*Prediction of Pancreatic Conditions Using Classification & Association Algorithms in*

*Python*

# Ryan Handley

*Abstract*:

The dataset selected for this paper aims to predict pancreatic conditions and pancreatic cancer before a patient is diagnosed with a disease. Based on a paper that sought to develop an accurate way of diagnosing PDAC – pancreatic ductal adenocarcinoma, the most common form of pancreas cancer – through analyzing samples of urine from a collection of biomarkers[[1]](#footnote-1). Using the data collected by the team that authored said paper, this analysis aims to validate the results using common data mining techniques. I used classification algorithms (Decision Tree, Random Forest, Naïve-Bayes) to make predictions about a diagnosis. The algorithms are compared using the receiver operating characteristic curve (ROC) scores generated for both the classification and clustering algorithms and different scoring metrics for the clustering algorithm. Association rule mining – specifically, the Apriori algorithm – is also used in this analysis to determine the most frequent combinations of biomarkers and patient statistics leading to a specific diagnosis. The results indicate that association rule mining may not be suitable, despite generating some interesting findings, and that classification algorithms predict the diagnosis more effectively.

*Related Work*:

The paper the dataset is derived from discusses how PDAC is one of the deadliest cancers diagnosable, with a staggering 9% patient survival rate over five years[[2]](#footnote-2). It is often diagnosed too late due to its tendency to remain asymptomatic in its early stages. The authors discuss the promise of a specific biomarker panel (LYVE1, REG1A, and TFF1) they had previously touched upon and sought to improve upon it through the introduction of biomarker REG1B, of checking its complementarity with CA19-9 plasma in the bloodstream, a standard marker used for diagnosis of PDAC, creatinine content in the urine sample, and the use of an algorithm to interpret the data known as PankRISK. Their paper found that REG1B was a better indicator of risk than REG1A. The algorithm they developed here takes a step towards a more precise way to surveil potential PDAC patients; however, they are still taking steps to evaluate further biomarkers associated with the PancRisk score.

The performance metrics of their algorithm for PancRisk were evaluated using the ROC curve in R. The analysis performed here also uses the ROC score as the primary metric for determining the accuracy of our model. Instead of using the PancRisk algorithm devised by the authors of this paper, the analysis performed here tests the accuracy using preexisting classification algorithms in Python. Their study replaces REG1A with REG1B, but for our purposes, we will seek a broader understanding of their roles in diagnosing PDAC. The apriori algorithm is intended to inform us on how different levels of the biomarkers in question impact the patient’s diagnosis. To determine what algorithms would be most effective in the context of disease diagnosis, I scoured for other papers on disease classification. One article gives an overview of the most common techniques in machine learning to attempt to classify diseases. Classification techniques like the decision tree algorithm, K-nearest neighbor, naïve Bayes, logistic regression, and support vector machine are among the most used algorithms in disease diagnosis and ensemble methods like random forest[[3]](#footnote-3). Using a handful of these algorithms to compare results on this dataset and comparing their results should show us which classification model is the most effective at predicting a diagnosis.

*Data*:

The data set for this analysis contained 590 records for standard patient demographics, such as their age, sex, and the cohort to which they belonged. It also included the biomarkers (creatinine, LYVE1, REG1A, REG1B), blood plasma levels of CA19-9, the origin of the sample, the diagnosis, (if cancerous) the stage of the disease, and (if benign) the disease diagnosis. Like most data sets, there was also a sample ID. Luckily, this data was already relatively clean except for the sample ID, some null values, and two irrelevant attributes. For the REG1A levels, only 306 patients had this assessed, so I imputed an average to the patients who did not have it evaluated. The cancer stage only applied to a diagnosis code of three, and the benign sample diagnosis only applied to diagnosis code two, so they would not necessarily contribute to the model predicting the diagnosis. The sample ID does not contribute to the model, so I dropped these columns from the data frame.

After removing any columns that had no significant impact on the model, I checked for any outliers in the dataset and removed them using Scikit-Learn’s Isolation Forest method. The algorithm isolates observations in the data with recursive partitioning to detect and remove anomalies[[4]](#footnote-4). We removed anomalies within a contamination level of 15%, finding that anything beyond this did not produce a significant difference in the new range of data. Anything beyond the range of 10% - 20% contamination alters the data drastically in that fewer diagnosis codes for cancer appeared in the analysis, which initially indicates that higher levels of the biomarkers in a urine sample have some correlation to the diagnosis.

From there, I discretized the levels of the biomarkers and plasma by creating bins within their respective ranges and dropped the original columns. I also made age groups from the patients’ ages a new feature in the data set to help identify those at higher risk of a diagnosis and dropped the age column. I then encoded the age group and sex columns into binary values for a more effective classification. At this point, I was ready to perform my classification algorithms.

