Classification of Lung Cancer Patients by Intensity Level

Karthik Raj Nitesh Sunku Mary Nguyen

4/23/2021

ECE 196, Spring 2021 - University of California, San Diego

Abstract - **Lung Cancer is often considered one of the most dangerous cancers, being the most common and claiming the most lives annually, with millions of victims. We aim to produce a Machine Learning Classification model that can effectively categorize lung cancer patients based on their severity, which can guide treatment plans and resource allocation, allowing for more hospital efficiency and more lives saved. The model is a KNN classification that takes in normalized data points that have undergone PCA and currently outputs the accuracy per training iteration, which with a 25% training dataset hits 97-99% on average. These promising results allow us to conclude that our model can accurately predict the severity of a patients' lung cancer and serve as an effective prognostic indicator for treatment plans.**

I. Introduction

Lung cancer is the most common cancer in the United States and globally, having more cases than the combined number of breast, colon, and prostate cancer patients, and having the greatest death count annually of any cancer, being responsible for one out of every five cancer deaths [1]. On average, roughly 1.4 million people die from lung cancer worldwide, and it is also considered the seventh leading cause of death in the world [2]. The majority of lung cancer cases are attributed to tobacco smoking, with particulate matter (PM), air pollution, hazardous working conditions, second-hand smoking, and biological factors (genetics), also contributing [3].

With the annually increasing rate of lung cancer due to continued tobacco use and increasing PM in the atmosphere, especially in industrialized areas, alongside an evermore overworked health care force, hospitals and physicians would benefit from diagnostic predictions of lung cancer based on the patient's lifestyle and common risk factors [3]. This could help guide the course of action for the patient by prescribing appropriate treatments and lifestyle changes based on the severity level and allocate resources more efficiently by targeting it to those in greater need more quickly. This is particularly important as the National Health Insurance system (NHI) records state that although critically ill lung cancer patients only account for ~4% of the medical population, they consume almost 30% of all medical expenditures [1]. This means that many resources are allocated to lung cancer patients, which could be debilitating for the hospital's other sectors, especially during the COVID-19 Pandemic, but it also placed an enormous cost on the patient. By having these early prognosticators, patients could save thousands of dollars and be prescribed treatment plans at much earlier stages, which allows physicians to enact preventative care and save more lives as a result.

An important aspect of lung cancer treatment, however, is that as there are two types of lung cancer, and they each have multiple stages in each, lung cancer can be visualized as a spectrum, which can make it difficult to efficiently and properly categorize. Therefore, to categorize lung cancer severity, rather than organizing by each stage, the model will organize the stages into a more general "low, medium, and high" severity groups, in which medium and high can have more attention for their more serious needs.

The goal is to develop a classifier that can accurately categorize lung cancer patients to their appropriate lung cancer severity level based on their lifestyle and common risk factors. We used a dataset containing records of 1000 lung cancer patients with their corresponding cancer severity and exposure to common risk factors. In order for any machine learning algorithm to function properly, the data needs to be cleaned (removal of missing values or potential replacement), qualitative information should be converted into quantitative information, and any columns irrelevant to the machine learning algorithm would be removed. We also wanted to check if there exists a correlation between their corresponding severity levels and provided lifestyle characteristics. Thus, we ran PCA, a linear algebra dimensionality reduction approach that reduces the original dataset into its principal components while retaining most of the variance of the original dataset. Upon plotting the two core components that capture the highest proportion of the variance of the original dataset and annotating the patient entries with their corresponding level, we hope to see three distinct clusters: one for low severity lung cancer patients, one for medium severity lung cancer patients, and another for high severity lung cancer patients. If there is promising clustering of these patients, then a classification model can be applied to our dataset's top principal components. Because our dataset does not have any intrinsic characteristics that lends itself to a specific classification model, we decided to also explore KNN, logistic regression, and SVM and choose the one that consistently results in the highest accuracy metrics.

II. Solutions

Initially, we downloaded the dataset and displayed it as a dataframe via the PANDAS module. A few checks were conducted: percent of missing values per column, number of qualitative variables, any clear unnecessary columns. We found there were zero missing values, so there was no need to replace certain values or remove columns predominantly containing missing values. The "patient_id" column was removed because a patient's id has no correlation with their severity of lung cancer. There were two qualitative columns: gender and level. Since the gender contained only two unique values: male and female, one-hot encoding was conducted on this column with '1' denoting a male and '0' for a female. The level column had three unique values: low, medium, and high. Thus, '0' was used for low, '1' for medium, and '2' for high. Now, we are one step closer to using PCA.

