University of Windsor
COMP4730

Final Project
Breast Cancer Subtypes

Due:
December 16, 2019

Submitted By:
Kolby Sarson - 104232327
Brandon Ferrari - 104396553

Summary

This paper will cover the topic of classifying breast cancer by molecular subtype. An introduction to breast cancer and its subtypes will be provided, alongside and introduction to the problem domain. Next, we will discuss our proposed solution to the problem followed by the display of our results and findings throughout the research. After seeing our results and findings, there will be a discussion of said results and findings followed by a discussion of the ethical implications of applying machine learning, in each of the societal, legal, and medical domains. Finally, we will close with our conclusions drawn from our research and any next steps we believe should be taken.

Introduction

Breast cancer is the most common cancer among women, accounting for approximately 25% of all new cancer cases in females. Additionally, 13% of all deaths caused by cancer in women can be attributed to breast cancer. Breast cancer is a fear for many women and the fact that it cannot be classified accurately is a scary thought for most. A misclassification, especially in the early stages of cancer, can have a drastic effect on the outcome experienced. The subtypes of breast cancer all have unique implications on how the patient will be impacted by the diagnosis. Some subtypes are more terminal than others while some are relatively harmless. Receiving an incorrect diagnosis could impact the course of action taken, when treatments are started (possibly prematurely or belated), the mental state of the patient and their family, or even the patient's current way of life. All in all, being able to accurately classify breast cancer by subtype is an important aspect in how cancer diagnoses are made, and this classification is what our research will investigate.

Problem Description

The problem we opted to solve is "Project 1: Classifying Breast Cancer Patients." This problem called for the classification of breast cancer patients into one of five classes; Normal, LumA, LumB, Her2, or Basal. These classes represent the five main molecular subtypes of breast cancer, being Normal-like, Luminal A, Luminal B, Her2-Enriched, and Triple-Negative/Basal-like. The classification of these subtypes was to be done by analyzing the gene expression ratios of 13,582 genes from a dataset of 158 samples. This indicates that there are 13,582 unique features in the dataset, which should also be considered during our investigation. With such a large number of features, the machine learning concepts of feature selection and dimensionality reduction should be examined.

Solution Description

The approach we opted to take to the problem outlined was to implement a variety of known classification, feature selection, and dimensionality reduction techniques in scikit-learn. We also decided to try some combinations of known techniques, as well as implementing one technique of our own, which we chose to name Variance Selection. This technique uses all features over a certain threshold of importance based on variance. The list of techniques we chose to use in our research includes Support Vector Machine with Linear Kernel (SVM-Linear), Support Vector Machine with Radial Basis Function Kernel (SVM-RBF), Random Forest Regressor (RFR), Kernel Principal Component Analysis (KPCA), and our personal Variance Selection (VS). The methods we included in our program are as follows: SVM-Linear, SVM-RBF, RFR, VS with SVM-Linear, VS with RFR, KPCA with SVM-Linear, and KPCA with RFR. All the above techniques and methods will be elaborated upon as we step through our results. We tested each of our proposed methods with a mixture of various test values (1, 2, 4, 6, 8, 10, 25, 50, 100) using loops to test the effect of changing various parameter inputs. Since we saw that a max depth of 6 performed best in straight RFR, we used 6 for the max depth in the VS/KPCA with RFR method, then used our test values on the VS/KPCA rather than the RFR.

Our solution creates an array of the methods we are using as well as an array of our test values. We iterate through our methods with an inner nested loop iterating through our test values. Within this nested loop are if statements which each handle one of our proposed methods. Finally, once all methods are tested, we output the results in a tabular display to summarize and compare all the methods and accuracies into one simple output, exporting it to a CSV file as well.

Before we begin discussing our methods, let us reiterate that the dataset being processed involves 158 rows of data that include 13,582 genes (or features). Thus, we are using a small dataset with a large number of features. When we referenced the course slides given by Dr. Rueda, we saw that Support Vector Machines (SVM) is a widely used classification algorithm that works best with datasets such as ours. We found that this would be a good starting point for our experimentation.

