In [1]:
```
## Author: Wong Yee En (feature ablation), Goh Boon Xiang (Hbase)
```

In [1]:
```
from pyspark.sql import SparkSession

spark = SparkSession\
        .builder\
        .appName("FeatureAblation")\
        .getOrCreate()
```

```
24/09/07 20:48:42 WARN Utils: Your hostname, WeirdSmile. resolves to a loopback addr
ess: 127.0.1.1; using 10.255.255.254 instead (on interface lo)
24/09/07 20:48:42 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another addr
ess
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(ne
wLevel).
24/09/07 20:48:43 WARN NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
```

Retrieve Train Test Data from HBase

In [3]:
```
sc = spark.sparkContext
sc.addFile("/home/student/G3_B/de_classes/data_storage/hbase_handler.py")
from hbase_handler import HBaseHandler
hbase_handler = HBaseHandler(host='localhost', port=9090)# Create an instance
```

In [4]:
```
train_df  = hbase_handler.retrieve_from_hbase('train_data')

# Show the retrieved DataFrame
train_df.show(5)
train_df.count()
```

```
24/09/07 20:48:54 WARN SparkSession: Using an existing Spark session; only runtime S
QL configurations will take effect.
```

```
+--------------------+---------+-------------+--------------------+---------------+
|              Review|Sentiment|SkuInfo_index|              tokens|number_of_tokens|
+--------------------+---------+-------------+--------------------+---------------+
|received thanks g...|        2|          1.0|[received, thanks...|              6|
|get coughing ever...|        0|          1.0|[get, coughing, e...|              8|
|            nice you|        2|          4.0|         [nice, you]|              2|
|translator object...|        2|          6.0|[translator, obje...|              7|
|       very thin ply|        0|          0.0|   [very, thin, ply]|              3|
+--------------------+---------+-------------+--------------------+---------------+
only showing top 5 rows
```

Out[4]: 1382

In [5]:
```
test_df  = hbase_handler.retrieve_from_hbase('test_data')

# Show the retrieved DataFrame
```

```
test_df.show(5)
test_df.count()
```

```
+--------------------+---------+------------+--------------------+---------------+
|              Review|Sentiment|SkuInfo_index|              tokens|number_of_tokens|
+--------------------+---------+------------+--------------------+---------------+
|                good|        2|         4.0|              [good]|              1|
|repeat purchase c...|        2|         3.0|[repeat, purchase...|              8|
|           less pack|        2|         0.0|        [less, pack]|              2|
|colour not same a...|        2|         6.0|[colour, not, sam...|              7|
|                good|        2|         1.0|              [good]|              1|
+--------------------+---------+------------+--------------------+---------------+
only showing top 5 rows
```

Out[5]: 644

In [6]:
```
train_df.groupBy('Sentiment').count().show()
```

```
+---------+-----+
|Sentiment|count|
+---------+-----+
|        2| 1216|
|        0|  166|
+---------+-----+
```

In [7]:
```
test_df.groupBy('Sentiment').count().show()
```

```
+---------+-----+
|Sentiment|count|
+---------+-----+
|        2|  575|
|        0|   69|
+---------+-----+
```

Close Hbase Connection

In [8]:
```
# Close the connection
hbase_handler.close()
```

Save train and test data files to json file as required by asgm

In [9]:
```
sc.addFile("../de_classes/data_storage/hadoop_file_handler.py")

# Import the HadoopFileHandler class
from hadoop_file_handler import HadoopFileHandler

# Create an instance of HadoopFileHandler
handler = HadoopFileHandler()
```

```
24/09/07 20:49:13 WARN SparkSession: Using an existing Spark session; only runtime S
QL configurations will take effect.
```

Write train_df to json (hadoop)

In [10]:
```
output_path = "data/train_test/train_json"
handler.write_json(train_df, output_path)
```

Write test_df to json (hadoop)

In [11]:
```
output_path = "data/train_test/test_json"
handler.write_json(test_df, output_path)
```

Read train_df from json(hadoop)

In [12]:
```
input_path = "data/train_test/train_json"
train_df = handler.read_json(input_path)
```

