Problem 1

1A: Bag-of-Words Design Decision Description

We used the Bag-of-Words (BoW) representation for the text data in provided training files for use with the Logistic Regression classifier in our pipeline. To implement this, we used SciKitLearn's CountVectorizer tool from the feature_extraction library to perform preprocessing (data cleaning and vocabulary selection) on the training text data. Our analyzer of choice was 'words' - this broke down the text into words for feature analysis.

To remove any case variations, all text was converted to lowercase by setting the lowercase parameter to True. Common words (such as 'a', 'the', 'and', etc.) were eliminated using the stop words list built into CountVectorizer by setting stop_words to 'english'. We also chose to remove any punctuation and numbers with the token_pattern parameter (Regex: r'\b\w+\b') to focus only on the words.

After the data cleaning, we used parameters within CountVectorizer to select our vocabulary. We utilized the max_df parameter to ignore words with a frequency strictly higher than 50% to prevent overfitting to the training dataset. We did not set a min_df limit because rarer words could carry valuable information and we did not want to reduce our vocabulary cohort further. Though this vocabulary size varied across different folds of the cross validation, it consisted of roughly 500 to 1000 words. Due to the nature of the BoW approach, words that appear in the validation and test datasets but not the training dataset are ignored because the model is trained solely on the unique words in the training dataset. Provided that the training set contains a diverse vocabulary and is large enough, the impact on the accuracy of this model could be minimal.

1B: Cross Validation Design Description

We used GridSearchCV, a SciKitLearn tool from the model selection package, to perform a **5-fold cross validation** (CV) across the a range of a **hyperparameter C**, the inverse of regularization strength. We first split the training dataset into training and validation, holding out 20% for validation using train_test_split from SciKitLearn. Then, to implement the 5-fold CV, GridSearchCV breaks the training dataset into 5 equal folds (⅕ of the 1920 line training dataset, ~384 lines), where 4 of the 5 folds are used for training and 1 fold is used for validation. This process is repeated 5 times and the fold held out for validation changes with each iteration. The results across all folds were then averaged to evaluate the model. The performance metric used to evaluate the hyperparameter and data was Area Under the Receiver Operator Curve (AUROC). GridSearchCV has built in functionality to specify the pipeline (which contains the CountVectorizer preprocessor and logistic regression classifier) the number of folds for cross validation, hyperparameter (C) grid, and scoring parameters (AUROC). After using CV to identify the best C value, we will make predictions on the test set using this hyperparameter.

1C: Hyperparameter Selection for Logistic Regression Classifier

Logistic regression with an L2 penalty was used as the classifier for this model to prevent overfitting to the training data. This promotes a model that generalizes better to additional test data. During training, **no convergence issues** were found as the max iterations in the logistic regression classifier was set to 10,000, allowing enough iterations to cover the dataset. To further avoid overfitting to the training data, the 5-fold CV ensured performance on multiple splits of the dataset.

The best **C (inverse of regularization strength)** was found using a grid of **20-log-spaced values ranging from 10e-10 to 10e10** to explore a wide range of regularization strengths. A higher C value corresponds to weaker penalization and a lower C value corresponds to a stronger penalization (more regularization).