SVM

The main idea behind SVM is to derive a linear (or nonlinear) classifier based on a convex optimization problem, that is, we are looking for a convex (minimizing) function. The overall goal of SVM is to acquire the hyperplane which can best classify the data points, or one that creates the greatest separation between classes. SVM as a classification technique is one that relies on the use of support vectors, which are the data points closest to the hyperplane. These support vectors affect the location and orientation of the classifying hyperplane and help to find the one which creates the best separation of classes.

Given our experiences from past assignments, we chose to pass the data through SVM and test what kind of output is given based on the chosen Kernel. In this case, we chose to implement SVM using the Linear Kernel, as well as the RBF Kernel.

Kernel Function

In machine learning, it is quite common that we run into the issue of trying to perform linear classification on a nonlinear problem. That’s where kernel functions come into play. A kernel function is a function that, once applied to each data point in a dataset, can map the dataset to a higher dimension. The reason we wish to map a non-linearly separable dataset to a higher dimension is because once it is in a higher dimension it may become linearly separable. The process of using a linear classifier to solve a nonlinear problem is also known as the kernel trick.

Linear Kernel

Linear Kernel is a function in which the value is determined by the inner product of two vectors, plus an optional constant. This kernel function is used when the boundary between classes is assumed to be linear.

Radial Basis Function Kernel (RBF)

RBF Kernel is a function in which the value is determined by the distance from the origin (or some other point given). This kernel function is used when the boundary between classes is assumed to be curved.

After implementing SVM with each of Linear Kernel and RBF Kernel, we could see that Linear outperformed RBF in terms of accuracy. Upon this discovery we decided to implement SVM-Linear in our later methods (VS with SVM-Linear and KPCA with SVM-Linear).

Random Forest Regressor (RFR)

Before we go over the technique itself, let us first explain the concept of random forests as a classifier. A random forest is a collection of decision trees that operate as an ensemble. This is known as ensemble learning. The basis of random forest is that a random sample from the dataset is chosen at each depth and is run through the many decision trees, with each tree outputting a specific label or class. Much like how voting is done in real life, the class that was outputted the most, or the class that has the most ‘votes’, is the label that is given to the sample data. This vastly reduces the chance of overfitting since inconsistent features will not be able to skew the results in a different direction. Random Forest Regression works very similarly to Random Forest Classification, however there is one key difference that sets itself apart from the other. RFR takes the mean prediction of the individual trees rather than the most outputted class.

When implementing RFR into our code, the testing comes from choosing a value for the max depth that the decision trees can reach. For example, say we chose the max_depth = 3, this means that each decision tree will only be as deep as 3 nodes. While this may decrease accuracy, it greatly improves computation time. This is because the algorithm doesn’t need to run data through many large decision trees, which would be very taxing on the system performing these computations.

Kernel Principal Component Analysis (KPCA)

In order to explain the concept of Principal Component Analysis (PCA), we must first explain the idea of Dimensionality Reduction (DR) in machine learning. We will explain DR by comparing it to another widely used machine learning technique which is Feature Selection (FS). FS involves choosing a subset of features from a large dataset in order to reduce noise and computation time. DR behaves similarly to

FS, however, here we focus on elimination rather than selection. With dimensionality reduction, we view the dataset as a dimension size, with the size corresponding to the number of features. For example, a dataset with three total features is referred to as a 3-dimensional dataset. Thus, we are essentially reducing the dimension of a dataset when we eliminate these features.

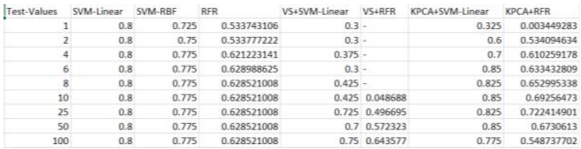
PCA is a method of DR that generates principal components for each feature and reduces the dimension of the dataset by removing all but the top n principal components, with n being a specified number of components by the user. In our solution we decided to implement KPCA which is simply PCA but with kernel implementation. We ultimately decided to use the linear kernel in our solution. It should be noted that KPCA with a linear kernel is no different than PCA on its own.