Read test_df from json(hadoop)

In [13]:
```
input_path = "data/train_test/test_json"
test_df = handler.read_json(input_path)
```

Vectorization (TD-IDF)

In [10]:
```
sc.addFile("../de_classes/data_preparation/data_transformation.py")

# Import the HadoopFileHandler class
from data_transformation import DataTransformations

# Create an instance of HadoopFileHandler
transform = DataTransformations()
```

In [11]:
```
tf_idf_train = DataTransformations.calculate_tfidf(train_df, tokens_col="tokens")
```

In [12]:
```
tf_idf_train.show()
```

```
+--------------------+---------+-------------+--------------------+---------------+
--------------------+--------------------+
|              Review|Sentiment|SkuInfo_index|              tokens|number_of_tokens|
rawFeatures|            features|
+--------------------+---------+-------------+--------------------+---------------+
--------------------+--------------------+
|received thanks g...|        2|          1.0|[received, thanks...|              6|
(10000,[3601,3958...|(10000,[3601,3958...|
|get coughing ever...|        0|          1.0|[get, coughing, e...|              8|
(10000,[1756,3506...|(10000,[1756,3506...|
|            nice you|        2|          4.0|         [nice, you]|              2|
(10000,[3370,4338...|(10000,[3370,4338...|
|translator object...|        2|          6.0|[translator, obje...|              7|
(10000,[293,2804,...|(10000,[293,2804,...|
|       very thin ply|        0|          0.0|   [very, thin, ply]|              3|
(10000,[2167,3944...|(10000,[2167,3944...|
|                  ok|        2|          0.0|                [ok]|              1|
(10000,[2645],[1.0])|(10000,[2645],[2....|
|fast delivery qua...|        2|         10.0|[fast, delivery, ...|              4|
(10000,[2645,3506...|(10000,[2645,3506...|
|fast delivery goo...|        2|         33.0|[fast, delivery, ...|              4|
(10000,[3601,6168...|(10000,[3601,6168...|
|  batch quality thin|        2|          1.0|[batch, quality, ...|              3|
(10000,[2167,3506...|(10000,[2167,3506...|
|well packed wrapp...|        2|         18.0|[well, packed, wr...|              7|
(10000,[157,3370,...|(10000,[157,3370,...|
|       delivery fast|        2|          8.0|    [delivery, fast]|              2|
(10000,[7128,9263...|(10000,[7128,9263...|
|waterproof breath...|        2|          1.0|[waterproof, brea...|              9|
(10000,[760,1382,...|(10000,[760,1382,...|
|                good|        2|         16.0|              [good]|              1|
(10000,[6168],[1.0])|(10000,[6168],[0....|
|fast delivery nic...|        2|         13.0|[fast, delivery, ...|             12|
(10000,[1140,1916...|(10000,[1140,1916...|
|received good con...|        2|          4.0|[received, good, ...|              3|
(10000,[3601,5328...|(10000,[3601,5328...|
|good products gre...|        2|          1.0|[good, product, g...|              8|
(10000,[447,750,1...|(10000,[447,750,1...|
|parcel has been r...|        2|          6.0|[parcel, ha, been...|             10|
(10000,[938,1687,...|(10000,[938,1687,...|
|i received goods ...|        2|          2.0|[i, received, goo...|              6|
(10000,[1687,1756...|(10000,[1687,1756...|
|all good fast del...|        2|          3.0|[all, good, fast,...|              9|
(10000,[80,867,41...|(10000,[80,867,41...|
|ok received good ...|        2|          4.0|[ok, received, go...|              4|
(10000,[2645,3601...|(10000,[2645,3601...|
+--------------------+---------+-------------+--------------------+---------------+
--------------------+--------------------+
only showing top 20 rows
```