PCA

PCA was used to verify whether the data can be accurately split into the three distinct levels: low, medium, and high. If successful, the PCA generated dataset lends itself to shorter training and testing times due to the reduced dimensionality of the dataset. To use PCA, the data needs to be standardized where the columns or features all have zero mean and have a variance of 1 as well. Intrinsically, we are treating each of the columns as gaussian distributions [4]. Then, within PCA, the covariance matrix (a square matrix that identifies how features are correlated with each other) can be decomposed into a summation of eigenvalues multiplied with their corresponding eigenvectors [4]. PCA strives to reduce dimensionality by initially identifying whether the original dataset can be restructured in such a way that less dimensions are needed to explain the variance or the variability of the original dataset. Each eigenvalue explains how much variance of the initial dataset is explained by the corresponding eigenvector or axis of direction [4]. Large eigenvalues indicate large separation of the data when projected on the corresponding eigenvector, while really small eigenvalues close to 0 reflect low separation of the data when projected on its corresponding eigenvector [4]. Also, larger eigenvalues indicate the corresponding eigenvectors explain more of the variance of the initial dataset. The user can take the absolute value of the eigenvalues and sort them in descending order. Depending on the threshold the user sets for their ideal explained variance, they can choose the corresponding number of components that satisfies this cutoff. Each of the principal components is created by projecting the original normalized data onto the eigenvectors, by taking the inner product of the original normalized dataset and the eigenvector [4]. Finally, the top principal components, which can encompass the highest explained variance of the original dataset, can be plotted. Because of the intrinsic characteristic of the top principal components, we should see great separation of the data.

As mentioned above, one of the key metrics of PCA's success is how well each component encapsulates the variance of the entire dataset. Unfortunately, our top two components only retained 50% of the variance of the original dataset when reduced to lower dimensions. However, another key metric for PCA is to plot the top principal components and annotate for each data point, their corresponding level. Ideally, for this specific dataset, we should be seeing three distinct clusters: one for patients with low lung cancer severity, one for patients with medium lung cancer severity, and one for high lung cancer severity.

When we plotted the top two components, we observed clustering of the low and high lung cancer severity level patients, but the medium lung cancer severity level patients were sporadic.

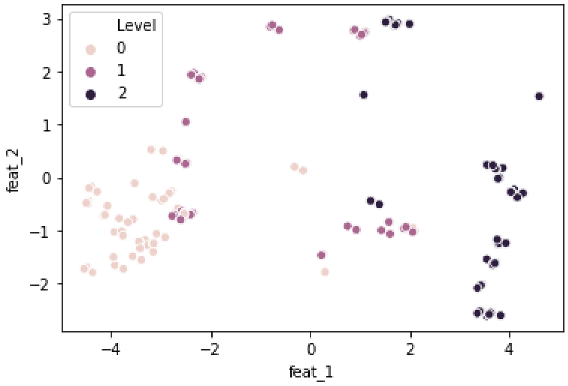


Figure 1: 2D Plot of Top Two Principal Components

We were curious to see whether the low and high level lung cancer patients separated well, so the following plot contains the same data, but with removal of the medium level lung cancer patients.

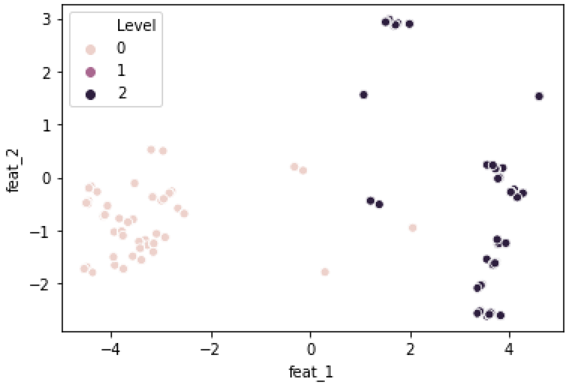


Figure 2: 2D Plot of Top Two Principal Components with "Medium" Severity Redacted

In conjunction, these plots illustrate that the PCA generated components do separate low and high level patients well, but have trouble isolating the medium level patients. One possibility is that there is indeed separation of all three levels, but this would only be observed, by using another principal component to create a 3-dimensional plot. When we graphed the top three principal components, we observed the best separation of the data.

