**CS 6200 - Natural Language Processing**

**Anurathi Bala**

**Khushi Neema**

**Text Classification Using N-Grams**

**Introduction:**

The aim of this project is to classify medical transcription data into different medical specialties using various machine learning models. Medical transcription data is used to convert voice-recorded medical reports into a written format. These reports are used for various purposes, such as maintaining patient records, communicating with other medical professionals, and processing insurance claims. Medical transcription data is hard to find because it contains sensitive and confidential information about patients, such as their medical history, diagnoses, and treatments. Therefore, access to this data is restricted and requires strict privacy regulations to ensure that patient confidentiality is maintained. Additionally, medical transcription requires specialized skills and knowledge of medical terminology, making it more expensive and time-consuming to obtain compared to other types of text data. The ability to accurately classify medical transcription data into different medical specialties can improve the efficiency and quality of healthcare services.

**Approach:**

**Pre-processing:**

The first step of our approach is to combine similar medical specialties together to increase the amount of data per class. This step is important because some medical specialties may have too few samples to be accurately classified. After combining similar medical specialties, we clean the transcription data by tokenizing, removing punctuation, stopwords, and lemmatizing the words. This step helps to reduce the complexity of the data and remove any irrelevant information.

**Visualization:**

Next, we visualize the top words in every category to get a sense of what the important features are in each medical specialty. We then correct the class imbalance by performing upsampling or downsampling based on the word count of each class. Class imbalance is a common issue in many machine learning applications, including medical transcription data. In a classification task, class imbalance occurs when the number of instances belonging to one class is significantly higher or lower than the number of instances belonging to other classes. It can significantly impact the accuracy and performance of machine learning models, as the model can become biased towards the majority class and perform poorly on minority classes. In medical transcription data, the number of instances belonging to one medical specialty or diagnosis could be much greater or much smaller than others and can result in misclassification of important medical information, leading to incorrect diagnoses or treatments.

**Exploratory Data Analysis:**

To further understand the data, we use t-SNE visualization to cluster the high-dimensional data into 2D space. t-SNE (t-distributed stochastic neighbor embedding) is a powerful tool for visualizing high-dimensional data in a lower-dimensional space, typically in two or three dimensions. It is often used to visualize complex patterns in data and is especially useful when dealing with large datasets with many variables. In the case of medical transcription data, t-SNE can be used to visualize the similarities and differences between transcriptions in each category. By projecting the high-dimensional data into a 2D space, t-SNE can reveal patterns and clusters of related transcriptions that may be difficult to see in the original high-dimensional space. By using t-SNE to map these vectors into two-dimensional space, we can obtain a scatter plot of the transcriptions, where each point corresponds to a single transcription. Points that are close together in the plot correspond to transcriptions that are similar to each other, and points that are far apart correspond to transcriptions that are dissimilar. This visualization can be used to ensure that the categories we have defined actually group similar transcriptions together. If there are outliers or points that are far away from the main clusters, we may need to revisit our preprocessing or category definitions. Additionally, t-SNE can be used to identify areas where the model is struggling to classify the data correctly, which can help guide future improvements to the model. From Figure 1, we can see points having the same color are closer together. This representations tells us that the model will not face difficulty in classifying the data accurately.