In [13]:
```
tf_idf_train.select("Review", "features").show(truncate=True)
```

```
+--------------------+--------------------+
|              Review|            features|
+--------------------+--------------------+
|received thanks g...|(10000,[3601,3958...|
|get coughing ever...|(10000,[1756,3506...|
|            nice you|(10000,[3370,4338...|
|translator object...|(10000,[293,2804,...|
|       very thin ply|(10000,[2167,3944...|
|                  ok|(10000,[2645],[2....|
|fast delivery qua...|(10000,[2645,3506...|
|fast delivery goo...|(10000,[3601,6168...|
|  batch quality thin|(10000,[2167,3506...|
|well packed wrapp...|(10000,[157,3370,...|
|       delivery fast|(10000,[7128,9263...|
|waterproof breath...|(10000,[760,1382,...|
|                good|(10000,[6168],[0....|
|fast delivery nic...|(10000,[1140,1916...|
|received good con...|(10000,[3601,5328...|
|good products gre...|(10000,[447,750,1...|
|parcel has been r...|(10000,[938,1687,...|
|i received goods ...|(10000,[1687,1756...|
|all good fast del...|(10000,[80,867,41...|
|ok received good ...|(10000,[2645,3601...|
+--------------------+--------------------+
only showing top 20 rows
```

In [14]:
```
tf_idf_test = DataTransformations.calculate_tfidf(test_df, tokens_col="tokens")
```

In [15]:
```
tf_idf_test.select("Review", "features").show(truncate=True)
```

```
+--------------------+--------------------+
|              Review|            features|
+--------------------+--------------------+
|                good|(10000,[6168],[0....|
|repeat purchase c...|(10000,[80,747,40...|
|           less pack|(10000,[1195,6547...|
|colour not same a...|(10000,[2525,4041...|
|                good|(10000,[6168],[0....|
|good delivery goo...|(10000,[80,447,39...|
|good product fast...|(10000,[447,1738,...|
|condition okay go...|(10000,[747,1226,...|
|mask very thick a...|(10000,[3048,3944...|
|second time buy p...|(10000,[80,3446,8...|
|perfect size adul...|(10000,[72,520,13...|
|received good con...|(10000,[2067,3506...|
|goods have been s...|(10000,[1299,1485...|
|           terbaekkk|(10000,[5163],[5....|
|        ok good item|(10000,[1916,2645...|
|          well worth|(10000,[157,9679]...|
|good product fast...|(10000,[447,524,3...|
|very good seller ...|(10000,[1709,2478...|
|so bad i asked ma...|(10000,[141,387,1...|
|price cheap but b...|(10000,[1738,1989...|
+--------------------+--------------------+
only showing top 20 rows
```

Handle Class imbalance

In [16]:
```
tf_idf_train.groupBy('Sentiment').count().show()
```

```
+---------+-----+
|Sentiment|count|
+---------+-----+
|        2| 1216|
|        0|  166|
+---------+-----+
```

Oversampling

In [17]:
```
print("Initial Sentiment Counts:")

# Separate the majority and minority classes
major_df = tf_idf_train.filter(tf_idf_train.Sentiment == 2)
minor_df = tf_idf_train.filter(tf_idf_train.Sentiment == 0)

# Check the counts of each class
major_count = major_df.count()
minor_count = minor_df.count()

print(f"Majority class count (Sentiment 2): {major_count}")
print(f"Minority class count (Sentiment 0): {minor_count}")
```

```
Initial Sentiment Counts:
Majority class count (Sentiment 2): 1216
Minority class count (Sentiment 0): 166
```

In [18]:
```
oversampled_train = DataTransformations.oversample(tf_idf_train, label_col="Sentime
```

In [19]:
```
print("Combined Sentiment Counts after Oversampling:")
oversampled_train.groupBy('Sentiment').count().show()
```

```
Combined Sentiment Counts after Oversampling:
+---------+-----+
|Sentiment|count|
+---------+-----+
|        2| 1216|
|        0| 1162|
+---------+-----+
```

In [20]:
```
oversampled_train.printSchema()
```

```
root
 |-- Review: string (nullable = true)
 |-- Sentiment: integer (nullable = true)
 |-- SkuInfo_index: double (nullable = true)
 |-- tokens: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- number_of_tokens: integer (nullable = true)
 |-- rawFeatures: vector (nullable = true)
 |-- features: vector (nullable = true)
```

