Predictive Modeling of Cancer with GBDTs

Peter Jourgensen
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
pjourgensen@cs.ucla.edu

1 Introduction

Medicine is in the process of undergoing a development from treating ailments as they arrive to preventing them before they happen. Causal inference and research is paramount to understanding how to prevent diseases, but the complexity of these relationships remains an ongoing struggle. It is rarely one variable that causes a disease to occur, but rather many. As researchers attempt to uncover the effects of combinations of variables, they quickly run into a combinatorial explosion of options to consider and variables to control for. Traditional methods of research struggle on this pursuit, but machine learning is well adapted to it.

Machine learning tools have the capacity to take a high dimensional feature space and discover underlying patterns that lead to a particular disease. This can be used to accurately and quickly assess a patient's diagnose and recommend a proper treatment plan. Further, the tools take no predisposition as to what they find important, thus opening the doors for possibly finding causal relationships never previously thought of by researchers. These results can then be used to guide future research towards preventative measures.

In this project, I attempt to do just this by leveraging data from the CDC's National Health and Nutritional Examination Survey (hereby referred to as NHANES) to predict whether or not a patient has cancer. This data set contains over 500[1] survey responses and lab results of roughly 50,000 individuals labeled as either having or not having cancer at some point in their life. I begin by hand-selecting a list of potentially predictive features. I then pre-process them, filter out the optimal subset of features for prediction, and parameter tune a collection of models for optimal performance. I then make predictions on a held-out test set to assess the final performance[2] of the model.

2 Feature Selection and Pre-Processing

2.1 Initial Selection

My initial plan for feature selection was to analyze each variable independently. I was planning to scrape the NHANES website to get all of the variable codes, then individually pull the data for each feature and measure its mutual information with the target class. I would have set a threshold for mutual information and simply continued with every variable that was above that threshold. I quickly realized, however, that each feature requires personalized pre-processing. While it would be feasible to develop a method to automate the process of analyzing the structure of a feature and deciding upon the optimal pre-processing procedure, I decided this wasn't tractable for the scope of this project.

As an alternative, I continued by hand-selecting particular features that I deemed to be worthwhile. This process plays a critical role in the capability of the model. Features were selected that are

[1] An approximation.

[2] See Section 3.1 for detailed discussion of performance.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

generally correlated with Non-Communicable Diseases (NCDs) and that have research supporting their relationship to cancer specifically.

Cancer is among the NCDs. NCDs, or chronic diseases, are those that have long duration and generally slow progression. Common risk factors for NCDs include tobacco, harmful use of alcohol, insufficient physical activity, unhealthy diet, raised blood pressure, obesity, raised cholesterol, and raised blood sugar. Furthermore, the presence of NCDs - such as stroke, heart disease, respiratory disease, and diabetes - tends to result in higher rates of cancer[3]. These were the primary influences on whether or not to include a feature in the analysis. After these risk factors were considered, lab results for a number of metals and other volatile compounds that have shown a potential relationship to cancer in research were also considered. Ultimately, 81 features were chosen for initial analysis. The list of features, along with a brief description as to why it was chosen and which pre-processing function was applied to it are contained in the following table:

Table 1: Initial feature selection.

Code	Description	Reasoning	Pre-Processing
DEMO-RIDAGEYR	Age in years	Cancer more likely with older patients	preproc_cont
DEMO-RIAGENDR	Gender	Cancer may be more prevalent in one gender vs. the other	preproc_onehot
DEMO-RIDRETH3	Race	Cancer may be more prevalent with particular races	preproc_onehot
DEMO-DMDEDUC2	Education Level	Education may have implication on lifestyle	preproc_onehot
DEMO-DMDHHSIZ	Household Size	Household size may have implication health	preproc_onehot
DEMO-INDHHIN2	Household Income	Household income may have implication on lifestyle	preproc_cont
EXAM-BPXCHR	60-sec HR	Heart rate may have implications on general health	preproc_cont
EXAM-BPXPLS	60-sec Pulse	Pulse may have implications on general health	preproc_cont
EXAM-BPXSY2	Systolic Blood Pressure	Blood pressure may have implications on general health	preproc_cont
EXAM-BPXDI2	Diastolic Blood Pressure	Same as above	preproc_cont
EXAM-BMXWT	Weight	Higher weight may have adverse effects on health	preproc_cont
EXAM-BMXHT	Height	To be used in combination with weight	preproc_cont
EXAM-BMXBMI	Body Mass Index	Additional measure of general health	preproc_cont
EXAM-FCX02DI	Fluorosis DI (2M)	Curious whether any relation exists with cancer	preproc_onehot
EXAM-OHAROCDT	Decayed Teeth	Dental care may reflect overall health	preproc_onehot
LAB-URXUMS	Albumin level	May be marker of undiagnosed cancer	preproc_cont
LAB-URX4TDA	Diaminotoluene Level	It is a carcinogen	preproc_cont
LAB-URXUAS	Urinary Arsenic	Toxic compound that may be related to cancer	preproc_cont
LAB-LBXBPB	Blood Lead	Lead is a known toxic chemical	preproc_cont
LAB-LBXBCD	Blood Cadmium	Cadmium is a known carcinogen	preproc_cont
LAB-LBDBMNSI	Blood Manganese	Studies have shown correlation with Mn and cancer	preproc_cont

