“service-issue”)

else if classifier.predict(sc.transform([[6,7,7,8]])) == “3”:

print(“perfect”)

**Final Reflection Understanding**

This model is not ideal in production because scikit- learn is not ideal for big data (for example large no. of customers review).