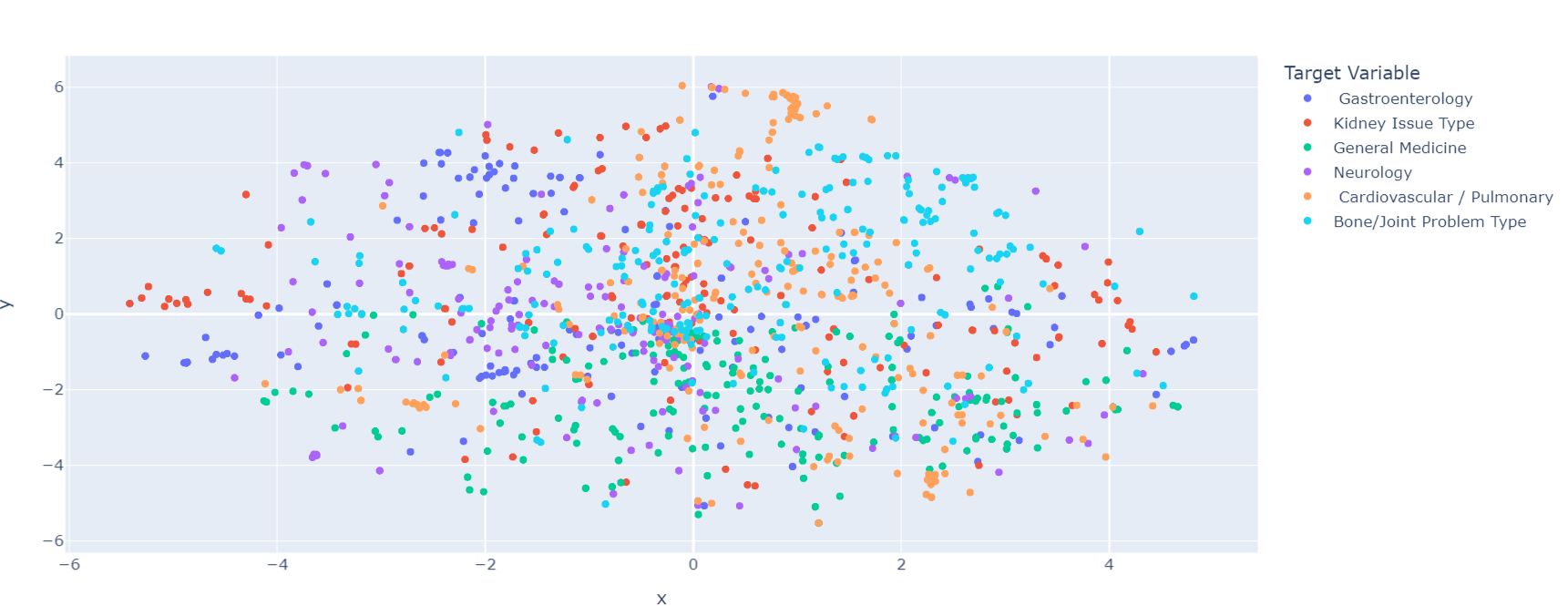


Figure 1 : Visualization of high dimensional data in a 2-D Space using t-SNE method

**Data Preparation:**

We then split the data into train and test sets and use N-gram split (unigram, bigram, trigram) to convert the text into vector form. N-grams can be used on medical transcription data for feature extraction, which is a crucial step in building machine learning models for text classification. In the context of medical transcription data, N-grams refer to contiguous sequences of N words. A unigram is a single word, a bigram is a sequence of two words, and a trigram is a sequence of three words. To use N-grams in machine learning models for medical transcription data, the first step is to tokenize the text into individual words. Once the text is tokenized, N-grams can be extracted by sliding a window of size N over the text. Each window will contain a sequence of N words, which is the N-gram. These N-grams can then be used as features in a machine learning model. By including N-grams as features in the model, the model can better understand the relationship between words and therefore improve its accuracy in classifying medical transcription data. For example, the bigram "heart attack" is more informative than the unigrams "heart" and "attack" separately, as they might occur frequently in the text independently of each other.

**Model Selection:**

The Multinomial Naive Bayes, K-Nearest Neighbors, and Decision Tree models are used on medical transcription data split into n-grams. The input to the model would now be the n-gram features instead of the raw text. It is worth noting that the choice of n-gram size and the number of n-grams used can significantly impact the performance of these models. A smaller n-gram size may capture more context, but may also result in higher dimensionality and sparsity, while a larger n-gram size may result in overfitting. The number of n-grams used can also be limited to avoid high dimensionality and reduce noise. These parameters would need to be tuned during experimentation to optimize performance.

**Multinomial Naive Bayes** (MNB) is a probabilistic model that is commonly used for text classification. It is based on Bayes' theorem, which calculates the probability of a class given the input features. In the case of text classification, the input features are the words or n-grams in the text. MNB works well for high-dimensional data, such as text data, and can handle sparse data.

**K-Nearest Neighbor** (KNN) is a non-parametric model that is based on similarity. It classifies an instance by finding the k-nearest neighbors to the instance in the training set and assigning the majority class of these neighbors to the instance. In the case of text classification, the similarity is based on the distance between the vector representations of the text. KNN can be computationally expensive for high-dimensional data but is relatively easy to interpret.

**Decision Tree** is a model that is based on a tree-like structure where each node represents a test on an input feature and each branch represents the outcome of the test. The leaves of the tree represent the class labels. Decision trees are easy to interpret and can handle non-linear relationships between the input features and the class labels. In this problem, each of the nodes are words (n-gram) and the branches are the category to which each n-gram belongs to.

The LSTM model is used on the cleaned medical transcription data and is compared with the other models. **Long Short-Term Memory** (LSTM) is a type of recurrent neural network (RNN) that is commonly used for sequential data, such as text data. LSTM can handle variable-length input sequences and can capture long-term dependencies between the words in the text. LSTM has been shown to work well for text classification tasks, including medical text classification.