[3] According to the World Health Organization

LAB-LBXTC	Total Cholesterol	High cholesterol associated with adverse health effects	preproc_cont
LAB-LBDBCRSI	Chromium level	Considered a carcinogen	preproc_cont
LAB-LBDBCOSI	Cobalt level	Considered a carcinogen at certain levels	preproc_cont
LAB-LBXWBCSI	White blood cell count	Low count may make patients more susceptible to disease	preproc_cont
LAB-LBXRBCSI	Red blood cell count	Extremes may pose adverse health effects	preproc_cont
LAB-LBXHGB	Hemoglobin	Extremes may pose adverse health effects	preproc_cont
LAB-LBXPLTSI	Platelet count	Extremes may pose adverse health effects	preproc_cont
LAB-LBDSCUSI	Serum Copper	High copper associated with certain types of cancer	preproc_cont
LAB-LBXCOT	Cotinine level	By-product of nicotine use	preproc_cont
LAB-LBXHCT	Hydroxycotinine level	By-product of nicotine use	preproc_cont
LAB-LBDRFO	RBC folate	Measure of general nutritional status	preproc_cont
LAB-LBXHBC	Hepatitis B	Chronic hepatitis B is a common risk factor for liver cancer	preproc_cont
LAB-LBXHCR	Hepatitis C	Chronic hepatitis C is a common risk factor for liver cancer	preproc_cont
LAB-LBXHP2C	Cobas HPV High Risk	HPV infections are a common cause of cervical cancers	Fill as "negative"
LAB-LBDINSI	Insulin level	Insulin levels can be an indicator of certain NCDs	preproc_cont
LAB-URXUAS5	Arsenic level	Arsenic exposure may have adverse health effects	preproc_cont
LAB-URX2MH	2-methylhippuric acid	Volatile compounds may have adverse health effects	preproc_cont
LAB-URXATC	2-amnothiazolne-4-carbxylic acid	Volatile compounds may have adverse health effects	preproc_cont
LAB-URXDPM	N-ace-S-(dimethylphenyl)	Volatile compounds may have adverse health effects	preproc_cont
LAB-LBXV1D	Dichlorobenzene level	Inhalation has produced tumors in mice	preproc_cont
LAB-LBXV4C	Tetrachloroethene level	Related to bladder cancer	preproc_cont
LAB-LBXVBZN	Benzonitrile level	It is on the hazardous substance list	preproc_cont
LAB-LBXVCT	Tetrachloride level	Inhaling caused liver tumors in animals	preproc_cont
LAB-LBXVIBN	Isobutyronitrile level	It is on the hazardous substance list	preproc_cont
LAB-LBXVBZ	Benzene level	It is on the hazardous substance list	preproc_cont
QUES-ALQ130	Alc drinks per day	High consumption of alcohol may have adverse health effects	preproc_cont
QUES-BPQ050A	Taking medicine for HBP	High blood pressure related to adverse health	preproc_onehot
QUES-CDQ001	Chest Pain	Representative of cardiovascular health	preproc_onehot
QUES-CDQ006	Chest Pain Relief	Representative of cardiovascular health	preproc_onehot

QUES-HSD010	General Health	Patient assessment of overall health	preproc_onehot
QUES-HSQ510	Stomach Illness	May be predictive of larger condition	preproc_onehot
QUES-HSQ520	Flu, Ear Infection	May be predictive of larger condition	preproc_onehot
QUES-DIQ170	Diabetes risk	Diabetes considered to double risk of certain types of cancer	preproc_onehot
QUES-DUQ211	Marijuana Use	Drug use may have adverse health effects	preproc_onehot
QUES-DUQ217	Marijuana Use	Drug use may have adverse health effects	preproc_onehot
QUES-DUQ272	Cocaine Use	Drug use may have adverse health effects	preproc_onehot
QUES-DUQ290	Heroin Use	Drug use may have adverse health effects	preproc_onehot
QUES-DUQ330	Meth Use	Drug use may have adverse health effects	preproc_onehot
QUES-DUQ430	Rehab Attendance	Drug use may have adverse health effects	preproc_onehot
QUES-MCQ010	Asthma	NCDs are related to higher cancer risk	preproc_onehot
QUES-AGQ030	Hay-fever	Possible comorbidity	preproc_onehot
QUES-MCQ053	Anemia	Possible comorbidity	preproc_onehot
QUES-MCQ080	Overweight	Possible comorbidity	preproc_onehot
QUES-MCQ160b	Congestive Heart Failure	Possible comorbidity	preproc_onehot
QUES-MCQ160c	Coronary Heart Disease	Possible comorbidity	preproc_onehot
QUES-MCQ180e	Heart Attack	Possible comorbidity	preproc_onehot
QUES-MCQ160f	Stroke	Possible comorbidity	preproc_onehot
QUES-MCQ180g	Emphysema	Possible comorbidity	preproc_onehot
QUES-MCQ160m	Thyroid Problem	Possible comorbidity	preproc_onehot
QUES-MCQ160k	Bronchitis	Possible comorbidity	preproc_onehot
QUES-MCQ160l	Liver Condition	Possible comorbidity	preproc_onehot
QUES-MCQ203	Jaundice	Possible comorbidity	preproc_onehot
QUES-DPQ020	Depression	Presence of cancer may impact mental state	preproc_onehot
QUES-DPQ030	Sleep issues	Presence of cancer may impact mental state	preproc_onehot
QUES-DPQ040	Low energy	Presence of cancer may impact mental state	preproc_onehot
QUES-DPQ050	Poor appetite	Presence of cancer may impact mental state	preproc_onehot
QUES-SLQ120	Feel sleepy	Presence of cancer may impact mental state	preproc_onehot
QUES-SMQ020	Smoked 100 Cigarettes	Smoking is a known cause of cancer	preproc_onehot
QUES-SMD470	Household Smokers	Second-hand smoke can be hazardous	preproc_onehot
QUES-WHD020	Current Weight	Being overweight associated with many adverse health effects[4]	preproc_cont

[4]Reasonings are informed by the American Cancer Society and World Health Organization.

