Task 3

I developed three language models to predict the genre of the song: unigram, bigram and mixed model, which combines the unigram and bigram models. I used the test set to evaluate all of these models on different performance metrics, more specifically the precision, recall, and f1 score which combines the precision and recall. The results if the evaluation can be seen in Table 1.

Model	Precision	Recall	F1 Score
Unigram	0.2292	0.2222	0.1742
Bigram	0.3393	0.3889	0.3003
Mixed	0.3393	0.3889	0.3003

Table 1: Evaluation results of different models.

As we can observe from Table 1, precision, recall, and f1 scores for the bigram model and the mixed model are exactly the same. This is due to the fact that my mixed model works the best with lambda value of 0.0 (meaning that it uses only the bigram model for the prediciton). I used a 90:10 training/validation split to find out the best lambda value. I tried experimenting with relatively small labmda values but none of them produced better results. My mixed model is thus not really mixed since it uses only the bigram. I believe that if I had more training and testing data, the mixed model would use both unigram and bigram models.

Task 4

I decided to use the McNemar's Test (sometimes also called "within-subjects chi-squared test") to compare the predictive accuracy of two models. McNemar's Test is a statistical test for paired nominal data. It is based on a 2 times 2 contingency table of the two model's predictions. I formulate the null hypothesis that the probabilities p(b) and p(c) are the same. Thus, the alternative hypothesis is that the performances of the two models are not equal.

	Model 2 correct	Model 2 wrong
Model 1 correct	a	b
Model 1 wrong	c	d

Table 2: Example contingency table for McNemar's Test.

The McNemar test statistic ("chi-squared") can be computed using Equation 1. The more commonly used variant which corrected the continuity is described in Equation 2.

$$\chi^2 = \frac{(b-c)^2}{(b+c)} \quad (1) \qquad \chi^2 = \frac{(|b-c|-1)^2}{(b+c)} \quad (2)$$

I am going to set the significance threshold to be $\alpha = 0.05$. The exact p-value for the McNemar's Test can be computed as follows:

$$p = 2\sum_{i=b}^{n} \binom{n}{i} 0.5^i (1-0.5)^{n-i} \quad (3)$$

where $n = b + c$, and the factor 2 is used to compute the two-sided p-value.

Because we have three different models, we should compute the McNemar's Test to all 3 pairs of models, but since my mixed model only uses the bigram model, I am going to exclude the mixed model from the significant testing. We can assume that the mixed model and the bigram model perform the same. Thus I am going to conclude significant testing between the unigram and bigram models. I used the predicted labels form both models and the true labels to create the contingency table which can be viewed in Table 3.

	Bigram correct	Bigram wrong
Unigram correct	3	1
Unigram wrong	3	8

Table 3: Contingency table of unigram and bigram predictions.

Based on this contingency table I was able to compute the chi-squared value and p-value. The chi-squared was 0.25 and the p-value was 0.617. Since my p-value is larger than my selected significance threshold $\alpha = 0.05$ I cannot reject my null hypothesis and assume that there is no significant difference between the two models. I have to accept my alternative hypothesis which states that the performances of the two models are not equal.

Task 5

Since my mixed model is the same as my bigram model I am going to discuss only cases for unigram and bigram models.

Unigram model

My unigram model was able to correctly predict 4 out of 15 test cases. As an example, the model correctly predicted the text of test case 11 ("Tell the leaves not to turn But don't ever tell me I'll learn...") as Pop. However, it did not classify the text of test case 14 ("This ain't a song for the broken-hearted No silent prayer for the faith-departed...") correctly as Rock. Instead, it classified it as Pop. The unigram model predicted a lot of the songs as Pop. It classified 3 true positives and 5 false positives for the Pop genre. My initial reasoning for this behavior was that pop had a lot more training data than other genres but after careful examination, I discovered that this is not the case because Rap had the most training data.

Bigram model

My bigram model was able to correctly predict 6 out of 15 test cases. As an example, the model correctly predicted the text of test case 14 ("This ain't a song for the broken-hearted No silent prayer for the faith-departed...") as Rock which was previously misclassified by the unigram model. However, it did not classify the text of test case 15 ("They call it stormy Monday But Tuesday's just as bad...") correctly as Blues. Instead, it classified it as Rock. The bigram model did not have the same behavior as the unigram model where it had a preference for one genre. However, it did not make a single prediction as Metal. I tried examining this behavior but was unable to find reasoning.

Overall project analysis

I followed the directions of the class activity when developing the unigram and bigram models and used soothing when computing the probability. I also decided to add 1 to the result of the probability to ensure that my probabilities are larger than 1. I did this due to the use of a logarithm when computing the final probability because the logarithm function returns positive values only for values greater than 1.

I also found an error in the test cases where the lines were concatenated without a separating space. I created a script that fixes this issue. I also decided to combine all of the files from one genre together so I could achieve a more even training/validation split. I am splitting the data based on the total number of lines in all songs instead of the number of songs. This approach resulted in better results when computing the ideal lambda value.

I tried using multiple different types of tokenization (word tokenization, WordPiece, stop word removal) and I found that using only word tokenization without stop word removal achieves the best results.