1. Data processing steps-

- I kept only two columns for my analysis - "text" and "gender"
- using na.drop I dropped all rows containing null or Nan values.
- From the above remaining rows I filtered rows having gender values as - male, female and brand. The remaining empty or unknown values in gender column were dropped.
- There were a lot of junk characters in text column. I used UDF named "removeSpecialCharacters" to remove any junk characer in text column. The result achieved acfter appliying udf is shown below:

```
After splitting csv: 2 : 16413
+------+--------------------+
|gender|               text1|
+------+--------------------+
|  male|Robbie E Responds...|
|  male|   It felt like t...|
|  male|i absolutely ador...|
|  male|Hi  JordanSpieth ...|
|female|Watching Neighbou...|
|female|Ive seen people o...|
| brand| BpackEngineer Th...|
|  male|Gala Bingo clubs ...|
|female|  Aphmau  the pic...|
|female| Evielady just ho...|
+------+--------------------+
only showing top 10 rows
```

- How I invoked the udf is shown below:

```
//remove unwanted characters from description and text columns
        sparkSession.udf().register("removeChar", removeSpecialCharacters, DataTypes.StringType);
        //csv = csv.withColumn("description1", callUDF("removeChar", csv.col("description"))).drop("description");
        csv = csv.withColumn("text1", callUDF("removeChar", csv.col("text"))).drop("text");
```

- udf function is shown below:

```
private static UDF1 removeSpecialCharacters = new UDF1<String, String>() {
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    public String call(final String str) throws Exception {
        return str.replaceAll("[^a-zA-Z0-9]", " ");
    }
};
```

- After applying udf on text column, there might be some columns which would have become empty (if they only had junk characters). To overcome this situation, I used below

```
csv = csv.filter(csv.col("text1").notEqual(""));
csv = csv.filter(csv.col("text1").isNotNull());
```

2. Model Building-

- I have used - (i) Random forest Model and (ii) Decision Tree
- columns used to train both the models: text and gender
- hyperparameters used - none

3. Evaluation Metrics-

- The result of evaluation metrics is shown below:

```
Random Forest Training Accuracy = 45 %
Random Forest Training weightedPrecision = 0
Random Forest Training weightedRecall = 0
Random Forest Training f1 = 0
Random Forest Test Accuracy = 42 %
Random Forest Test weightedPrecision = 0
Random Forest Test weightedRecall = 0
Random Forest Test f1 = 0
Decision Tree Training Accuracy = 51 %
Decision Tree Training weightedPrecision = 1
Decision Tree Training weightedRecall = 1
Decision Tree Training f1 = 1
Decision Tree Test Accuracy = 48 %
Decision Tree Training weightedPrecision = 0
Decision Tree Training weightedRecall = 0
Decision Tree Training f1 = 0
```

- My model is not facing any issue of overfitting or underfitting. There is a slight difference between Accuracy of training and test data but that is about 3% in both the cases which is acceptable.

4. Inferences and Suggestions-

1. Advantages of Random Forest-

- Random forests have less variance than a single decision tree. It means that it works correctly for a large range of data items than single decision trees.
- Random forests are extremely flexible and have very high accuracy.
- They also do not require preparation of the input data. We do not have to scale the data.
- It also maintains accuracy even when a large proportion of the data are missing.

Disadvantages of Random Forest-

- The main disadvantage of Random forests is their complexity. They are much

harder and time-consuming to construct than decision trees.

- They also require more computational resources and are also less intuitive. When you have a large collection of decision trees it is hard to have an intuitive grasp of the relationship existing in the input data.
- In addition, the prediction process using random forests is time-consuming than other algorithms.

Advantages of Decision Tress-

- Decision Trees are easy to explain. It results in a set of rules.
- It follows the same approach as humans generally follow while making decisions.
- Interpretation of a complex Decision Tree model can be simplified by its visualizations. Even a naive person can understand logic.
- The Number of hyper-parameters to be tuned is almost null.

Disadvantages of Decision Tress-

- There is a high probability of overfitting in Decision Tree.
- Generally, it gives low prediction accuracy for a dataset as compared to other machine learning algorithms.
- Information gain in a decision tree with categorical variables gives a biased response for attributes with greater no. of categories.
- Calculations can become complex when there are many class labels.

2. Improvisation if any-

In my model, I spent a lot of time to combine the description and text columns to form one column of features using VectorAssembler but I could not succeed. Ultimately I commented that part in myfinal deliverable to mnake it run without errors. **Hence I believe, that the combination of both text and description column would give better accuracy.**