After performing these algorithms and comparing the results, I further prepared the data for an association rule analysis by encoding the diagnosis codes to generate rules about them more accurately. From there, I performed the Apriori algorithm to develop the maximum itemset rules for each diagnosis.

*Methodology*:

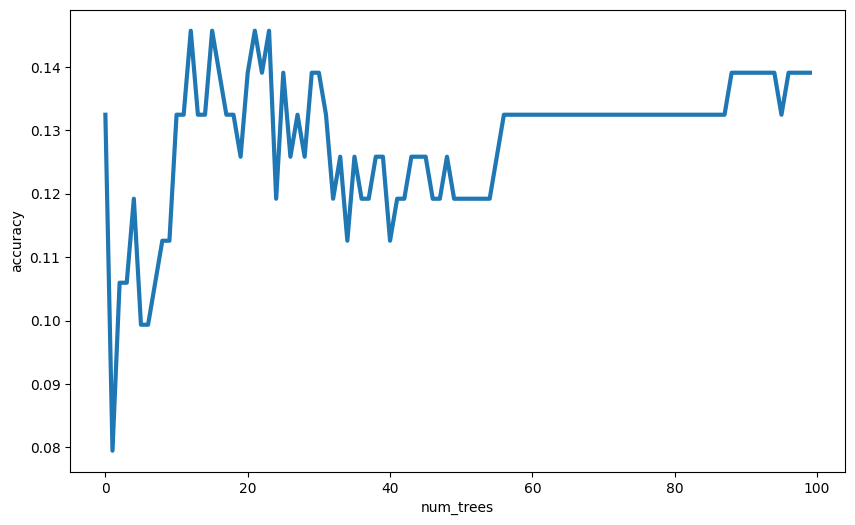
The first approach to analyzing the data set involved testing the efficiency of different classification and clustering algorithms and comparing their ROC-AUC scores to see how well each predicts the diagnosis. The receiver operating characteristic curve (ROC Curve) is a graph that ranks the performance of classification models on a curve plotting the true positive and false positive rates. The area under the ROC curve (AUC) measures the two-dimensional space under the curve from (0,0) – (1,1), where the score ranges from 0 to 1. A zero value would indicate that 100% of the predictions were incorrect, and a value of 1 indicates 1that 00% of the predictions are correct[[5]](#footnote-5).

The Scikit Learns library in Python was used to run the classification models selected to split the data set into training and testing sets at a 70/30 ratio using train\_test\_split. This method allows us to split the dataset into random training and testing subsets for classification models[[6]](#footnote-6). The attribute selected as our class attribute was the patient diagnosis. Since we are trying to predict the diagnosis, we need to target this attribute in the classification algorithm with this approach.

To run the Decision Tree algorithm, the Scikit-Learn library also allows users to create decision trees, so I made two trees, each measured with a different performance metric. For the first, we used entropy as our performance metric to derive the information gain; for the second, we used the Gini index to measure the unevenness of the tree. Both were ranked by their AUC scores, and the trees were plotted using plot\_tree from sklearn[[7]](#footnote-7). The scores of the trees were .7881 and .7877 with respect to the entropy and gini metrics.

Decision Tree measured by Entropy
 Decision Tree measured by Gini Index


I used Scikit Learn’s Random Forest Classifier to run the RLearn’sorest ensemble algorithm. Ahmed Abulkhair devised a method to determine the best number of n-estimators to run the algorithm. I employed it to determine the most suitable parameters for the data set[[8]](#footnote-8). As we can see using an initial n-estimator of 100, there is a sharp increase in the accuracy of the model with anywhere up to roughly 15 trees, with a stabilization around 55 trees. I ran the analysis with 55 trees and found that the ROC score of 0.822 was superior to both variations of the decision tree created. A confusion matrix and classification report were generated for the data, showing an average accuracy of 65%. Using this model, we ranked the importance of the features in the dataset to see if we found anything interesting.



Using Scikit Learn’s Naives-Bayes methods, I analyzed the split data set using the MultinomialNB approach to account for the binning of my data set[[9]](#footnote-9). A confusion matrix and classification report were generated for the data, showing an average accuracy of 61% and a ROC score of .784.

I also executed the Apriori algorithm to determine the top ten rulesets that could be generated with a diagnosis as the consequent. I measured the importance of the rules by their support, confidence, and lift scores. The lift scores were the most notable scores to examine as they indicated the strength of the correlation between antecedents and consequents. The rules for a healthy diagnosis were typically females from the second patient cohort, with the urine samples originating in England. For patients with non-cancerous pancreatic conditions, there was a weaker correlation to women being diagnosed with some disease. For cancerous diagnoses, most came from the first patient cohort.