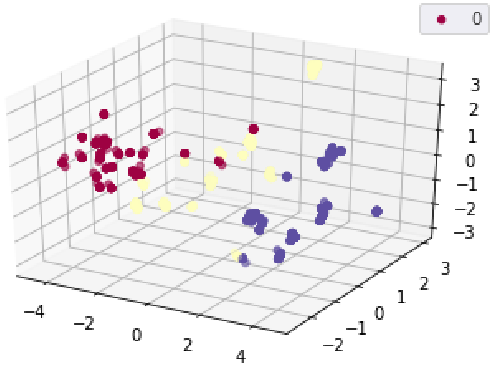
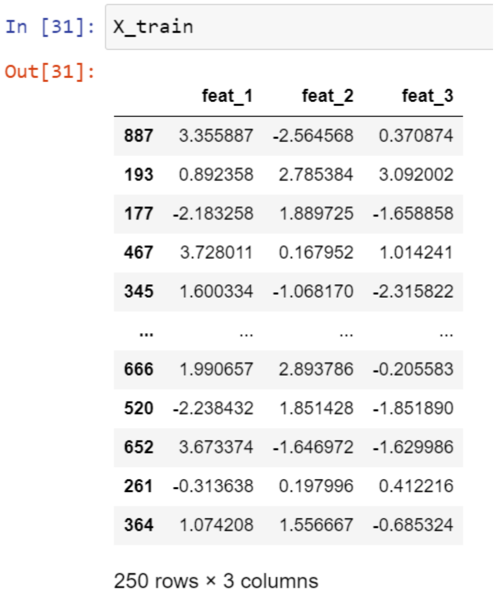


Figure 3: 3D Plot of Top Three Principal Components

This great clustering implies that the dataset did indeed have some core structure, which was successfully revealed via PCA. More importantly, this implies that this dataset can be successfully classified to any of the three lung cancer severity classes with only the top three principal components. After evaluating the accuracy metrics of logistic regression, support vector machines (SVM), and K-Nearest Neighbors (KNN), we decided to use KNN, which uses euclidean distance to classify data entries as a specific class.

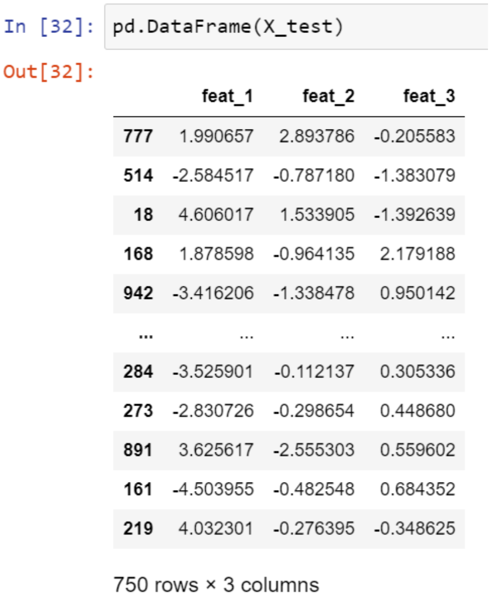
The training set holds a random 25% of the pca-transformed dataset. The testing dataset was also chosen randomly, containing the remaining 75% of the dataset. These values were deemed optimal after running through iterations of the training and testing process with a wide range of sizes for each of the training and testing sets.



```
In [31]: X_train
Out[31]:
```

	feat_1	feat_2	feat_3
887	3.355887	-2.564568	0.370874
193	0.892358	2.785384	3.092002
177	-2.183258	1.889725	-1.658858
467	3.728011	0.167952	1.014241
345	1.600334	-1.068170	-2.315822
...	...	...	...
666	1.990657	2.893786	-0.205583
520	-2.238432	1.851428	-1.851890
652	3.673374	-1.646972	-1.629986
261	-0.313638	0.197996	0.412216
364	1.074208	1.556667	-0.685324

250 rows × 3 columns



```
In [32]: pd.DataFrame(X_test)
Out[32]:
```

	feat_1	feat_2	feat_3
777	1.990657	2.893786	-0.205583
514	-2.584517	-0.787180	-1.383079
18	4.606017	1.533905	-1.392639
168	1.878598	-0.964135	2.179188
942	-3.416206	-1.338478	0.950142
...	...	...	...
284	-3.525901	-0.112137	0.305336
273	-2.830726	-0.298654	0.448680
891	3.625617	-2.555303	0.559602
161	-4.503955	-0.482548	0.684352
219	4.032301	-0.276395	-0.348625

750 rows × 3 columns

Figure 4: The data set split into 75% for testing and 25% for training.

