Analysis of Machine Learning Models on Cancer Classification

Gray Selby, Alex Angus

October 29, 2019

Introduction

The goal of this project is to use mircroRNA (miRNA) profiles of cancer patients, provided by The Cancer Genome Atlas (TCGA) repository [1], to accurately predict the type of cancer that a patient has. A single patient's miRNA profile is comprised of over 1800 individual miRNA expressions. Their correlations to specific cancer types are complicated, and likely codependent. This complexity is where machine learning algorithms, such as the ones we demonstrate in this analysis, are necessary. The TCGA dataset represents six kinds of cancers to classify: breast invasive carcinoma, kidney renal clear cell carcinoma, lung adenocarcinoma, lung squamous cell carcinoma, pancreatic adenocarcinoma, and uveal melanoma. In our analysis, we implement five ordinary classification models with the python **scikit-learn** package: **random forest**, **decision tree**, **support vector**, **multinomial logistic regression**, and **k-nearest neighbors**. We also implement a sixth **stacked ensemble classifier** with the **mxlearn** package.

Data Exploration

Initially, the TCGA dataset is not aggregated into two discrete sets of example's feature data and their respective labels, as is needed for training machine learning algorithms. The data is initially categorized by labels (cancer types), with files containing examples in each. In each example file is another file with feature data for that example. To ensure that the data is ready to be used for machine learning, we explore the dataset with the **explore_data()** function in the **organize.py** file. This function iterates through the examples of each cancer type to ensure that they have a uniform feature profile. Also in **organize.py**, we have written a function **combine_data()** that compiles and stores these data files into a 2-dimensional feature array and a 1-dimensional label array that can be accessed with the **get_data()** function[1].

Feature Selection

In the feature selection process, we remove features that, when considered, do not improve, or hinder the classification accuracy of our models. In this case, a feature corresponds to the presence of a specific miRNA in a patient's profile. Below we explain our method of feature selection for each model.

Random Forest and Decision Tree

To perform feature selection on the Random Forest (RF) and Decision Tree (DT) models, we use the attribute **feature_importances_** of the **RandomForestClassifier** and **DecisionTreeClassifier** classes to determine the importance of a feature to the model. Prior to preforming feature selection, using all 1881 features and the default parameters of the previously mentioned sklearn classes, the RF and DT models have accuracies of 97.6% and 76.5%, respectively. Figure 1 shows a plot of the ratio of feature importance versus feature for the Random Forest and Decision Tree models. In our feature importance analysis, we choose an importance threshold of $5 \cdot 10^{-5}$, which corresponds to 346 features of the original 1881 in the Decision Tree Model. With an importance threshold of $5 \cdot 10^{-4}$, feature selection for the Decision Tree model results in a new set of 76 features of the original 1881. Our reevaluation of the Random Forest model with the reduced set of features has an accuracy of 97.4%, a 0.17% decrease

[1] Code for our feature selection process and hyperparameter search is found in the Jupyter Notebook 'Cancer Classification.ipynb'

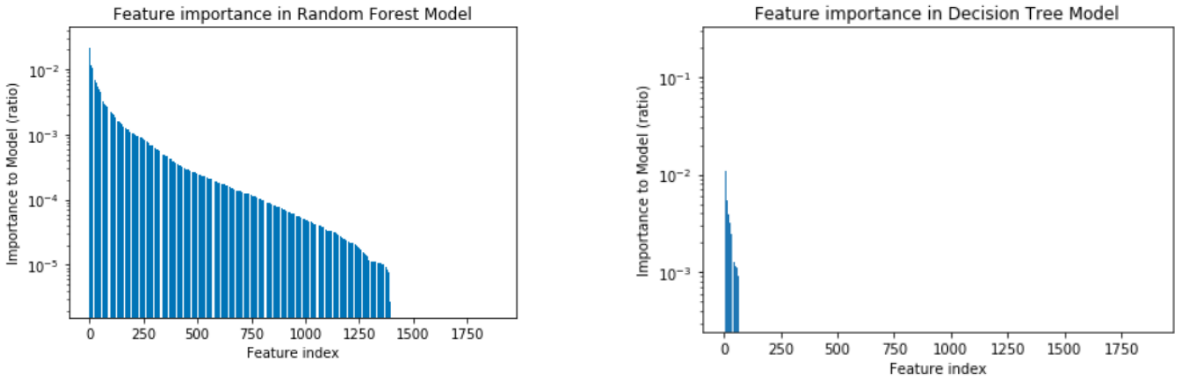

Figure 1: Feature importance of Random Forest (left) and Decision Tree (right) models. Note that features with zero importance are not used to make a 'decision' that decreases the entropy of the dataset. We see that the Decision Tree model has many more features with zero importance than the Random Forest model because the Random Forest is comprised of many trees, while the Decision Tree is a single tree.

from before feature selection. Feature selection produces little to no change in accuracy in the Random Forest model. The Decision Tree model, with the greatly reduced set of features, has an accuracy of 80.8%, a 4.3% increase from before feature selection. Feature selection therefore produces a moderate increase in accuracy in the Decision Tree model.