2.2 Pre-Processing

Pre-processing involves the handling of outliers and missing data, as well as the transformation and encoding of existing data. Ultimately a predictive model is only as good as the data that is fed to it, so pre-processing is imperative in setting a foundation for success. While there exist many fundamental methods in this field, there is no "one-size-fits-all" approach to handling data and the researcher must rely on prior experience and trial and error to determine what is most appropriate for a collection of data.

When handling outliers, a researcher must first define what qualifies as such, and must second define how they will be treated. Z-score and percentile thresholds are common tools for the identification phase. Percentiles are less impacted by extreme values and are, therefore, chosen for this analysis. Furthermore, the percentile thresholds can then be used as values to map the outliers to.

Handling missing values is a more a difficult task. In the case that the values are missing at random (MAR), imputing them with the mean or median may be appropriate. Alternatively, a linear regression, or higher order method, may be useful if the other features are well filled. There is additional error with every imputation, however, so caution must be taken if attempting to impute values based on previously imputed values. A researcher must also consider if there is an underlying reason behind why the value may be missing. If such a relationship exists, filling the values with the mean or median may institute additional bias in the model. However, this can be mitigated with the use of an indicator variable that signals whether or not the value was imputed. In other words, the state of the variable's presence becomes its own feature for the model.

While optimizing performance requires handling each feature individually to become familiar with its distribution and relation to the target class, I have elected to separate my features into categorical and continuous variables to each be handled with a broad schema. Please find rigorous descriptions of these schemas in the following sections and the code used for each in the appendix.

2.2.1 Categorical Variables

Categorical variables are initially analyzed to determine their distribution before being one-hot encoded. If a single response accounts for over 85% of the non-missing responses of a particular feature, then the missing values are filled with that response. Otherwise, no assumptions are made about the missing values before encoding. This logic is implemented in the function `preproc-onehot` and is used on all categorical variables.

2.2.2 Continuous Variables

Continuous variables require a few more steps. First, outliers, in this scope defined as values below the 1st percentile or above the 99th percentile, will be mapped to their respective boundary values. Second, the mean and standard deviation will be calculated and stored. Third, missing values will be filled with the median to be resistant to skew and a second feature will be generated to indicate if the value was imputed[5]. Finally, the original feature will be normalized according to the original mean and standard deviation. This process has been developed into a function called `preproc-cont` and applied to all continuous variables[6].

2.3 Filtering

Filtering methods are used to identify a subset of features that provide the greatest opportunity for optimal model performance. In general, a threshold is set for a given metric and, depending on the particular metric, features that are either above or below that threshold are eliminated from further analysis. For this analysis, a correlation filter is used to identify highly correlated features, a mutual information filter is used to identify features who share dependence with the target, and a variance filter is applied to remove features with limited predictive power. Lasso regression is also tested as a means of reducing the feature space, but `Table 2` illustrates that this reduction results in diminishing performance.

[5] 1 if imputed, 0 if not.

[6] Pre-processing results in 379 features

2.3.1 Correlation Filter

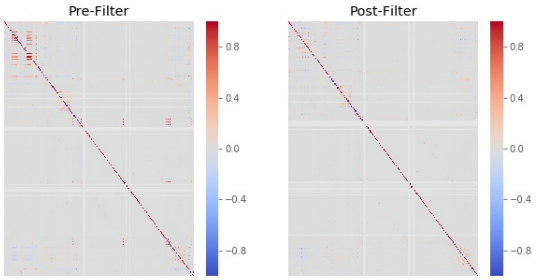

Figure 1: Filtering features with correlation > 0.7.

The first step in reducing the feature space is to identify highly correlated features and eliminate a single feature of the pair. This is known as a correlation filter and it serves multiple purposes. The first is removing redundancy from the feature space to improve computational cost without a potential loss in information. This reduction also promotes more generalized prediction as it reduces the potential for overfitting. Secondly, logistic regression and decision tree models are influenced by correlated features. While their performance may not suffer, the interpretability of their coefficients or branches may. Therefore, using fewer correlated features improves run-time without significant loss in performance and improves post-prediction analysis. That said, in this last step it is still imperative to compare predictive features to those that they are highly correlated with to assess which may be causal and which may be confounding. A Pearson correlation of 0.7 was used as the filter. `Figure 1` illustrates the effects of this process.

2.3.2 Mutual Information Filter

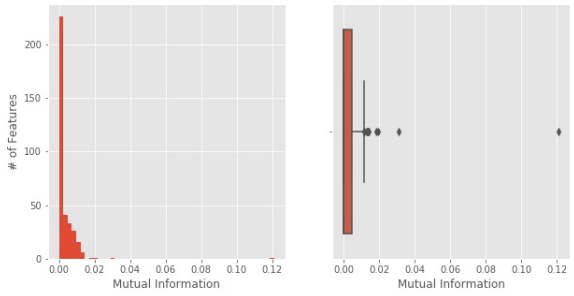

Figure 2: Distribution of mutual information scores.