In [21]:
```
tf_idf_test.printSchema()
```

```
root
 |-- Review: string (nullable = true)
 |-- Sentiment: integer (nullable = true)
 |-- SkuInfo_index: double (nullable = true)
 |-- tokens: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- number_of_tokens: integer (nullable = true)
 |-- rawFeatures: vector (nullable = true)
 |-- features: vector (nullable = true)
```

Modeling + Feature Ablations

In [22]:
```
sc.addFile("../de_classes/modelling.py")

# Import the HadoopFileHandler class
from modelling import ModelTrainer

# Create an instance of HadoopFileHandler
modelTrainer = ModelTrainer()
```

Model 1

Feature: X1 = features(reviews after tf-idf)

In [23]:
```
# Prepare features with only TF-IDF features
train_df1= ModelTrainer.prepare_features(oversampled_train, feature_cols=["features
test_df1= ModelTrainer.prepare_features(tf_idf_test, feature_cols=["features"])

# Train the model and evaluate
model1 = ModelTrainer.train_model(train_df1, label_col="Sentiment")
predictions1, accuracy_1, precision_1, recall_1, f1_score_1 = ModelTrainer.evaluate
```

```
Accuracy: 0.8633540372670807
Precision: 0.8836300986058025
Recall: 0.8633540372670807
F1 Score: 0.8720196979522022
```

Model 2

Features: X1 = features, X2 = number_of_tokens

In [24]:
```
# Feature ablation with TF-IDF features and StarCount
train_df2 = ModelTrainer.prepare_features(oversampled_train, feature_cols=["feature
test_df2 = ModelTrainer.prepare_features(tf_idf_test, feature_cols=["features","num

model2 = ModelTrainer.train_model(train_df2, label_col="Sentiment")
predictions2, accuracy_2, precision_2, recall_2, f1_score_2 = ModelTrainer.evaluate
```

```
Accuracy: 0.8571428571428571
Precision: 0.8802215689205575
Recall: 0.8571428571428571
F1 Score: 0.8671032153227644
```

Model 3

Features: X1 = features, X2 = SkuInfo_index, X3 = number_of_tokens

In [25]:
```
# Prepare features
feature_columns = ["features", "SkuInfo_index","number_of_tokens"]
train_df3 = ModelTrainer.prepare_features(oversampled_train, feature_cols=feature_c
test_df3 = ModelTrainer.prepare_features(tf_idf_test, feature_cols=feature_columns)

model3 = ModelTrainer.train_model(train_df3, label_col="Sentiment")
predictions3, accuracy_3, precision_3, recall_3, f1_score_3 = ModelTrainer.evaluate
```

```
Accuracy: 0.8773291925465838
Precision: 0.8889660076891068
Recall: 0.8773291925465838
F1 Score: 0.8826397957280104
```

Comparison tables for all models

In [26]:
```
# Step 1: Create a list of tuples representing the metrics for each model
data = [
    ("Model1", accuracy_1, precision_1, recall_1, f1_score_1),
    ("Model2", accuracy_2, precision_2, recall_2, f1_score_2),
    ("Model3", accuracy_3, precision_3, recall_3, f1_score_3)
]

# Step 2: Create a DataFrame with columns for Model, Accuracy, Precision, Recall, F
df = spark.createDataFrame(data, ["Model", "Accuracy", "Precision", "Recall", "F1Sc

# Step 3: Register the DataFrame as a SQL temporary view
df.createOrReplaceTempView("model_metrics")
```

In [27]:
```
# Step 4: Use Spark SQL to query and present the data, rounding to 2 decimal places
spark.sql("""
    SELECT
        Model,
        ROUND(Accuracy, 4) AS Accuracy,
        ROUND(Precision, 4) AS Precision,
        ROUND(Recall, 4) AS Recall,
        ROUND(F1Score, 4) AS F1Score
    FROM model_metrics
""").show(truncate=False)
```

```
+------+--------+---------+------+-------+
|Model |Accuracy|Precision|Recall|F1Score|
+------+--------+---------+------+-------+
|Model1|0.8634  |0.8836   |0.8634|0.872  |
|Model2|0.8571  |0.8802   |0.8571|0.8671 |
|Model3|0.8773  |0.889    |0.8773|0.8826 |
+------+--------+---------+------+-------+
```