*Results*:

After analyzing the ROC-AUC scores of all the classification and ensemble methods performed on the data, the random forest classification was determined as the best model for predicting the disease, with a ROC-AUC score ranging from .82 to .84. The second-best model was the Naive Bayes classification, with a ROC-AUC score ranging from .784 to .813. The decision tree classifier was the least accurate of these three approaches, with some variation depending on the criterion selected. In terms of accuracy, random forest also produced the highest accuracy score of the classifications used at around 72%. The classification of pancreatic cancer is difficult enough, given it is hard to detect by nature. The biomarkers being analyzed show some promise in making predictions about a diagnosis but could benefit from a more extensive data set. The use of association rule mining gives us some, but very limited, insight in terms of predicting a diagnosis. The rules generated have lift scores indicating a strong positive correlation between the biomarkers found in a sample but lack support and confidence levels that are statistically significant enough to accurately predict a diagnosis without more data.

# Works Cited

* Davis, J. (2020, December 10). *Urinary biomarkers for pancreatic cancer*. Kaggle. <https://www.kaggle.com/datasets/johnjdavisiv/urinary-biomarkers-for-pancreatic-cancer>
* Debernardi, S., O’Brien, H., Algahmdi, A. S., Malats, N., Stewart, G. D., Plješa-Ercegovac, M., Costello, E., Greenhalf, W., Saad, A., Roberts, R., Ney, A., Pereira, S. P., Kocher, H. M., Duffy, S., Blyuss, O., & Crnogorac-Jurcevic, T. (2020, December 10). *A combination of urinary biomarker panel and PANCRISK score for earlier detection of pancreatic cancer: A case–control study*. PLOS Medicine. <https://journals.plos.org/plosmedicine/article?id=10.1371%2Fjournal.pmed.1003489>
* Ahsan, M. M., Luna, S. A., & Siddique, Z. (2022, March 15). *Machine-learning-based disease diagnosis: A comprehensive review*. Healthcare (Basel, Switzerland). <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/#sec5-healthcare-10-00541>
* *Sklearn.ensemble.isolationforest*. scikit. (n.d.). <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html>
* Google. (2022, July 18). *Classification: Roc curve and AUC  |  machine learning  |  google for developers*. Google. <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=AUC%20stands%20for%20%22Area%20under,across%20all%20possible%20classification%20thresholds>.
* *Sklearn.model\_selection.train\_test\_split*. scikit. (n.d.-b). <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>
* *Sklearn.tree.plot\_tree*. scikit. (n.d.-c). <https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html>
* Abdulkhair, A. (2021, August 16). *Best n\_estimators for randomforest*. Kaggle. <https://www.kaggle.com/code/ahmedabdulhamid/best-n-estimators-for-randomforest>
* *Sklearn.naive\_bayes.multinomialnb*. scikit. (n.d.-c). <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>

1. Davis, J. (2020, December 10). *Urinary biomarkers for pancreatic cancer*. Kaggle. https://www.kaggle.com/datasets/johnjdavisiv/urinary-biomarkers-for-pancreatic-cancer [↑](#footnote-ref-1)
2. Debernardi, S., O’Brien, H., Algahmdi, A. S., Malats, N., Stewart, G. D., Plješa-Ercegovac, M., Costello, E., Greenhalf, W., Saad, A., Roberts, R., Ney, A., Pereira, S. P., Kocher, H. M., Duffy, S., Blyuss, O., & Crnogorac-Jurcevic, T. (2020, December 10). *A combination of urinary biomarker panel and PANCRISK score for earlier detection of pancreatic cancer: A case–control study*. PLOS Medicine. https://journals.plos.org/plosmedicine/article?id=10.1371%2Fjournal.pmed.1003489 [↑](#footnote-ref-2)
3. Ahsan, M. M., Luna, S. A., & Siddique, Z. (2022, March 15). *Machine-learning-based disease diagnosis: A comprehensive review*. Healthcare (Basel, Switzerland). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/#sec5-healthcare-10-00541 [↑](#footnote-ref-3)
4. *Sklearn.ensemble.isolationforest*. scikit. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html [↑](#footnote-ref-4)
5. Google. (2022, July 18). *Classification: Roc curve and AUC  |  machine learning  |  google for developers*. Google. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=AUC%20stands%20for%20%22Area%20under,across%20all%20possible%20classification%20thresholds. [↑](#footnote-ref-5)
6. *Sklearn.model\_selection.train\_test\_split*. scikit. (n.d.-b). https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.train\_test\_split.html [↑](#footnote-ref-6)
7. *Sklearn.tree.plot\_tree*. scikit. (n.d.-c). https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot\_tree.html [↑](#footnote-ref-7)
8. Abdulkhair, A. (2021, August 16). *Best n\_estimators for randomforest*. Kaggle. https://www.kaggle.com/code/ahmedabdulhamid/best-n-estimators-for-randomforest [↑](#footnote-ref-8)
9. *Sklearn.naive\_bayes.multinomialnb*. scikit. (n.d.-c). https://scikit-learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html [↑](#footnote-ref-9)