Mutual information is used to measure the dependence between two variables. In this scenario, mutual information of each feature is measured with the target class to identify which features may have a relationship to cancer prevalence. If cancer happened to be completely deterministic from one of the features, they would have a mutual information of 1, while if they were independent, they would have a mutual information of 0. The benefit of mutual information over correlation and other statistical tests is that it is resistant to influence of non-linear relationships. This makes it a robust test to varying functions of dependency.

`Figure 2` illustrates the distribution of mutual information scores of the features with the target. It is evident that many features show independence from cancer, so the 25th percentile was used as the filter to eliminate these features. The maximum level of mutual information observed corresponded to a patient's age. This is logical considering the increase in cancer risk with age.

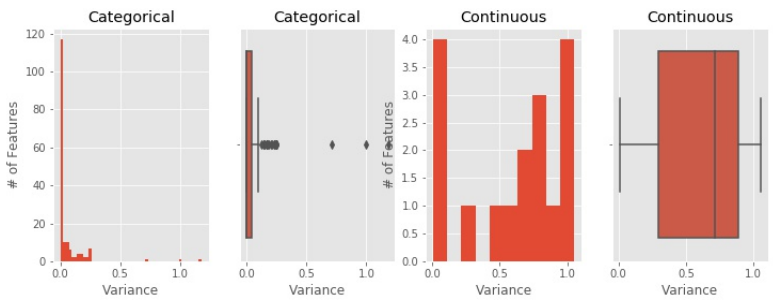

Figure 3: Distribution of variances by data type.

2.3.3 Variance Filter

A variance filter is used to identify features with greater variance levels. Features with low variance tend to have less predictive power and are, thus, removed from further analysis. Categorical variables and continuous variables were encoded on different scales, and, therefore, had to be handled separately. `Figure 3` illustrates the differences in distributions.

Many categorical variables showed no variance and thus had no predictive capabilities. A threshold of the 25th percentile was used to eliminate these.

Continuous variables were intended to be transformed such that their variance was 1. The descriptive statistics, however, were calculated based upon available data to be resistant to changes in the distribution brought about by imputing missing values. The imputations only serve to reduce the variance in the distributions which is why 1 is the max. There were still a few features that had 0 variance, however, so these were eliminated with a threshold of 0.

2.3.4 Lasso Regression

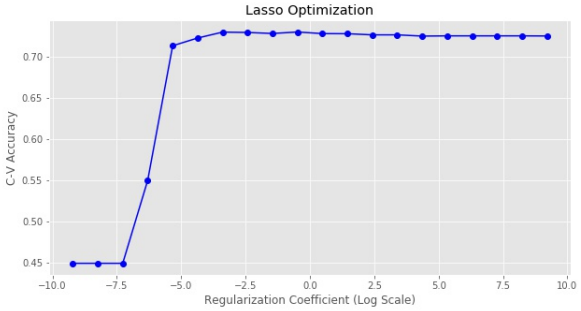

Figure 4: Optimization of regularization for lasso regression.

Lasso regression[7] adds a penalty to the absolute value of coefficients during optimization, therefore, pushing the coefficients towards zero. This form of optimization tends to lead towards models with few non-zero coefficients, lending itself to greater generalizability. Furthermore, these coefficients can then be used to identify the combination of the most predictive features in the feature space. For classification, logistic regression is used with an `L1` penalty with varying levels of regularization strength. `Figure 4` illustrates the effect of the regularization coefficient on cross-validated accuracy. The non-zero coefficients of the model that performs best[8] are used to identify the set of features that will be used for further optimization.

`Table 2`, however, illustrates that the features chosen by lasso regression resulted in a noticeable drop-off in model performance. There are a number of reasons why this may be the case. Foremost is

[7] Also referred to as L1-Regularization

[8] See section 3.1 for a detailed discussion on performance

the fact that the results of lasso regression are inconsistent on different folds during cross-validation. This instability may be related to the marginal values of mutual information of the features with the target class. Without clear relationships, small changes in a training set may have a significant impact on the non-zero coefficients generated by lasso regression. Without consistent results, this method is unjustified and, therefore, not considered for further analysis.

2.4 Feature Engineering

Feature engineering broadly refers to the creation of new features and transformation of current ones. Two examples of this process are Principal Component Analysis, which reduces the feature space by mapping features onto vectors that capture the highest amount variance, and polynomial feature inclusion, which increases the feature space by calculating higher order features and combinations of features. For this analysis, I chose to test model performance with the addition of polynomial features. This method, however, showed no improvement in predictive performance[9], but increased runtime significantly, so the additional features were not used for further analysis.

2.4.1 Polynomial Feature

As discussed in the introduction, it is unlikely that individual risk factors are making large contributions to the likelihood of having cancer, but rather many nonlinear combinations of these risk factors . To model these types of interactions, polynomial combinations of the remaining features are considered. As the degree of polynomial increases, however, a combinatorial explosion of features occurs and the risk of overfitting intensifies. As such, only a polynomial degree of 2 was considered in this analysis. As a possible extension, a researcher could look at the effects of higher order polynomials, along with other transformations[10], on the predictive capabilities on cancer.