Variance Selection (VS)

This is a method of feature selection that was found outside of the scope of the course and is a popular starting point for solving any feature selection problem. This method involves computing the variance of every feature in the dataset and choosing a select number of features based on a specified threshold. We took this technique and made one minor tweak to how it works. Instead of filtering the features that fall below a specified variance threshold, we modified it to work similarly to Principal Component Analysis, where the top n features are taken, where n is the specified number of features. This lets us control how many features are returned and made the process much more consistent with the other experiments.

Our reason for filtering the features based off their variances is due to the fact that the higher the variance of a feature is, the more influence it has on the final classification on the specific row of data. Thus, when we chose the 20 highest variances in the list, we are choosing the 20 top 20 features in the dataset. While this may remove many features that could help with the classification process, it should still keep the accuracy relatively high while drastically minimizing the computation time when processing the data and training the machine learning algorithms.

Results



Test-Values	SVM-Linear	SVM-RBF	RFR	VS+SVM-Linear	VS+RFR	KPCA+SVM-Linear	KPCA+RFR
1	0.8	0.725	0.533743106	0.3	-	0.325	0.003449283
2	0.8	0.75	0.533777222	0.3	-	0.6	0.534094634
4	0.8	0.775	0.621223141	0.375	-	0.7	0.610259178
6	0.8	0.775	0.628988625	0.3	-	0.85	0.633432809
8	0.8	0.775	0.628521008	0.425	-	0.825	0.652995338
10	0.8	0.775	0.628521008	0.425	0.048688	0.85	0.69256473
25	0.8	0.775	0.628521008	0.725	0.496695	0.825	0.722414901
50	0.8	0.775	0.628521008	0.7	0.572323	0.85	0.6730613
100	0.8	0.775	0.628521008	0.75	0.643577	0.775	0.548737702

The above table shows the accuracies of each of our proposed methods for each given test value.

Results Discussion

SVM-Linear

Using SVM-Linear, we apply our test values to the C parameter which signifies the strength of regularization in the algorithm. This means we are specifying the degree to which we want the algorithm to avoid misclassifying the data. We can see that the test values had no impact on the accuracy, with SVM-Linear maintaining a constant accuracy of 80%.

SVM-RBF

Using SVM-RBF, our test values were again applied to the C parameter in the scikit library. We saw results were slightly worse than SVM-Linear, also staying constant after reaching the test value of 4, maxing out at 77.5%.

RFR

Using RFR, we can see that it performs much worse than either of SVM-Linear or SVM-RBF. Using a test value of 6 (which dictates max depth in RFR) we see our best result for this method, at a low 62.9%. Seeing that the test value of 6 was the best performer, this value will be used for the max depth in later methods involving RFR.

VS with SVM-Linear

Using VS with SVM-Linear, we can see that combining our VS technique with SVM-Linear seems to approach the results of straight SVM-Linear. Instead of constantly maintaining 80%, it appears to move closer to 80% as we increase the test value, with our data capping at 75% for a test value of 100.

VS with RFR

Using VS with RFR, we see that until we reach a test value of 10, the results are all negative, afterwards it jumps very quickly from 4.9% to 64.4%. At a test value of 100 and an accuracy of 64.4%, VS with RFR exceeds that maximum accuracy of straight RFR. It is unlikely that increasing the test value further would have any major impact as the current increase of 50 only produced an accuracy increase of approximately 7%.

KPCA with SVM-Linear

Using KPCA with SVM-Linear, we saw our best results, capping out at 85% at test values of 6, 10 and 50. To avoid overfitting, the best result would be 85% at a test value of 6, since the test value for this method was number of features. This method also showed us that combining methods can have positive impacts on results in some cases, as it improved upon straight SVM-Linear which maxed out at 80%.

KPCA with RFR

Using KPCA with RFR, we can see that similarly to combining RFR with VS, combining it with KPCA also helped it perform better than straight RFR. That being said, it still underperformed most other methods maximizing its accuracy at 72.2% for a test value of 25.