KNN

KNN works by comparing the surrounding points to the one it is attempting to classify and assumes the point's class will be the same as the majority class among its neighbors. Because of this meticulous process, KNN takes a lot of time to train and test. However, it was not seen to be an issue with our dataset, which only contains 1000 entries. Furthermore, PCA's dimensional reduction helped reduce the potential computational time, allowing for quick classification.

To determine the optimal k-value for KNN, we computed the percent error ($\#\ of\ incorrect\ predictions\ /\ \#\ of\ predictions$)for all potential values ranging from 1 to 20. We ran this experiment multiple times to see if there is a consensus optimal number of neighbors observed from all the iterations, and the results are visualized in Figure 5 below.

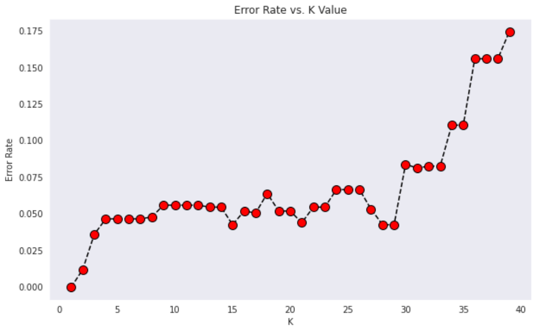


Figure 5: Graph of Error Rate vs K-Value for KNN

During the process of finding the optimal k-value and seeing the observation of increasing k-value yields an increasing error value, the error rate was graphed to verify the optimal k-value. Figure 5 reflects that trend line relationship. This instance like the others reflected that the optimal k-value is 1 and it had negligible error on average.

We observed that with increasing the k-value (parameter defining a field for reference points), there was an associated increase in error as seen by Figure 5 [5]. A potential explanation lies within the 3D PCA plot. Within the PCA 3D plot, we observe three dispersed dense clusters. As the k-value is increased, the number of points that can influence the final classification of a point increases accordingly. Thus, with very large k-values, it relies on other clusters for classification, making it likely that the classification of the point will be incorrect. Therefore, due to the observed clusters, it is rational for the k-value to be 1, despite the inherent risk that the nearest point of interest could be different from the point it is classifying.

After determining the optimal k-value to be 1, we trained our model via KNN from 25% of the original dataset and tested the model on the remaining 75% of the dataset.

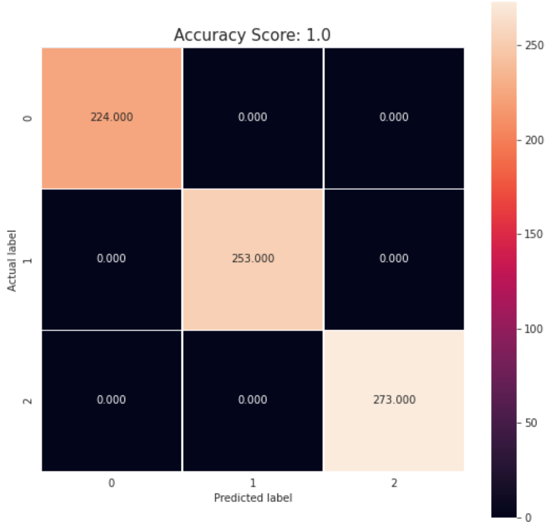


Figure 6: Confusion Matrix

The confusion matrix is a result of one instance of the process. The confusion matrix calculates the number of false negatives/positives that can occur during one iteration of the testing and training process. We defined a false positive as when the predicted is a higher severity than the actual severity. For instance, the algorithm deems a patient with high lung cancer severity when in reality the patient has low lung cancer severity.

On the other hand, a false negative is when the predicted lung cancer severity is lower than the actual lung cancer severity. For example, the algorithm classifies a patient with having low lung cancer severity when in reality the patient has high lung cancer severity. For this iteration, the model came out to be around 100% accurate, with having 75% of data being tested. As the test data size increased, we noticed that the accuracy only slightly diminished. We determined that these results are the product of a high quality dataset.

In the healthcare setting, an algorithm can only be implemented if it is not only accurate, but also consistent. To see whether this was the case for ours, we ran 100 iterations of the training and testing process with KNN and recorded the accuracy scores. Each iteration contains a randomly assigned testing and training dataset. The graph depicted in Figure 7 will reflect said iterations.