The model architecture consists of an embedding layer that takes the input shape of the data and converts it to a vector of 128 dimensions. The input length of the data is set to the maximum length of the input sequence. This is followed by a Bidirectional LSTM layer that takes the 128-dimensional input vector and processes it through a set of LSTM cells that process the input sequence in both directions. This allows the model to capture dependencies and relationships between the data across time steps. The output of the LSTM layer is then fed into two dense layers with 32 and 64 neurons respectively, which apply a rectified linear unit (ReLU) activation function to the output of the previous layer. These dense layers act as a feature extractor, reducing the dimensionality of the input data and mapping it to a lower-dimensional space. Dropout regularization is applied after each dense layer to prevent overfitting. Dropout randomly drops out a fraction of the neurons during training, forcing the remaining neurons to learn more robust features. Finally, a dense layer with 6 neurons and a softmax activation function is added to the model, which maps the output of the previous layer to a probability distribution over the six possible output classes. The categorical cross-entropy loss function is used as the loss function, with the Adam optimizer used to optimize the weights of the model during training.

Overall, these models can be trained on medical transcription data that has been preprocessed and represented using n-grams. The models can then be evaluated on various metrics such as accuracy, precision, recall, and F1 score to determine their performance. The choice of which model to use may depend on the specific characteristics of the data and the desired trade-offs between interpretability and accuracy.

**Assessment Methodology:**

We perform cross-validation to evaluate the performance of our models and avoid overfitting. Cross-validation is a technique used to evaluate the performance of machine learning models by splitting the data into multiple subsets, or folds, and using each fold as a test set while training the model on the remaining data. Cross-validation helps to address overfitting by providing a more reliable estimate of how well the model will perform on new data. By evaluating the model on multiple subsets of the data, we can get a better estimate of its true performance, and if the model is overfitting, it will perform poorly on the test sets.

The evaluation metrics we use include accuracy, precision, recall, and F1 score. Accuracy measures the percentage of correctly classified instances over the total number of instances. It indicates the overall performance of the model in correctly identifying the classes. Precision measures the percentage of correctly classified instances of a particular class over the total number of instances predicted as that class. It indicates the ability of the model to correctly classify instances of a particular class. Recall measures the percentage of correctly classified instances of a particular class over the total number of instances belonging to that class. It indicates the ability of the model to correctly identify instances belonging to a particular class. F1 score is the harmonic mean of precision and recall, and it provides a balanced measure of both precision and recall. It indicates the overall performance of the model in correctly identifying instances across all classes. In the context of medical transcription data classification, these metrics provide valuable insights into the performance of the models in accurately classifying the medical transcription data into different medical specialties.

**Results:**

**Conclusion:**

The Naive Bayes model with bigram features has the highest accuracy of 0.879167, followed by the Naive Bayes model with unigram features at 0.870833, the LSTM with 0.868027, and the Naive Bayes model with trigram features at 0.864583. This suggests that the Naive Bayes algorithm is well-suited to this task of medical transcription classification using n-grams as features.

The LSTM model has an accuracy of 0.868027, which is comparable to the Naive Bayes models. LSTMs are a type of recurrent neural network that can capture sequential information from the input data, making them suitable for natural language processing tasks.

The KNN and Decision Tree models have lower accuracies compared to the Naive Bayes and LSTM models. This suggests that these models may not be the best fit for this particular task or dataset. The KNN model with bigram features has the lowest accuracy at 0.435417, indicating that it is the least suitable model for this task. KNN is a distance-based algorithm that calculates the distance between the data points to classify them. In the case of unigrams, each word is treated as a separate feature, and the distance between two documents is calculated based on the number of common unigrams. Therefore, it works well with unigrams because the distance between two documents can be calculated based on a single feature. However, when it comes to bigrams and trigrams, the number of features increases significantly, and the distance between two documents cannot be calculated based on a single feature. Instead, it is a combination of multiple features that determine the distance between two documents. This high-dimensional feature space causes the KNN algorithm to struggle in finding the nearest neighbors, and it may not be able to find the best match between two documents. This is why KNN with bigrams and trigrams does not work as well for classifying medical transcription data as it does with unigrams.

For each model, we evaluated the performance using unigrams, bigrams, and trigrams on each medical specialty. We selected and compared the best model performances that achieved the maximum accuracy for a specific n-gram input. From Figure 2, we can see that Naive Bayes using Bigrams has proved to be the most accurate in almost every category. Naive Bayes - Bigram is the optimal model for this particular problem. The LSTM model is a close second for an optimal model, however, its interpretability is a bit more difficult and the implementation is not as quick. Hence, the Naive Bayes model is our top pick.

