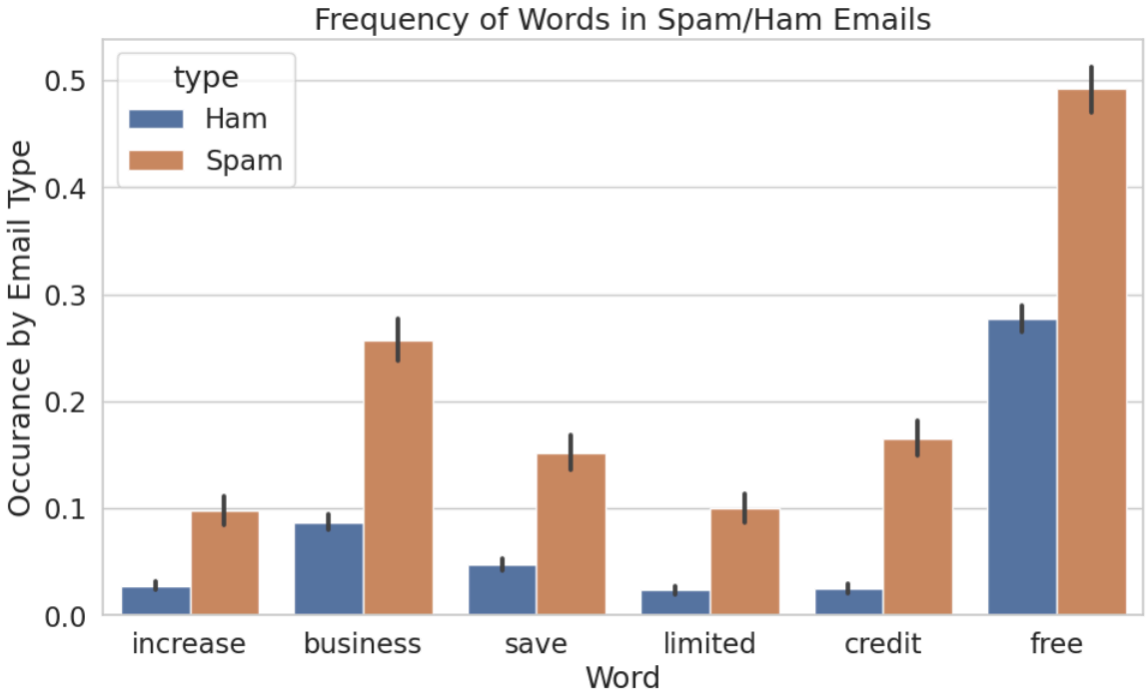
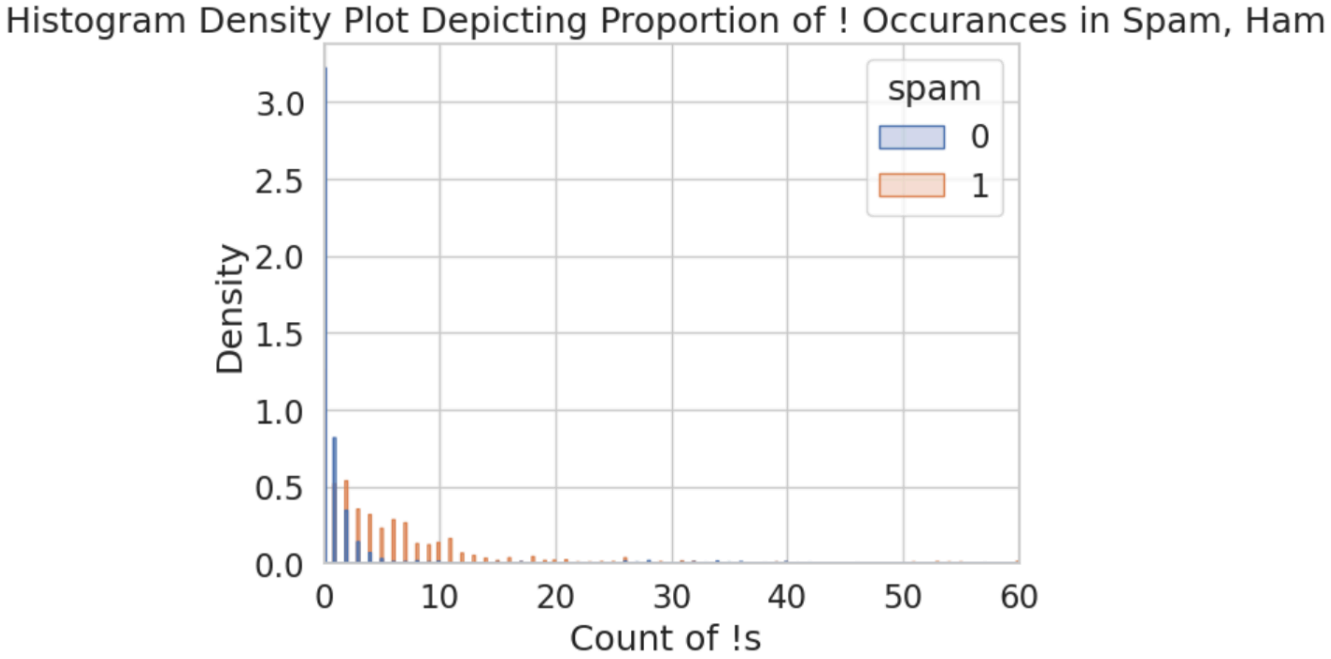
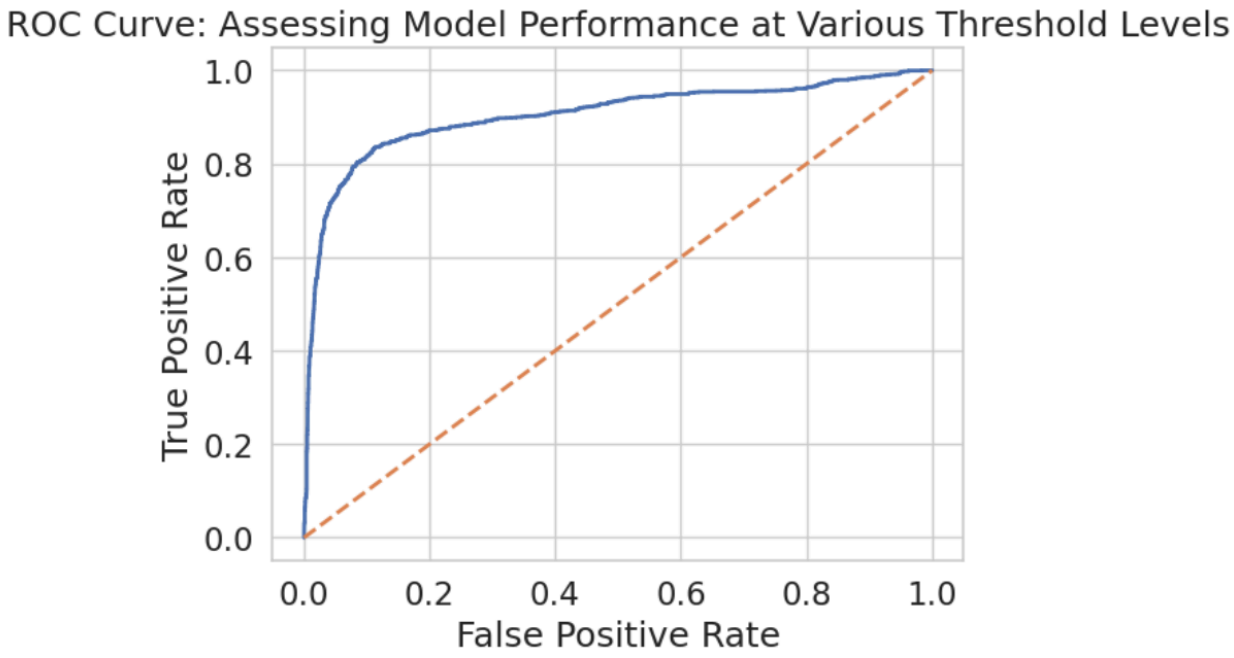

Frequency of Words in Spam/Ham Emails
type
Ham
Spam
Occurance by Email Type
0.0
0.1
0.2
0.3
0.4
0.5
increase
business
save
limited
credit
free
Word


Histogram Density Plot Depicting Proportion of ! Occurances in Spam, Ham
spam
0
1
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
10
20
30
40
50
60
Count of !s




```
# Define your processing function, processed data, and model here.
# You may find it helpful to look through the rest of the questions first!
def processingFunction(data, spamWords):
    for word in spamWords:
        data[word + '_count'] = 0
        data[word + '_count'] = data['email'].apply(lambda x: countWords(x, word))

    data["!count"] = data["email"].str.count("!")
    data["emailWordCt"] = data["email"].apply(len)

    for word in ["spam", "id","subject","email"]:
        if word in data.columns:
            data = data.drop(columns = [word])

    return data
def countWords(text, word):
    return text.count(word)
someWords= ["free","win","selected","winner", "credit","click","urgent", "100%",
            "#1", "trial", "earn", "guarantee", "viagra", "dick","ad ", "scam",
            "income","url", "div", "font", "body", "link",
            "html", "http", "head", "hot", "debt",
            "prize", "$","\n", "==","type"]
myModel = LogisticRegression(C=.3, max_iter=10000, solver="liblinear")

X_train = processingFunction(train,someWords)
Y_train = np.array(train["spam"])
myModel.fit(X_train,Y_train)
```

```
LogisticRegression(C=0.3, max_iter=10000, solver='liblinear')
```