Ethical Discussion

There are many implications of using Artificial Intelligence (AI) or Machine Learning (ML) in the real world. These mostly revolve around the idea of incorporating AI and ML into everyday life, including ethical issues both legally and societally, as well as ethical implications on our breast cancer classification research.

Overview

To begin, AI and ML are used to simplify and/or improve (speed, accuracy, cost, etc.) upon the current way a task is done. This could be done in a multitude of ways such as automating processes to save wages, using precision machines for precise tasks, or automating long processes for quicker results.

One example of that is the automation that Amazon has implemented, both in their warehouses, and with their delivery drones. Amazon has incorporated a variety of AI and ML based machines into their business model. In their warehouse they have small robots, which look like larger scale Roombas, that can pick up and transport shelves of products easily and efficiently. They also have robotic arms that can do the heavy lifting and transporting of heavy products. Outside of the warehouse, Amazon has the aforementioned delivery drones which can deliver small packages in large cities much easier and quicker than delivery drivers. But this automation is not perfect. Just this past Black Friday weekend (2019) we saw that Amazon's automation failed customers by making multiple incorrect deliveries to purchasers of Nintendo Switch gaming consoles. Amazon instead delivered incorrect products such as batteries, books, tambourines, and even condoms, instead of the Nintendo Switches they had purchased.

So even though AI and ML can be seen to improve or simplify processes, they still are not guaranteed to be flawless, and we will look at some of the impacts of incorporating AI into society.

Societal Impacts

Though AI machines can improve the simplicity and performance of some tasks, they might also act as replacements to human labour. Jobs such as forklift drivers, delivery drivers, and order pickers have all been made obsolete in Amazon's company due to the implementation of these AI machines. Additionally, the fact that the majority of the work is done by these robots may lead to laziness amongst the remaining human employees due to the reduced workload. Worker morale may be negatively affected by the introduction of robots following their replacing of fellow employees. These are just a few examples of how AI machines and unemployment can affect our humanity as a whole.

There are also the issues revolving around inequality with regards to AI machines. If AI machines do reach the point of replacing most work, how will the wealth be distributed? The most likely outcome is that the owners or founders of the technology will retain most of the wealth while those replaced by these machines will be left to suffer. This is an issue that would need to be addressed prior to reaching a post-labour society.

Additionally, we have the issue of security and misuse. As technology advances its beneficial uses, it also advances its nefarious uses. Thus, the security of AI technology must be considered before being used openly as the negative impact of it getting into the wrong hands could be drastic, which we can see if we look at ransomware. There were over 200 million ransomware attacks in 2018 with 75% of those requesting more than $500 USD. This shows how lucrative the criminal uses of technology can be, and AI technology would be no different.

Legal Implications

Our society maintains order through rewards and punishments for good or bad deeds. But how does one punish a technology? Would reprimandations even have any impact on technology? If not, then who should be held liable for the technology, if anyone? These are the questions we need to ask when discussing the incorporation of AI machines into society.

If something goes wrong with a technology and loss of life is experienced, then who is to blame? We could blame the designer of the technology for not making it one hundred percent safe. Or we could blame the user for accepting to use a technology that had the potential to be lethal, no matter how small of a chance. Thus, we have the issues involving the decision itself, how we make the decision, and whether it's an ethical one or not.

We can also look at the legal implications of the trolley problem and its variants. In summary, the trolley problem poses a decision on who should be killed if a trolley accident is unavoidable, but the people who might die are unique. For example, if a car is going to hit someone regardless, but it could either hit a grandma or an infant, who should it hit? The legal ethics of this issue are apparent when we look at this problem since we are deciding who should be killed. Situations like these must be accounted for when designing AI technology. We must also consider whether this is considered a crime or not, such as murder or manslaughter, and if so, who would be held liable.

