**PROBLEM 1**

• Using Python, read in the 2 clickbait datasets (See section DATA), and combine both into a

single, shuffled dataset. (One function to shuffle data is numpy.random.shuffle)

• Next, split your dataset into **train, test, and validation** datasets. Use a split of 72% train, 8%

validation, and 20% test. (Which is equivalent to a 20% test set, and the remainer split 90%/10%

for train and validation).

• Estimation of **Target Rate**

**PROBLEM 2 –** Baseline Performance

• Assume you have a trivial baseline classifier that flags **every** text presented to it as clickbait.

What is the **precision, recall, and F1-score** of such a classifier on your test set? Do you think

there is another **good baseline classifier** that would give you higher F-1 score?

**PROBLEM 3** – Training a single Bag-of-Words (BOW) Text Classifier

train a BOW naïve bayes model.

classes CountVectorizer and MultinomialNB. Include both unigrams and bigrams in

your model in your vectorizer vocabulary

Compute the precision, recall, and F1-score on both your training and validation datasets using

functions in sklearn.metrics.

**PROBLEM 4**

Using the ParameterGrid class, run a small grid search where you vary at least 3 parameters of your

model

• **max\_df** for your count vectorizer (threshold to filter document frequency)

• **alpha** or smoothing of your NaïveBayes model

• One other parameter of your choice. This can be non-numeric; for example, you can consider a

model with and without bigrams (see parameter "ngram" in class CountVectorizer)

Show metrics on your **validation** set for precision, recall, and F1-score. If your grid search is very large

(>50 rows) you may limit output to the highest and lowest results.

**PROBLEM 6 –** Key Indicators

Using the log-probabilities of the model you selected in the previous problem, select **5 words** that are

strong **Clickbait indicators**. That is, if you needed to filter headlines based on a single word, without a

machine learning model, then these words would be good options. Show this list of keywords in your

notebook.

You can choose how to handle bigrams (e.g., "win<space>big"); you may choose to ignore them and

only select unigram vocabulary words as key indicators.

**PROBLEM 7 –** Regular expressions

Your IT department has reached out to you because they heard you can help them find clickbait. They

are interested in your machine learning model, but they need a solution today.

• Write a regular expression that checks if any of the keywords from the previous problem are

found in the text. You should write one regular expression that detects any of your top 5

keywords. Your regular expression should be aware of word boundaries in some way. That is,

the keyword "win" should not be detected in the text "Gas prices up in winter months".

• Using the python re library – apply your function to your test set. (See function re.search). What

is the precision and recall of this classifier? Show your results in your notebook