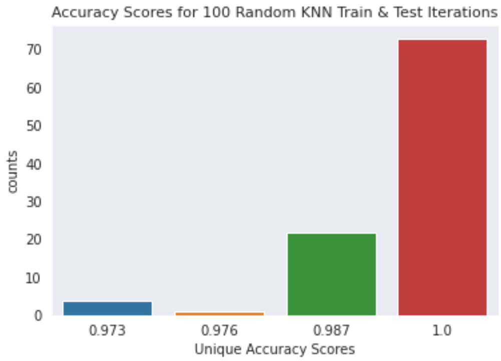


Figure 7: Plot of accuracy scores for 100 iterations of KNN train and test trials. 73 instances of 100%, 22 instances of 98.7%, 4 instances of 97.3%, and 1 instance of 97.6%

Figure 7 cements the fact that once we fit KNN to our training dataset (consisting of a random 25% of the PCA-transformed dataset), we consistently see high accuracy scores (ranging from 97% to 100%) when comparing our predictions for the testing dataset to their actual results. Also, the high testing to training size ratio highlights the algorithm's generalizability. As mentioned previously, these great results are primarily a testament to our high quality dataset and success with PCA & KNN.

III. Discussion

Our lung cancer machine learning model aimed to accurately predict a patient's lung cancer severity based on the parameters comprising common risk factors, which are mainly lifestyle choices. With medical data generally being imperfect with data redaction, there were issues finding a dataset that would create a good model. The lung cancer dataset was mainly integer values and did not have any missing data, making it a perfect dataset to work with. As a result, this model can significantly increase hospital efficiency when treating lung cancer patients by accurately categorizing patients' lung cancer severity, which will reduce stress on physicians and nurses by creating a more streamlined diagnostic process, and increase lung cancer survival rates for patients.

In the KNN part of the model, it is deemed as one of the more simplistic ways of implementation for testing and training. Though many in the machine learning community have deemed it a "lazy" way for classification and regression, it does give slight insight into classification of the various levels of cancer as previously discussed and is a great check for other methods either statistical or machine learning based such as PCA and linear regression.

Future improvements of this project would comprise of applying it to different lung cancer data sets to assure the high accuracy is not limited to the working dataset, as the current Lung Cancer Patient dataset provided by Kaggle is irregular in the sense that there are no missing parameters, NaNs, redacted information, or corrupted files, which is generally common in medical datasets. Furthermore, the model output upwards of ~98% accuracy on average as seen in Figure 7, even when testing consisted of 95% of the total dataset, which is an extreme that is not generally considered for real-life use. Therefore, it is pertinent to test our model against imperfect datasets as this high quality dataset brings concern for stable reproducibility for lower quality datasets. Additionally, as this is a health dataset, it is pertinent to reduce false positives, a low severity predicted as a high, and negatives, a high severity predicted as a low, so doctors provide adequate treatment as soon as possible. Our current priority is to reduce false negatives as although this leads to inefficient hospital resource allocation, it is safer than false positives, in which someone with high severity may be brushed off as a non priority.

Additionally, with some tweaking, this KNN model can be extended to other cancers and diseases that have definitive severity levels, allowing hospitals to increase efficiency beyond just lung cancer.

IV. Bibliography

[1] K.-J. Wang, J.-L. Chen, and K.-M. Wang, "Medical expenditure estimation by Bayesian network for lung cancer patients at different severity stages,"

Computers in Biology and Medicine, 24-Jan-2019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0010482519300174.

[2] C.-C. Pan, P.-T. Kung, Y.-H. Wang, Y.-C. Chang, S.-T. Wang, and W.-C. Tsai, “Effects of Multidisciplinary Team Care on the Survival of Patients with Different Stages of Non-Small Cell Lung Cancer: A National Cohort Study,” *PLOS ONE*. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0126547.

[3] C. S. Dela Cruz, L. T. Tanoue, and R. A. Matthay, “Lung cancer: epidemiology, etiology, and prevention,” *Clinics in chest medicine*, Dec-2011. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3864624/.

[4] T. Coleman, *Principal Components Analysis.MP4*. San Diego, CA, USA: Todd P. Coleman, October 25th 2020 Available: https://drive.google.com/file/d/148GMs2timapj5TedpmxHEBKc2IBCWpWX/view

[5] *How kNN algorithm works*. YouTube, 2014. Available: https://www.youtube.com/watch?v=UqYde-LULfs