Ethics in Cancer Biomarker Diagnosis

Our project looks at identifying the subtype of breast cancer patients based on AI and ML techniques. In practice, this can lead to ethical issues arising. Misdiagnosing the subtype (which is plausible given the less than perfect prediction accuracy) can lead to different treatments and final outcomes. For example, if a subtype is misdiagnosed as a less lethal subtype, treatment may not be as aggressive as needed, leading to treatments that fail. On the flip side, if a subtype is misdiagnosed as a more lethal subtype, then a patient might go through unnecessarily aggressive treatments. Either of these could lead to more undesirable outcomes than if a correct subtype diagnosis was given, impartial to the fact that the patient still has breast cancer.

Additionally, the implementation of AI in subtype diagnosis may lead to less effective doctor diagnoses. Doctors may come to rely on these AI subtype diagnoses and begin failing to notice information or cues that would normally let the doctor give the correct diagnosis. Any of these situations would need to be considered before allowing the use of AI for medical diagnoses, whether it be breast cancer or others.

Conclusions

Through our research we concluded that KPCA with SVM-Linear was the most effective method from those explored. KPCA with SVM-Linear was able to predict breast cancer subtypes with 85% accuracy. In terms of percentages, 85% seems accurate. But in terms of a medical diagnosis, 85% could be seen as unfavourable still. Classifying breast cancer by subtype seems a simple task on paper, but consideration must be given to the fact that these predictions affect real people. We can not blindly assume that an 85% accuracy is acceptable because the 15% inaccuracy could be life changing for some. It could possibly lead someone to a death they may not have otherwise faced and this is reason enough to question our 85% accurate prediction method. For the future we think it would be beneficial to examine even more methods in the hopes that we could discover one with an acceptable level of accuracy.

Contributions

Kolby Sarson - 104232327

In terms of our code, I implemented the methods of SVM-RBF, RFR, and KPCA with RFR. I also implemented a tabular output that summarized our results and exported it to a CSV file.

As for the report, I was responsible for writing up the Summary and the Introduction. I also wrote the first half of the Solution Description, up to and including the RBF explanation. I added the Results section, and in the Results Discussion, I was responsible for methods that I personally implemented in our program (SVM-RBF, RFR, and KPCA with RFR). Finally, for the ethical discussion, I was responsible for the first half, up to and including the Societal Impacts.

Brandon Ferrari - 104396553

My implementation involved the use of several methods such as SVM-Linear, Variance Selection with SVM-Linear, Variance Selection with RFR, KPCA with SVM-Linear. I oversaw setting up the python environment with scikit-learn and getting our code ready so that we can begin the implementation of the various methods listed prior.

For the report, I oversaw the write-up for the problem description and the conclusion. I also wrote the second half of the solution description starting from Random Forest Regressor. I also did the write-up for the results discussion involving the methods I wrote in our python script (SVM-Linear, VS with RFR, and KPCA with SVM-Linear). Regarding the ethical discussion, I was responsible for the second half, starting from the legal implications.

References

Bossmann, Julia. “Top 9 Ethical Issues in Artificial Intelligence.” *World Economic Forum*, https://www.weforum.org/agenda/2016/10/top-10-ethical-issues-in-artificial-intelligence/.

Perper, Rosie. “Customers Are Fuming Saying They Received Black Friday Amazon Orders That Contained Condoms, Toothbrushes, or Even a Tambourine Instead of a Nintendo Switch.” *Business Insider*, Business Insider, 6 Dec. 2019, https://www.businessinsider.com/amazon-customers-sent-condoms-batteries-tambourine-instead-of-nintendo-switch-2019-12.

Gordon, Kyle. “Topic: Ransomware.” *Www.statista.com*, https://www.statista.com/topics/4136/ransomware/.

“Average Demanded Ransom from Ransomware Attacks 2017.” *Statista*, https://www.statista.com/statistics/701003/average-amount-of-ransom-requested-to-msp-clients/.

Rueda, L 2019, *C03-LinearClassifiers*, lecture notes, Advanced Topics in AI - Machine Learning and Pattern Recognition COMP-4730/8740-01, University of Windsor, delivered 19 September 2019.

Rueda, L 2019, *C09-DimensionalityReduction*, lecture notes, Advanced Topics in AI - Machine Learning and Pattern Recognition COMP-4730/8740-01, University of Windsor, delivered 14 November 2019.