SVC and Logistic Regression

To preform feature selection for the Support Vector Classification (SVC) and Logistic Regression (LR) models we use the sklearn class **SelectFromModel**. The primary difference from our feature selection method for the Random Forest and Decision Tree models is that for SVC and LR we allow sklearn to determine the thresholds for eliminating features. Without feature selection (1881 features) and default sklearn hyperparameters, the SVC accuracy is 96.2% while the LR model accuracy is 96.7%. Feature selection for the SVC model resulted in a reduced set of 869 features, while feature selection for the LR model resulted in a reduced set of 826 of the initial 1881 features. Reevaluating the SVC model with the reduced set of features and default hyperparameters results in an accuracy of 97.1%, a 0.86% increase as compared to the same model without feature selection. Reevaluating the LR model with the reduced set of features and default hyperparameters results in an accuracy of 97.2%, an increase of 0.52% from before preforming feature selection. As with the SVC model, we see a slight increase in accuracy from feature selection for the LR model.

kNN

The k-Nearest Neighbors classifier differs from the other considered models in that there is no training portion of the algorithm. Therefore, in order to have a feature importance metric for the kNN classifier, we would need to create models that omit one feature at a time, and compare the performance of these individual models. For this dataset, that would mean implementing 1881 models, which would take entirely too long, as kNN is already a computationally intense algorithm. The k-Nearest Neighbors model with all 1881 features and the default sklearn hyperparameters predicts with an accuracy of 86.5%.

Hyperparameter Tuning

We preform hyperparameter tuning by running a random hyperparameter search with scikitlearn's **RandomizedSearchCV** class and running a grid hyperparameter search around the best hyperparameters found in the random search with scikitlearn's **GridSearchCV**. Each model (except in the case of k-nearest neighbors, where the concept

of over-fitting does not apply) includes hyperparameters that penalize over-fitting such as **max_depth** in the RF and DT models. By preforming hyperparameter searches with cross validation, the optimal over-fitting penalizing hyperparameters are found. With these hyperparameters, we ensure that we minimize bias and preserve model accuracy from dataset to dataset.

A summary of our individual model random and grid hyperparameter searches is below:

Random Forest Classifier	
Random Hyperparameter Search	Grid Hyperparameter Search
n estimators : 146 min samples split : 2 min samples leaf : 2 max features: auto max depth : 38 bootstrap : False	n estimators : 145 min samples split : 2 min samples leaf : 2 max features: auto max depth : 36 bootstrap : False
Best accuracy : 96.7%	Best accuracy : 96.0%

Decision Tree Classifier	
Random Hyperparameter Search	Grid Hyperparameter Search
splitter : best presort : False min samples split : 3 min samples leaf : 2 max leaf nodes : 500 max features : sqrt max depth : 130	splitter : best presort : False min samples split : 3 min samples leaf : 2 max leaf nodes : 495 max features : sqrt max depth : 133
Best accuracy : 82.0%	Best accuracy : 82.2%

Support Vector Classifier	
Random Hyperparameter Search	Grid Hyperparameter Search
kernel : linear gamma : auto C : 0.01	kernel : linear gamma : auto C : 0.005
Best accuracy : 97.2%	Best accuracy : 97.4%

Multinomial Logistic Regression Classifier	
Random Hyperparameter Search	Grid Hyperparameter Search
multi class : ovr max iter : 9400 fit intercept C : 0.252	multi class : ovr max iter : 9000 fit intercept : True C : 0.244
Best accuracy : 96.9%	Best accuracy : 96.9%

k-Nearest Neighbors Classifier	
Random Hyperparameter Search	Grid Hyperparameter Search
weights : distance n neighbors : 7 leaf size : 35	weights : distance n neighbors : 7 leaf size : 32
Best accuracy : 84.1%	Best accuracy : 85.8%

Model Stacking

We implement a sixth model with the ensemble method of stacking. With model stacking, each of our individual models is trained independently, and their predictions are then used as features to train a meta-classifier. The meta-classifier then makes the final prediction. We implement two stacked classifiers: one comprised of all our previous

models, and one with the top three performers. In both models, we find the Multinomial Logistic Regression Classifier to perform best as our meta-classifier. The results in the table below show that the two stacked models perform almost identically, correctly classifying 96.9% of the time. These are strong models, but the Support Vector Classifier still performs better with an accuracy of 97.4%.

