0.0.1 Question 1: Feature/Model Selection Process

In this following cell, describe the process of improving your model. You should use at least 2-3 sentences each to address the follow questions:

1. How did you find better features for your model?
2. What did you try that worked or didn't work?
3. What was surprising in your search for good features?

1. Analysing spams and trying to find common and frequently appearing words. Testing selected words by fitting the model with them and run on validation set to see if there is any improvement.
2. Words like "i", "we" are mostly useless. Since they are very common in both spam and ham emails. Words like "promotion", "sale", "last chance" are usually useful because they are normally connected with spams like ads.
3. _____

Question 2a Generate your visualization in the cell below.

```
In [15]: ...
```

```
Out[15]: Ellipsis
```

Question 2b Write your commentary in the cell below.

Type your answer here, replacing this text.

0.0.2 Question 3: ROC Curve

In most cases we won't be able to get 0 false positives and 0 false negatives, so we have to compromise. For example, in the case of cancer screenings, false negatives are comparatively worse than false positives — a false negative means that a patient might not discover that they have cancer until it's too late, whereas a patient can just receive another screening for a false positive.

Recall that logistic regression calculates the probability that an example belongs to a certain class. Then, to classify an example we say that an email is spam if our classifier gives it ≥ 0.5 probability of being spam. However, *we can adjust that cutoff*: we can say that an email is spam only if our classifier gives it ≥ 0.7 probability of being spam, for example. This is how we can trade off false positives and false negatives.

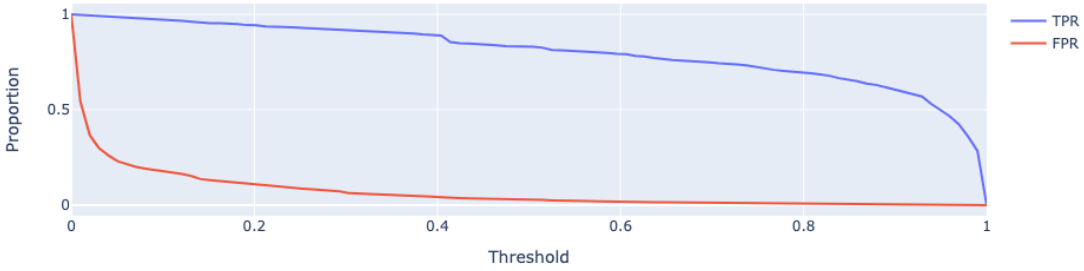
The ROC curve shows this trade off for each possible cutoff probability. In the cell below, plot a ROC curve for your final classifier (the one you use to make predictions for Gradescope) on the training data. Refer to Lecture 20 to see how to plot an ROC curve.

Hint: You'll want to use the `.predict_proba` method for your classifier instead of `.predict` so you get probabilities instead of binary predictions.

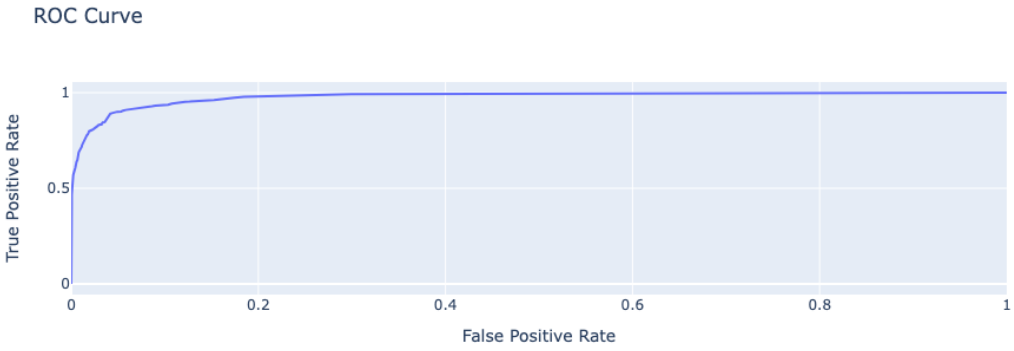
In [29]:
```
def predict_threshold(model, X, T):
    prob_one = model.predict_proba(X)[:, 1]
    return (prob_one >= T).astype(int)


def tpr_threshold(X, Y, T): # this is recall
    Y_hat = predict_threshold(model, X, T)
    return np.sum((Y_hat == 1) & (Y == 1)) / np.sum(Y == 1)

def fpr_threshold(X, Y, T):
    Y_hat = predict_threshold(model, X, T)
    return np.sum((Y_hat == 1) & (Y == 0)) / np.sum(Y == 0)
```

In [30]:
```
# compute for different thresholds on train set
thresholds = np.linspace(0, 1, 100)
tprs = [tpr_threshold(X_train, Y_train, t) for t in thresholds]
fprs = [fpr_threshold(X_train, Y_train, t) for t in thresholds]
```

In [34]:
```
import plotly.graph_objects as go
fig = go.Figure()
fig.add_trace(go.Scatter(name = 'TPR', x = thresholds, y = tprs))
fig.add_trace(go.Scatter(name = 'FPR', x = thresholds, y = fprs))
fig.update_xaxes(title="Threshold")
fig.update_yaxes(title="Proportion")
```



In [37]:
```
import plotly.express as px
fig = px.line(x=fprs, y = tprs, hover_name=thresholds, title="ROC Curve")
fig.update_xaxes(title="False Positive Rate")
fig.update_yaxes(title="True Positive Rate")
fig
```




```
Error in atexit._run_exitfuncs:
Traceback (most recent call last):
  File "/opt/conda/lib/python3.9/site-packages/popularity_contest/reporter.py", line 105, in report_popu
    libraries = get_used_libraries(initial_modules, current_modules)
  File "/opt/conda/lib/python3.9/site-packages/popularity_contest/reporter.py", line 74, in get_used_lib
    all_packages = get_all_packages()
```

```
  File "/opt/conda/lib/python3.9/site-packages/popularity_contest/reporter.py", line 51, in get_all_pacl
    for f in dist.files:
TypeError: 'NoneType' object is not iterable
```