In [28]:
```
predictions = predictions3.select("Review", "prediction")
```

In [29]:
```
predictions.show(truncate = False)
```

```
+--------------------------------------------------------------------------+------
----+
|Review                                                                    |predic
tion|
+--------------------------------------------------------------------------+------
----+
|good                                                                      |2.0
|
|repeat purchase comfortable value buy fast shipping delivery              |2.0
|
|less pack                                                                 |2.0
|
|colour not same advertised meltblown between waterproof                   |0.0
|
|good                                                                      |2.0
|
|good delivery good products next time buy again thanks                    |2.0
|
|good product fast delivery satisfied good response reasonable price       |2.0
|
|condition okay good many times repeat order kids like                     |2.0
|
|mask very thick according face not how big                                |0.0
|
|second time buy praise                                                    |2.0
|
|perfect size adults comfortable fit stylish design convenient earloop design|2.0
|
|received good condition package quality looks good yet try                |2.0
|
|goods have been securely accepted number enough i am satisfied            |2.0
|
|terbaekkk                                                                 |2.0
|
|ok good item                                                              |2.0
|
|well worth                                                                |2.0
|
|good product fast delivery days cheaper product much affordable           |2.0
|
|very good seller responsive friendly highly recommended                   |2.0
|
|so bad i asked mask didn t go his nose split mask me i wanted change      |0.0
|
|price cheap but bit thin                                                  |2.0
|
+--------------------------------------------------------------------------+------
----+
only showing top 20 rows
```

In [30]:
```
# Filter predictions to show only rows where prediction is 0
predictions.filter(predictions.prediction == 0).show(truncate=False)
```

```
+--------------------------------------------------------------------------------
-----------------------------------------------------+----------+
|Review
|prediction|
+--------------------------------------------------------------------------------
-----------------------------------------------------+----------+
|colour not same advertised meltblown between waterproof
|0.0       |
|mask very thick according face not how big
|0.0       |
|so bad i asked mask didn t go his nose split mask me i wanted change
|0.0       |
|i ordered pcs there can be
|0.0       |
|quality not really abit thin but colour same pic what i expect
|0.0       |
|strong smell lar can t hold
|0.0       |
|very nice thank you seller
|0.0       |
|very thin claimed be korea but they are made china
|0.0       |
|less pack mask order pack up packs only
|0.0       |
|top seller always well done
|0.0       |
|i order pcs bt i get pcs only i wan refund
|0.0       |
|thin
|0.0       |
|toooo thin maak
|0.0       |
|received but top not zipped properly order light green but colour difference what s
sell also dark red also pink                          |0.0       |
|your mask not good hurt people all
|0.0       |
|seller did not send blue color mask hope buy again
|0.0       |
|normal quality delivery fast but wrong colour given bought navy but given other blu
e pls double check again so same mistake can be prevent|0.0       |
|goods are up good condition t thank you pd sell flash my delivery fast
|0.0       |
|shop happy ja son purchase deliver quickly
|0.0       |
|less one pack pink mask bought packs pink mask but only get packs masks have strong
chemical smell poor quality                           |0.0       |
+--------------------------------------------------------------------------------
-----------------------------------------------------+----------+
only showing top 20 rows
```

In [31]:
```python
sc.addFile("../de_classes/data_storage/hadoop_file_handler.py")

# Import the HadoopFileHandler class
from hadoop_file_handler import HadoopFileHandler
```

```
# Create an instance of HadoopFileHandler
handler = HadoopFileHandler()
```

```
24/09/07 20:52:52 WARN SparkContext: The path ../de_classes/data_storage/hadoop_file
_handler.py has been added already. Overwriting of added paths is not supported in t
he current version.
```

In [32]:
```
# Save the entire DataFrame to a JSON file
handler.write_json(predictions, "data/predictions/predictions3.json")
```

In [33]:
```
spark.stop()
```

In [ ]:
```
```