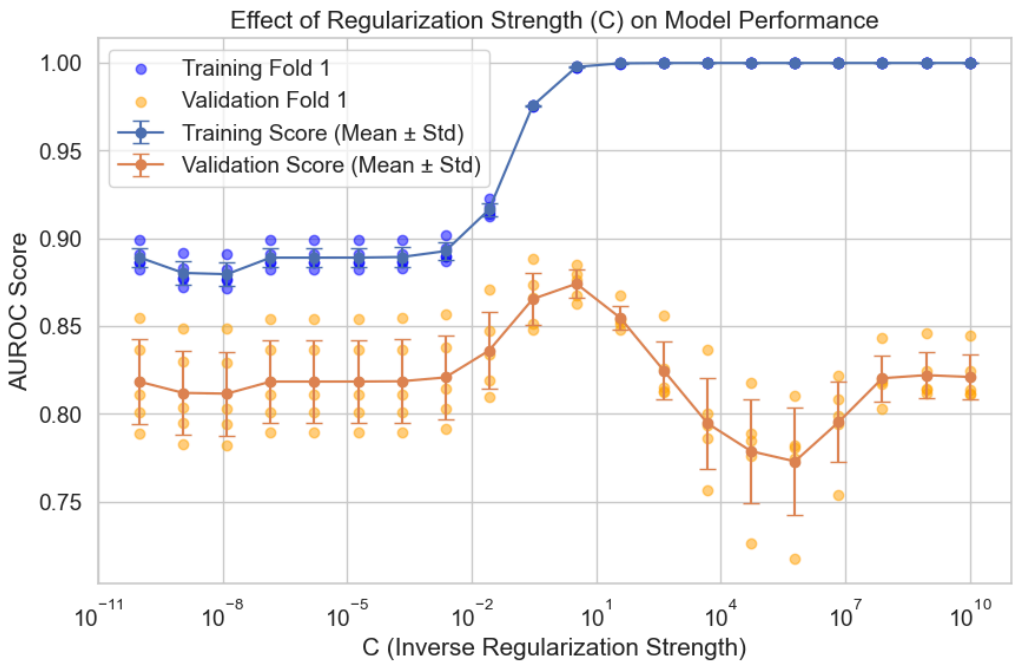

Figure 1.1: This plot shows the effect of regularization strength (C) on model performance, with AUROC score vs C (inverse regularization strength) on the training and held out validation data.

Our results, as depicted in Figure 1.1, showed that at higher values of C, the model tends to overfit the training data (the AUROC score approaches 1, ~0.997), and underfits at lower values of C. To choose the best C value, we wanted to maximize the validation AUROC (~0.89 AUROC) and minimize the error bar across the folds. This C value occurred at the peak of the validation curve, where **C was ~3.35 and the AUROC score was ~ 0.89**.

1D: Analysis of Predictions for the Best Classifier

	website_name	text
1907	yelp	for 40 bucks a head, i really expect better food.
1964	yelp	I also decided not to send it back because our waitress looked like she was on the verge of having a heart attack.
846	imdb	But this movie is not funny, considering the ridiculousness of it.
1743	yelp	I was disgusted because I was pretty sure that was human hair.
193	amazon	The loudspeaker option is great, the bumpers with the lights is very ... appealing.
1640	yelp	When my order arrived, one of the gyros was missing.
42	amazon	Unfortunately the ability to actually know you are receiving a call is a rather important feature and this phone is pitiful in that respect.
1706	yelp	Soggy and not good.
204	amazon	If you plan to use this in a car forget about it.
1692	yelp	say bye bye to your tip lady!

Figure 1.2: Examples of text categorized as ‘false positives’ from the heldout validation data.

From the examples in Figure 1.2 that show false positives (text that was incorrectly classified as a positive sentiment), our biggest observation was that the model did not pick up on context clues that the writer may have intended to convey. For instance, the use of the ellipses in row_id 193 for Amazon,

the use of the ellipses to a human would indicate sarcasm (the customer *did not* find the lights on the bumper appealing), but because of the presence of a generally 'positive' sentiment word - 'appealing' - this text was classified as positive. Either use of negation words, such as 'not', seemed to be overlooked *or* certain words, such as 'soggy', were not a part of the vocabulary and therefore were not assigned a sentiment in the heldout dataset.

	website_name	text
2355	yelp	The sides are delish - mixed mushrooms, yukon gold puree, white corn - beateous.
2378	yelp	Lordy, the Khao Soi is a dish that is not to be missed for curry lovers!
1450	imdb	This is a witty and delightful adaptation of the Dr Seuss book, brilliantly animated by UPA's finest and thoroughly deserving of its Academy Award.
1539	imdb	You wont regret it!
2374	yelp	The goat taco didn't skimp on the meat and wow what FLAVOR!
1563	imdb	I don't think you will be disappointed.
1425	imdb	The characters are fleshed out surprisingly well, particularly Grimes and Blake, and all the actors deliver their sharply scripted lines with just the right amount of deadpan tongue in cheek to make the dialogue both hilarious and realistic.
492	amazon	Their Research and Development division obviously knows what they're doing.
666	amazon	I did not have any problem with this item and would order it again if needed.
1452	imdb	He's a national treasure.

Figure 1.3: Examples of text categorized as 'false negatives' from the heldout validation data.

From the examples in Figure 1.3 that show false negatives (text that was incorrectly classified as a negative sentiment), it appears that both the negation words ('not', 'didn't', 'don't', etc.) and limited training vocabulary played a role in classifying these sentences as negative. Once again, the BoW representation cannot pick up on the context and tone of the customer - the model is only trained on the filtered vocabulary from the training dataset. The use of context clues and analysis of more than 1 word for assigning sentiment should be considered for Problem 2.

1E: Report Performance on Test Set via Leaderboard

Our AUROC for the test set per the **leaderboard score was 0.863,** which is lower than our **training (~0.997 AUROC)** and comparable to our **heldout (~0.89 AUROC)** AUROC scores for our best C value. This suggests that the model's performance is sensitive to the diversity and size of the training dataset. If the heldout and test sets contained vocabulary that the model was not trained on, then those words were ignored, potentially impacting its ability to generalize. The lower performance score on the test set is possibly due to overfitting the training dataset - the results are nearly perfect for the training dataset, but does not generalize well for the unseen data in the validation and test sets. Though noise was removed in the CountVectorizer preprocessing step through the elimination of stop words and filtering high-frequency terms (max_df), further refinement perhaps further filtering needs to be done or a wider range of words needs to be included in the training set. Another limitation of BoW is that it utilizes only unigrams for analysis, thereby missing context in the text from word combinations.