After the generation of polynomial features, they were passed through the same filtering pipeline used previously to identify the subset that is highly variable and shares some dependency with the prevalence of cancer. `Table 2` illustrates, however, that these additional features provide no improvement in model performance[11], while incurring a significant computational cost. The lack of improvement stems from overfitting the training set. Therefore, they were not considered for further analysis.

2.5 Summary

Table 2: Summary of feature selection and pre-processing.

Method	Random Forest (%)	SVM (%)	Logistic Regression (%)
Unchanged	72.4	72.7	72.7
Correlation Filter	73.0	72.8	72.8
Mutual Info Filter	72.1	72.6	72.8
Variance Filter	72.5	72.7	73.0
Lasso Regression	63.1	63.7	64.0
Polynomial Features	72.2	72.5	71.5

Overall, 138 out of the original 379 features made it through the filtering pipeline. These features share the qualities of being uncorrelated with one another, sharing mutual information with cancer, and having high variability. The reduction in feature space promotes greater generalizability in the model and perhaps further isolates causal predictors. While both lasso regression filtering and polynomial feature inclusion were considered during this analysis, they were unjustified by performance loss and computational cost.

[9] See section 3.1 for a detailed discussion on performance.

[10] Logarithmic, Exponential, etc..

[11] See section 3.1 for a detailed discussion on performance

3 Predictive Modelling and Optimization

Once the data is encoded and filtered, predictive modelling and optimization can be performed to identify the model and parameterization that optimally performs on the data set according to some key performance indicator (KPI). The KPI for this analysis is discussed in depth in the following section.

Much like the step of pre-processing, however, there is no single model or parameterization that performs best on every data set. Therefore, many must be tested to identify the specific one that performs optimally. In this analysis, two baseline models - Decision Trees, Support Vector Machines - are tested to gauge a baseline performance, two ensemble methods - Random Forests, Gradient-Boosted Decision Trees - are then tested, and finally a densely connected neural network is tested.

3.1 Key Performance Indicator

The key performance indicator that I used to judge the performance of a given model and data set was a 5-fold cross-validation score. Cross-validation is the process of dividing a training data set into k folds, training a model on k-1 folds and predicting on the remaining for each combination of folds. Prediction accuracy is then measured on each of the 5 trials and averaged. This method is resistant to outlier distributions of testing sets and generalizes well to unseen data.

The trade-offs with precision and recall are also important to consider during analysis. Precision is the proportion of true positives to the total number of positive predictions. In the context of cancer, it is the proportion of patients the model correctly predicted as having cancer to the total number of people it predicted as having cancer. A perfect precision of 1 would mean that whenever the model predicted cancer, it was correct. Alternatively, recall is the proportion of true positives to the total number of true occurrences. In this context, it is the proportion of patients the model correctly predicted as having cancer to the total number of patients that have cancer. A perfect recall of 1 would mean that every patient who has cancer was predicted as such. To assess which metric is more applicable, one must determine if there is greater risk in wrongly classifying a patient as having cancer (low precision, high recall) or wrongly classifying a patient as not having cancer (high precision, low recall).

These metrics are also influenced by imbalances in the target class. In the NHANES data set, only 10%[12] of patients are labelled as having cancer. Therefore, a model trained on the entirety of the data set may achieve high accuracy by always predicting that a patient does not have cancer, but achieve nothing in terms of precision or recall. However, if a model is trained on a balanced subset of the data, it is far less susceptible to such extreme results and puts a natural balance on both precision and recall. Ultimately, there are pros and cons of each performance metric. A patient wrongly classified as having cancer may be unnecessarily exposed to harmful chemotherapy. A patient wrongly classified as not having cancer may not receive the medication and therapy they need to fight the disease. I chose to make no assumptions with respect to which is more significant. This is why for the purpose of this research, I will only be focusing on cross-validated accuracy when trained on balanced data sets.

3.2 Baseline Models

Baseline models are used to determine a baseline level of performance for which more complex models can strive to improve upon. For this analysis, decision trees and support vector machines were chosen for this task.

3.2.1 Decision Trees

The first model I chose to consider for cancer prediction was a decision tree. Decision trees work by recursively forking on particular values of features to generate the greatest amount of information gain[13] in the target class. The benefits of decision trees are their computationally efficient training times, their ability to learn nonlinear functions, and their clear audit trails at prediction time. The downsides of decision trees are that they are quickly susceptible to overfitting as the depth of the tree increases and are unstable in the sense that small changes in the data may lead to large changes in the

[12] An approximation.

[13] Defined as the greatest reduction in entropy

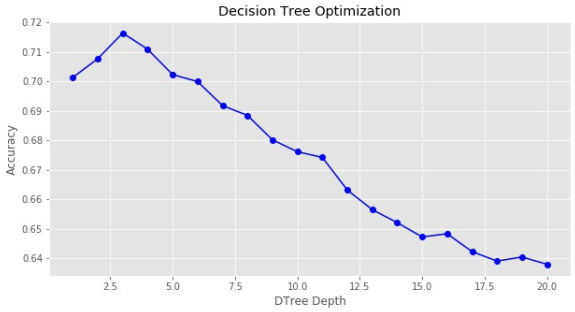

Figure 5: Parameter tuning of decision trees.

model. In a application setting, decision trees also struggle with updating to new data, as they must be entirely retrained with each addition.