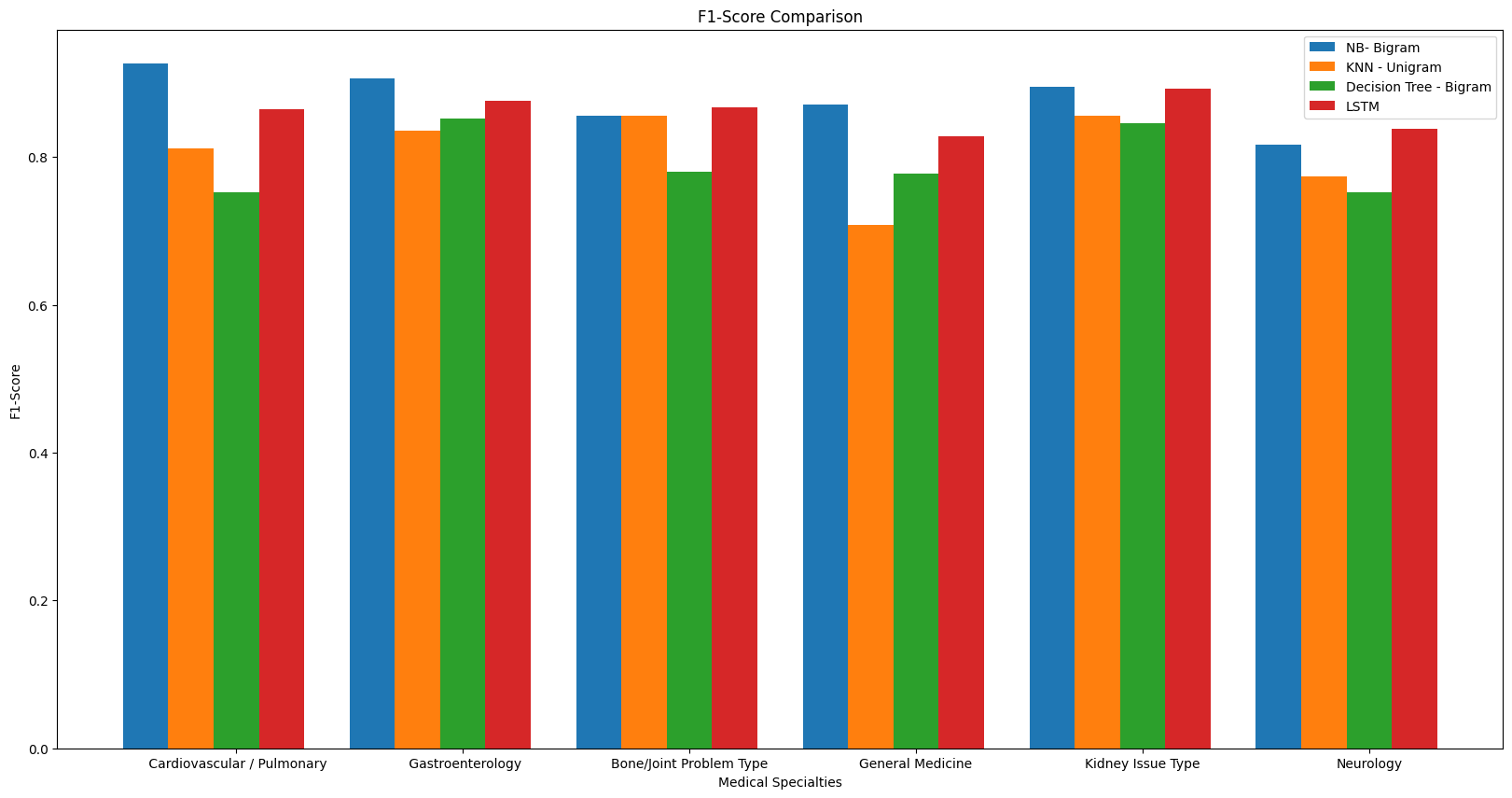


Figure 2 : Analysis of F-1 Scores in each category for the best selected models

In conclusion, our approach of combining similar medical specialties, cleaning the data, and using N-gram split and various ML models like Naive Bayes, KNN, Decision Tree, and LSTM for classification of medical transcription data is effective and accurate. Our results demonstrate the potential for using machine learning models to classify medical transcription data into different medical specialties, which can improve the efficiency and quality of healthcare services.

**Future Experiments:**

Future experiments could include using other machine learning models or feature extraction techniques to improve the performance of the classification models. Additionally, more data could be collected to increase the accuracy of the models and reduce class imbalance. Finally, the models could be tested on data from different healthcare systems to evaluate their generalizability.

**Reference:**

[1] "A Survey on Text Classification: From Traditional to Deep Learning Techniques" by S. S. Bhatia et al.

[2] B. J. Marafino, J. M. Davies, N. S. Bardach, M. L. Dean, and R. A. Dudley, “N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit” by B. J. Marafino, J. M. Davies, N. S. Bardach, M. L. Dean, and R. A. Dudley.

[3] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[4] H. Wang, Z. Gao, Y. Huang, H. Yu, X. He, and H. Zhang, "Bidirectional LSTM-CRF models for sequence tagging," arXiv preprint arXiv:1508.01991, 2015.