Problem 2

2A: Feature Representation Description

We used Bidirectional Encoder Representations from Transformers (BERT) embeddings for both the training and test sets. First, the sentences in the training and testing set are translated within a tokenizer. Here, the sentences are split into words and subwords called tokens, the tokens are mapped

to specific IDs in BERT's vocabulary, and special tokens are added. This preprocessing transforms sentences into a format compatible for BERT's processes. The BERT model then takes these numerical tokens and generates contextual embeddings for each token. This captures the meaning of and relationships between words within a sentence. We used the provided BERT embeddings for the training and testing set, which included fixed-length feature vectors of 768 dimensions that were suitable for classification tasks. This was a more sophisticated approach as compared to problem 1 as it considers word order and context.

2B: Cross-Validation (or Equivalent) description

Our cross-validation process was nearly identical to that described in 1B, but with a different hyperparameter. We again used GridSearchCV to perform a 5-fold cross validation (CV). The same steps were carried out, using train_test_split from SciKitLearn to split our dataset and hold out 20% of the data for the validation process. GridSearchCV again broke the training dataset into 5 equal folds, where 4 of the 5 folds were used for training and 1 fold was used for validation. This process was repeated 5 times, with the holdout validation set changing on each iteration. Model performance was again evaluated by computing average of the Area Under the Receiver Operator Curve (AUROC) across all folds. GridSearchCV allowed us to specify our hyperparameters, which in this case was across a range of our hyperparameter alpha. This alpha represented the strength of our L2 regularization term to penalize overfitting. CV helped us identify the best alpha value as averaged over the validation sets, and we used this hyperparameter to make predictions on our test set.

2C: Classifier Description with Hyperparameter Search

The classifier we used was the **Multi-Layer Perceptron (MLP),** an artificial neural network, with an L2 penalty. This is a more complex classifier as compared to our logistic regression classifier from problem 1, and can do well with recognizing complex patterns such as those in the BERT embeddings. The L2 penalty also prevents overfitting to our training test set.

During training, **no convergence issues** were found as our max iterations in the MLP classifier was set to 5,000. We initially used the number of iterations as a hyperparameter, but no significant difference in the AUROC score was found once the iterations were high enough to converge. Consequently, we switched our hyperparameter to the alpha term. To avoid overfitting to the training data, the 5-fold CV ensured performance on multiple splits of the dataset.

To tune the **L2 penalty**, we tested a grid of **10 log-spaced values ranging from 10e-2 to 10e2**. A higher alpha term corresponds with a stronger penalization and higher regularization while a lower alpha term corresponds with a weaker penalization.

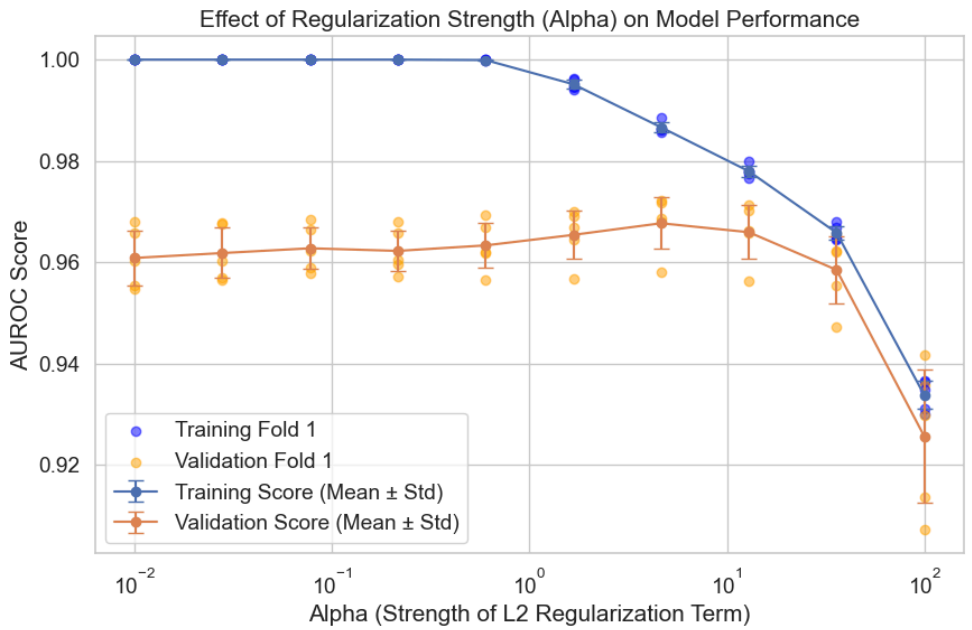

Figure 2.1: This plot shows the effect of regularization strength (alpha) on model performance, with AUROC score vs alpha (L2 Regularization term) on the training and held out validation data.