As the depth of decision trees increases, the risk of overfitting intensifies. Taken to the extreme, unless multiple samples have the exact same feature vector, they will ultimately be given their own leaf node for classification. The effects of this behavior are illustrated in `Figure 5`. Performance is maximized at a tree depth of 3 before steadily declining with each additional branching level. A decision tree with a depth of 3 produced a cross-validation score of 71.8%. This will serve as a good indicator for the performance of more robust models.

3.2.2 Support Vector Machines

Table 3: Summary of SVM trials (C-V Accuracy %).

Regularization Strength	Linear	Radial Basis Function
0.1	73.0	72.4
1.0	**73.1**	72.7
10	72.7	72.5

The second model I chose to consider for baseline performance was a support vector machine. Support vector machines work by attempting to find a linear hyperplane capable of separating the data set into 2 categories for classification. The optimal hyperplane is the one which maximizes the distance to the closest points. This is known as the margin. Maximizing the margin is used to promote greater generalizability in the classifier. Data, however, is rarely perfectly separable, so a regularization parameter is introduced to allow for varying amounts of misclassification. This must be tested at a range of values to ensure model optimality. Support vector machines can then be extended to identify non-linear hyperplanes by passing the features through kernel transformations. When working in high dimensions where it is difficult to visualize the structure of the data, it is important to test multiple kernel functions to ensure model optimality.

For this research, I considered a regularization parameter range of `[0.1,10]` in log-space with linear and radial basis function kernels. `Table 3` illustrates the results of each of the trials. A linear model with a regularization of 1.0 generated the highest cross-validated accuracy. It achieved 73.1%, thus setting a new baseline for which to compare to more complex models. While the differences in model performance were marginal, the effects of the regularization strength are clear. Low values in the parameter allow for fewer misclassifications, which risks greater overfitting, as evidenced here with the parameter set to 0.1. High values in the parameter, however, allow for greater misclassifications, resulting in underfitting, as evidenced here with the parameter set to 10. Finally, it should be noted that the polynomial kernel was not tested because consideration of polynomial features had already been eliminated from further analysis.

3.3 Ensemble Models

Ensemble models are built on the hypothesis that combining multiple models may produce a stronger single model. Generally, the collection of initial models are referred to as weak learners, as they tend to perform poorly on their own. Ideally, however, these models are structured in a way where they do not make the same mistakes as each other. Therefore, when enough weak learners are combined with a voting system, the consensus predictive capabilities can be quite strong. Random forests and gradient-boosted decision trees were chosen as ensemble models for this analysis.

3.3.1 Random Forests

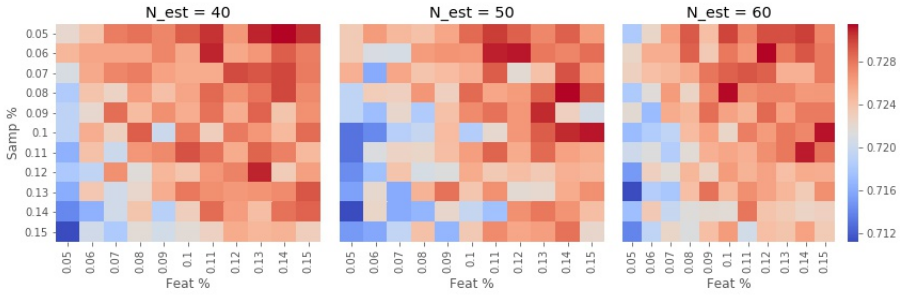

Figure 6: Parameter tuning of random forest.

Random forests are an implementation of a general ensemble method known as bagging. Bagging is the process of independently developing a collection of weak learners before passing them through a voting system. The driving principle is that each of the weak learners may capture a separate piece of the underlying structure of the data, while the whole is able to leverage the voting system to capture the true overall pattern.

In the case of random forests, these weak learners are decision trees developed on random subsets of the data. Initially, a selection of samples is randomly chosen. Then, at each branching opportunity, a selection of features is randomly chosen to be considered for forking. The number of samples chosen for a particular tree and the number of features considered at a specific decision point must be tuned by the researcher to ensure optimal model performance. If both are too high, there is risk for simply re-developing the same tree multiple times. If both are too low, there is risk for not identifying the true signal in the data. The overall number of trees must also be managed to control for overfitting.

For this research, I performed a grid search whereby I initially considered the number of estimators in the range `[50,150]`, the percentage of samples in the range `[0.1,0.7]`, and the percentage of features in the range `[0.1,0.7]`. The model at all of the lower bounds ended up performing optimally, so I then did a more detailed search around that parameterization. `Figure 6` illustrates the results of this secondary grid search. It is clear that combinations of lower sample percentages and higher feature percentages perform better, regardless of the number of estimators. This is an interesting result as lower sample percentages make the individual models weaker, but high feature percentages make the individual models stronger. Ultimately, the best parameterization discovered by this search was a model with 60 estimators, each using 6% of the samples and 12% of the features, performed the best overall. This model achieved a cross-validated score of 73.2%.

3.3.2 Gradient-Boosted Decision Trees

Another popular ensemble method is known as boosting. Rather than training a collection of models independently, this method relies on iteratively developing models whose training depends on the previously developed models. At each iteration, a model is trained to correct for the error of the previously developed models.

In the case of gradient-boosting, each iterative model is trained on the residuals[14] of the previous collection of models. After the models are developed, they are then weighted and averaged for final

[14]Negative gradient of loss function.