Stacked Classifier	
Five stack classifier composed of all five previous models	Three stack classifier composed of the RF, SVC, and LR models
Accuracy of five model stack : 96.9%	Accuracy of three model stack : 96.9%

Conclusions

Through a scoring metric based on precision, recall, and f-1 score, we determine that our **Random Forest Classifier model preforms the best**, with all three of these values peaking at 98%. In [2], a Support Vector Classifier is trained that consistently achieves an accuracy greater than 90%. They experiment with feature selection and find that even reducing the number of considered features to only include the top 10% most important features yields similar results. Our results are comparable to [2], and it can be said that their results and ours are in agreement. Our results are also in agreement with work that precedes [2], such as [3], which implements similar classification methods. Figures 2 through 6 show confusion matrices and classification reports for each of our final models.

The random forest classification report shows perfect classification of breast invasive carcinoma, kidney renal clear cell carcinoma, pancreatic adenocarcinoma, and uveal melanoma, while the model classifies lung adenocarcinoma and lung squamous cell carcinoma with over 90% accuracy. The logistic regression classification report shows perfect classification of breast invasive carcinoma, kidney renal clear cell carcinoma, and uveal melanoma with over 95% accuracy for lung adenocarcinoma, lung squamous cell carcinoma, and pancreatic adenocarcinoma.

All of our models classify uveal melanoma with 100% accuracy. For kidney renal clear cell carcinoma our Random Forest, Decision Tree, and Logistic Regression models classify with 100% accuracy, while our k-Nearest Neighbor model performs with 99% accuracy. For all models, lung adenocarcinoma is classified with the worst accuracy when compared with other cancer types.

Given how different models are better classifiers of different cancer types, our stacked classifier seems to be a reliable classifier boasting 96.9% accuracy even when including the less reliable k-Nearest Neighbor classifier. However, the Random Forest Classifier alone produces higher accuracy.

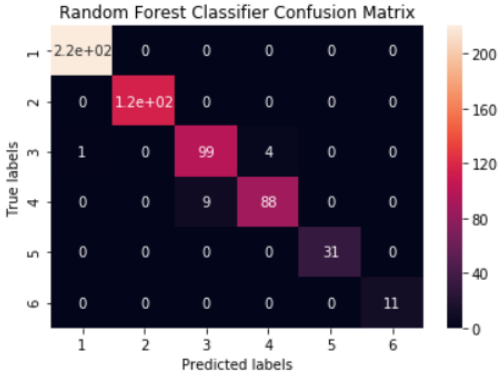

Figure 2: Confusion matrix of our final random forest classifier. In the confusion matrix, the numbered labels represent the order that the cancer types appear in the random forest classification report.

Random Forest Classification Report			
Cancer Type	Precision	Recall	f1-score
Breast Invasive Carcinoma	1.00	1.00	1.00
Kidney Renal Clear Cell Carcinoma	1.00	1.00	1.00
Lung Adenocarcinoma	0.92	0.95	0.93
Lung Squamous Cell Carcinoma	0.96	0.91	0.93
Pancreatic Adenocarcinoma	1.00	1.00	1.00
Uveal Melanoma	1.00	1.00	1.00
avg/total	0.98	0.98	0.98

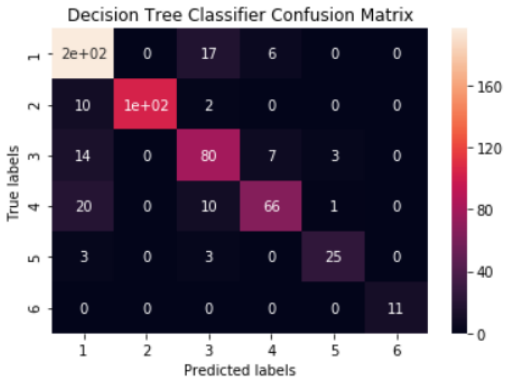

Figure 3: Confusion matrix of our final DT classifier. In the confusion matrix, the numbered labels represent the order that the cancer types appear in the DT classification report.