Our results are shown in Figure 2.1, where we can see an **alpha term of 4.642 is preferred**. The model tends to overfit to the training data at lower alpha terms, where L2 regularization is not as strong. When alpha is 4.642, the AUROC score for the validation set is maximized. After this point, the regularization term's strength causes underfitting. A clear peak is shown at our chosen alpha score, and our evidence is decisive.

2D: Error Analysis

	Index	True Label	Predicted Label	Text
0	1	0	0.712598	Im big fan of RPG games too, but this movie, its a disgrace to any self-respecting RPGer there is.
1	3	0	0.808811	A Lassie movie which should have been "put to sleep".... FOREVER.
2	34	0	0.997725	The loudspeaker option is great, the bumpers with the lights is very ... appealing.
3	38	0	0.603879	I ordered this for sony Ericsson W810i but I think it only worked once (thats when I first used it).
4	44	0	0.690419	I'll be looking for a new earpiece.
5	84	0	0.771420	say bye bye to your tip lady!
6	89	0	0.610997	Not only did it only confirm that the film would be unfunny and generic, but it also managed to give away the ENTIRE movie; and I'm not exaggerating - every moment, every plot point, every joke is told in the trailer.
7	106	0	0.902561	You can find better movies at youtube.
8	107	0	0.511139	I can't believe there's even a sequel to this!
9	152	0	0.877904	And, FINALLY, after all that, we get to an ending that would've been great had it been handled by competent people and not Jerry Falwell.

Figure 2.2: Examples of text categorized as 'false positives' from the heldout validation data.

Based on the false positives that were identified, reviews that contained positive words and phrases were falsely identified as positive. Most reviews contain phrases that typically denote a good experience, such as "big fan", "great", "appealing", "can't believe", "better", and "tip". It also seems that many of the predictions were around 0.6, meaning that the predicted probability of the text being categorized as positive was slightly higher than negative. One example, "The loudspeaker option is great, the bumpers with the lights is very … appealing", relies on sarcasm, which is hard for a model to

detect even with a contextual understanding of the words' relations to each other. Overall, it seems the classifier has a better understanding of words' contexts as compared to the Logistic Regression classifier, but struggles with more ambiguous reviews and reviews that rely on a more nuanced or cultural understanding.

	Index	True Label	Predicted Label	Text
0	2	1	0.299378	I've dropped my phone more times than I can say, even on concrete and my phone is still great (knock on wood!).
1	9	1	0.449888	The sides are delish - mixed mushrooms, yukon gold puree, white corn - beateous.
2	48	1	0.276797	You wont regret it!
3	58	1	0.439076	I have used several phone in two years, but this one is the best.
4	88	1	0.387448	Their Research and Development division obviously knows what they're doing.
5	96	1	0.456188	I did not have any problem with this item and would order it again if needed.
6	144	1	0.030625	Because both ears are occupied, background is not distracting at all.
7	151	1	0.354049	Gets a signal when other Verizon phones won't.
8	240	1	0.493004	" In fact, it's hard to remember that the part of Ray Charles is being acted, and not played by the man himself.
9	249	1	0.277869	I'm still infatuated with this phone.

Figure 2.3: Examples of text categorized as 'false negatives' from the heldout validation data.

From the examples of false negatives in Figure 2.3, similar themes emerge. Most of the reviews are gathered around 0.4 in the predicted probability of being positive. Words and phrases that tend to be classified as negative are also present in the sentences, including "dropped my phone more times than I can say", "used several phones in two years", "occupied", and more. It seems the model struggles with understanding a conclusion of positive sentiment when most of the sentence implies a negative one. It also misclassified "You wont regret it", which also shows that the contextual understanding of the words is not fully developed. In comparison with the Logistic Regression classifier from problem 1, a lot less false negatives slip through, and the model still demonstrates a better understanding of the context of words.

2E: Report Performance on Test Set via Leaderboard

Our AUROC for the test set per the leaderboard score was 0.96388. This was lower than our training AUROC score of 0.9833, but comparable to our validation AUROC score of 0.9676. As our test set score and heldout data scores are similar, it indicates our model is not overfit and can perform well on unseen data. This is a significant improvement as compared to our performance on problem 1, where we received an AUROC score of 0.863. This improvement is likely due to the BERT embeddings, which is a richer and more informative feature vector representation. It's ability to better understand the meaning of words and phrases improved our model. This more complex representation of our sentences, coupled with a classifier (MLP) that can recognize more intricate patterns as compared to Logistic Regression likely resulted in better performance.