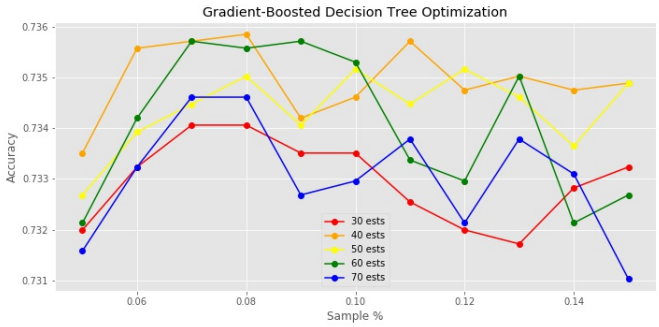

Figure 7: Parameter tuning of gradient boosted decision trees.

classification. Both the number of estimators and the number of samples must be varied to ensure optimal model performance.

For this research, I performed a grid search whereby I initially considered the number of estimators in the range `[50,150]` and the percentage of samples in the range `[0.1,0.7]`. I then did a more detailed search around the top performing parameterization. `Figure 7` illustrates the results of this secondary search. While all of the models are comparable in terms of performance, the extreme values of the number of estimators tend to perform worse. Too few models results in underfitting, while too many leads to overfitting. Additionally, there is a trend in model improvement with increasing sample percentage, which peaks around 8% before tending to decrease. Similar to the number of estimators, too few samples leads to underfitting, while too many leads to overfitting. Capturing the rise and decline in model performance for these parameters ensures that at least a local maxima has been found in the parameter space. Ultimately, the best parameterization discovered by this search was a model with 40 estimators, each using 8% of the samples. This model achieved a cross-validated score of 73.6%.

3.4 Neural Networks

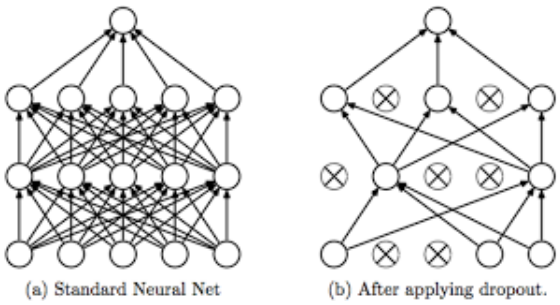

Figure 8: Illustration of dropout in dense neural networks.

Very generally, neural networks are a collection of highly interconnected nodes organized in layers, which process data through state responses generated from external inputs. They are most useful in recognizing patterns too complex to be formulated by people. The general structure contains an input layer consisting of the individual features, a collection of hidden layers of varying sizes, and an output layer consisting of the possible classification categories. In a densely connected neural network, each node of a particular layer is connected to each node of the next layer. These nodes each add up a linear combination of the inputs and a bias before passing the result through an activation function to determine its value to be passed to the next layer. The learning of the model occurs by determining the values of the weights through a process called backpropagation.

Backpropagation works by calculating the gradient of a cost function for a given sample. Therefore, the final results of the model are influenced by the step size used for gradient descent and the number of samples used for training. The architecture of the model also plays a significant role during final

prediction. Furthermore, neural networks are highly susceptible to overfitting when exposed to many passes of the same data or very deep[15] architectures. One approach to dealing with overfitting is to institute a dropout percentage, which is comparable to feature percentages used in tree-based methods. When dropout is used, a specified percentage of nodes at each layer are left out during an iteration. This ensures the model is not overly reliant on particular nodes. `Figure 8` illustrates this process. Parameter tuning neural networks is an art and requires a great deal of trial and error.

Given the computational cost, size of the parameter space, and time allotted for this project, only a few architectures and parameterizations were considered. Ultimately, the best performing neural network according to cross-validation score contained 5 hidden layers[16] and had a learning rate 0.001 and 50 epochs. This model, however, performed worse than even the baseline models, achieving a score of 72.7% overall. That said, there is plenty of room for further optimization and the performance of the gradient-boosted decision trees is well within reach.

4 Results

Table 4: Summary of results

Model	Parameterization	C-V Score (%)
Decision Tree	$Depth$ = 3	71.8
SVM	C = 0.1, $kernel$ = Linear	72.9
Random Forest	n_est = 60, $\%\_sam$ = 0.06, $\%\_feat$ = 0.12	73.2
Gradient-Boosted DT	n_est = 40, $\%\_sam$ = 0.08	73.6
Densely Connected NN	$layers$ = [100,50,25,10,10], l_rate = 0.001, $epochs$ = 50	72.7

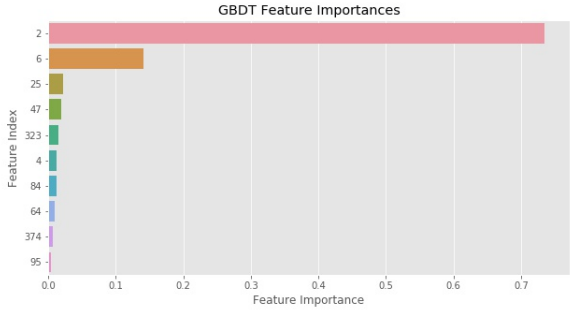

Figure 9: Feature importances of gradient-boosted decision trees.

`Table 1` shows the models with their optimal parameterizations and cross-validation scores. They were all developed on a balanced training set of 7200 samples split into 5-folds. Ultimately, gradient-boosted decision trees performed best, though there remains plenty of room for optimization with the densely connected neural network or other architectures.