Decision Tree Classification Report			
Cancer Type	Precision	Recall	f1-score
Breast Invasive Carcinoma	0.81	0.90	0.85
Kidney Renal Clear Cell Carcinoma	1.00	0.90	0.95
Lung Adenocarcinoma	0.71	0.77	0.74
Lung Squamous Cell Carcinoma	0.84	0.68	0.75
Pancreatic Adenocarcinoma	0.86	0.81	0.83
Uveal Melanoma	1.00	1.00	1.00
avg/total	0.84	0.83	0.83

Support Vector Classification Report			
Cancer Type	Precision	Recall	f1-score
Breast Invasive Carcinoma	0.98	0.99	0.98
Kidney Renal Clear Cell Carcinoma	1.00	0.99	1.00
Lung Adenocarcinoma	0.91	0.96	0.93
Lung Squamous Cell Carcinoma	0.96	0.89	0.92
Pancreatic Adenocarcinoma	0.96	0.97	0.94
Uveal Melanoma	1.00	1.00	1.00
avg/total	0.97	0.97	0.97

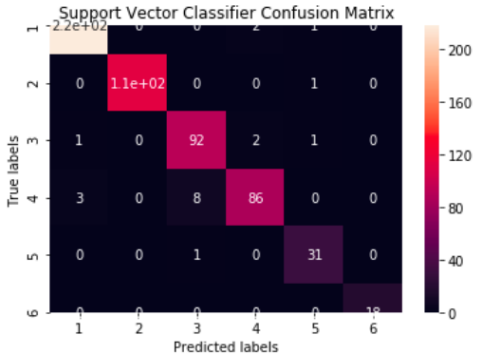

Figure 4: Confusion matrix of our final SVC classifier. In the confusion matrix, the numbered labels represent the order that the cancer types appear in the SVC classification report.

Figure 5: Confusion matrix of our final LR classifier. In the confusion matrix, the numbered labels represent the order that the cancer types appear in the LR classification report.

Multinomial Logistic Regression Classification Report			
Cancer Type	Precision	Recall	f1-score
Breast Invasive Carcinoma	0.99	1.00	1.00
Kidney Renal Clear Cell Carcinoma	1.00	0.99	1.00
Lung Adenocarcinoma	0.92	0.95	0.93
Lung Squamous Cell Carcinoma	0.96	0.91	0.93
Pancreatic Adenocarcinoma	1.00	1.00	1.00
Uveal Melanoma	1.00	1.00	1.00
avg/total	0.98	0.98	0.98

k-Nearest Neighbor Classification Report			
Cancer Type	Precision	Recall	f1-score
Breast Invasive Carcinoma	0.77	0.94	0.85
Kidney Renal Clear Cell Carcinoma	0.99	0.93	0.96
Lung Adenocarcinoma	0.83	0.70	0.76
Lung Squamous Cell Carcinoma	0.88	0.56	0.68
Pancreatic Adenocarcinoma	0.70	0.95	0.81
Uveal Melanoma	1.00	0.94	0.97
avg/total	0.85	0.83	0.83

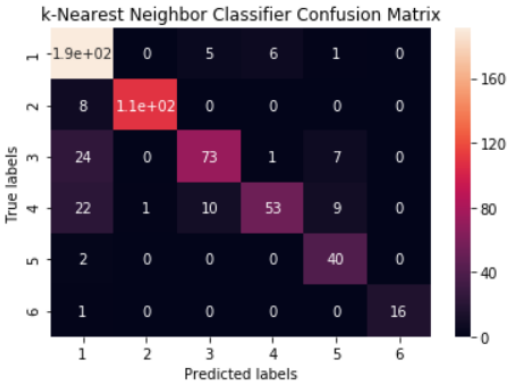

Figure 6: Confusion matrix of our final k-nearest neighbor classifier. Label 1 represents breast invasive carcinoma, 2 represents kidney renal clear cell carcinoma, 3 represents lung adenocarcinoma, 4 represents lung squamous cell carcinoma, 5 represents pancreatic adenocarcinoma, and 6 represents uveal melanoma.

References

[1] GDC. (n.d.). Retrieved October 29, 2019, from https://portal.gdc.cancer.gov/.

[2] Telonis, A. G. (2017). Knowledge about the presence or absence of miRNA isoforms (isomiRs) can successfully discriminate amongst 32 TCGA cancer types. Neucleic Acids Research. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5389567/

[3] Volinia, S. (2006). A microRNA expression signature of human solid tumors defines cancer gene targets. Proc Natl Acad Sci U S A. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1413718/