The performance gains in the gradient-boosted decision trees comes at a slight cost in computation time and ease of interpretability. That said, feature importances can be used to estimate which features played the largest roles in the model's predictive ability. `Figure 2` illustrates the top 10 most important features. The feature that plays the largest role in prediction is age, which is a logical result as the risk of cancer greatly rises as a patient gets older. The second most important was a binary variable indicating whether the patient was a Non-Hispanic Black individual or not. This is an interesting result as the SEER Cancer Statistical Review found that Non-Hispanic Black individuals had the third highest rates of cancer behind Non-Hispanic White and Hispanic individuals. This is likely a spurious result stemming from small sample sizes within each class.

Finally, to get a sense of how the model would function under a real-world scenario, the model was tested on a held out set that was imbalanced in the target class. The overall accuracy came out

[15] Context dependent.

[16] Layer sizes of [100,50,25,10,10] respectively.

to 76.5%, with a precision of 69.2% and recall of 24.2%. When the same model is trained on an imbalanced set and retested, the overall accuracy comes out to 90.4%, with a precision of 0% and recall of N/A. While developing on the balanced set results in a loss in accuracy, both precision and recall improve. This is more applicable in a real-world scenario, where doctors are likely to prefer sacrificing pure accuracy for the sake of more accurately identifying at-risk patients.

5 Conclusion

Cancer prevention and risk prediction are among the foremost problems facing medical researchers today. Solving these problems provides the opportunity to save lives and healthcare costs of those who are susceptible. While traditional studies continue to persist, machine learning practitioners are leveraging the massive amounts of patient data in an attempt to discover underlying patterns and new avenues of research.

In this paper, I presented such an approach, whereby a multitude of patient characteristics were used to predict whether or not that patient would be exposed to cancer in their lifetime. Data was pulled from the NHANES 2015-2016 database and pre-processed according to self-defined algorithms. The primary source for further optimization is contained in this step. A greater selection of features combined with further personalized pre-processing schemas is the key to ensuring the opportunities of the data are maximized. This includes further rigorous testing of feature filtering and elimination techniques. A collection of models were then tested and optimized for accuracy on a balanced training set. The gains made in the optimization process of these models is marginal when compared with the impact of feature selection and engineering, but there exists many more binary classification models that could be applied to this analysis. A rigorous optimization of each would be required to ensure overall model completion.

Overall, there remains plenty of room for improvement with this model and I plan to continue researching going forward.

6 Appendix

6.1 Pre-processing Functions

```
def preproc_onehot(df_col, args=None):
    if (df_col.value_counts().iloc[0] / df_col.value_counts().sum()) > 0.85:
        df_col[pd.isna(df_col)] = df_col.value_counts().index[0]
    df = pd.get_dummies(df_col, prefix=df_col.name, prefix_sep='#')
    return df

def preproc_cont(df_col, args=None):
    # control for outliers
    lower = df_col.quantile(.01)
    df_col[df_col < lower] = lower
    upper = df_col.quantile(.99)
    df_col[df_col > upper] = upper
    # descriptive stats
    df_col_mean = df_col.mean()
    df_col_median = df_col.median()
    df_col_std = df_col.std()
    # Fill na
    imputed = pd.Series(np.zeros(len(df_col)),index=df_col.index)
    imputed[pd.isna(df_col)] = 1
    df_col[pd.isna(df_col)] = df_col_median
    # Normalize
    df_col = (df_col - df_col_mean) / df_col_std
    # Concat
    df_col = pd.concat([df_col,imputed],axis=1)
    return df_col
```

Figure 10: Proprietary functions used for pre-processing.

References

[1] Howlader N, Noone AM, Krapcho M, Miller D, Brest A, Yu M, Ruhl J, Tatalovich Z, Mariotto A, Lewis DR, Chen HS, Feuer EJ, Cronin KA (eds). SEER Cancer Statistics Review, 1975-2016, National Cancer Institute. Bethesda, MD, https://seer.cancer.gov/csr/1975_2016/, based on November 2018 SEER data submission, posted to the SEER web site, April 2019.

[2] B. C. Ross "Mutual Information between Discrete and Continuous Data Sets". PLoS ONE 9(2), 2014.

[3] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[4] Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Joel Ostblom, Saulius Lukauskas, ... Adel Qalieh. (2018, July 16). mwaskom/seaborn: v0.9.0 (July 2018) (Version v0.9.0). Zenodo. http://doi.org/10.5281/zenodo.1313201

[5] Dietert, R. R. (2016). The human superorganism: How the microbiome is revolutionizing the pursuit of a healthy life. NY, NY: Dutton.

[6] Hastie, T., Tibshirani, R., & Friedman, J. H. (2004). The elements of statistical learning: Data mining, inference, and prediction: With 200 full-color illustrations. New York: Springer.

[7] NHANES - National Health and Nutrition Examination Survey Homepage. (n.d.). Retrieved from https://www.cdc.gov/nchs/nhanes/index.htm

[8] Research On Cancer | Cancer Researcher. (n.d.). Retrieved from https://www.cancer.org/research.html

[9] Yu, H., Huang, F., & Lin, C. (2010). Dual coordinate descent methods for logistic regression and maximum entropy models. Machine Learning, 85(1-2), 41-75. doi:10.1007/s10994-010-5221-8

[10] Cancer. (2019, February 05). Retrieved from https://www.who.int/cancer/en/