Portfolio Milestone

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Applied Data Science
Syracuse University, School of Information

Nadia Paquin

June 16 2024

Table of Contents

I. Program Learning Outcomes

The Applied Data Science Program at Syracuse University's School of Information features a wide range of courses that provide students with a comprehensive instruction of how data can be used to inform decision making. Courses in this program are well-rounded for the needs of a data scientist, drawing upon various areas of studies: computer science, statistics, social sciences, management, machine learning, research design, cybersecurity, and more. Ultimately, students are enabled to acquire, explore, analyze, manage, and visualize large data sets using the latest technologies to provide data-driven insights to any kind of problem.

The learning outcomes as described on the program's official website go as follows:

1. Collect, store, and access data by identifying and leveraging applicable technologies.
2. Create actionable insight across a range of contexts (e.g. societal, business, political), using data and the full data science life cycle.
3. Apply visualization and predictive models to help generate actionable insight.
4. Use programming languages such as R and Python to support the generation of actionable insight.
5. Communicate insights gained via visualization and analytics to a broad range of audiences (including project sponsors and technical team leads).
6. Apply ethics in the development, use and evaluation of data and predictive models (e.g., fairness, bias, transparency, privacy).

This document is a means of showcasing projects completed with the program outcomes in mind. The following sections summarize final projects that demonstrate proficiency in the program outcomes.

II. IST718 Big Data Analytics

Course Scope: Clean, manipulate, and analyze large datasets using Python and Apache Spark and apply analytics to real-world problems and communicate results effectively. Learn to obtain, screen, clean, link, manipulate, analyze and display data while creating summaries, overviews, models, analyses and basic tables, histograms, trees and scattergrams.

Project Goal: Differentiate between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) in patients using only their genetic expression data.

Collaborators: Vu Ton and Isabela Tuda

Data: DNA microarray was used to quantify the relative expression of over 7,000 genes from bone marrow and peripheral blood samples of 72 patients with AML and ALL. Intensity values were re-scaled such that overall intensities for each chip are equivalent. The data was published by Golub et al in a 1999 proof-of-concept paper on the subject. The publication included the initial (training, 38 samples) and independent (test, 34 samples) datasets.

Method: Using the training set, several models were built using python packages to classify the cancer types. Models include: support vector machines, K-means, logistic regression, Naive Bayes, and Random Forest. Models were tested against the training set provided and evaluated for accuracy.

Outcome: Most models yielded high accuracy when tested against the testing set except K-means. This is expected as K-means takes an unsupervised approach whereas the rest of the models are supervised. Logistic regression yielded the highest accuracy at around 97.06%. All models erred with a small handful of false positives only, aside from the Random Forest model which predicted false negatives. Based on the high accuracy of the models, we conclude that we are able to effectively differentiate between the two types of cancer using sequenced DNA with approximately 95% accuracy. Doctors can consider relying on bone marrow biopsy alone for diagnosis, eliminating the need for several assays in distinct, highly specialized labs. This finding, in the years of its original publication, opened the door to early detection and proper treatment for patients with leukemias. These results suggest that there is a relationship between gene expression and cancer pathologies. This raises the question that there perhaps might be a pathological relationship between specific genes and leukemia. More research should be done to determine which genes are most critical, the proteins and pathways they are associated with, and if these relationships are causal. Future research can also be done to see if these models work for other cancers.

Reflection: My project effectively demonstrates several core learning outcomes of this course and Syracuse University's Applied Data Science Program. I sourced and analyzed a large dataset (500,000 data points) and applied a range of machine learning models to classify cancer types based on genetic expression. This process involved leveraging technologies for data collection and management, utilizing Python programming skills for model implementation, and applying visualization techniques to interpret results. Achieving high accuracy in differentiating between AML and ALL underscores proficiency in applying machine learning techniques to real-world medical diagnosis. Communicating these findings effectively reflects the program's emphasis on conveying insights to inclusive audiences, showcasing comprehensive training in data-driven decision-making provided by the program.

III. IST664 Natural Language Processing

Course Scope: This course covers linguistic and computational aspects of natural language processing (NLP), teaching students to process and analyze text using Python and NLTK. Topics include tokenization, word-level semantics, part-of-speech tagging, syntax, discourse, sentiment analysis, and dialogue systems.

Project Goal: Write a Python script to detect spam emails from regular emails.

Data: The Enron public email corpus contains 1500 spam emails and 3672 non-spam “ham” emails to build a classifier to parse and detect spam.

Method: I used the NLTK package to pull in data, tokenize the corpora, build feature sets to train a Naive Bayes classifier, and cross-validate the results. I experimented with several different methods to build the classifier, including changing the vocabulary size, changing the corpus size, adding stop and negation words, POS tagging, using n-grams, and looking at the ratio of words to punctuation.

Given the input of 1000 corpora files (500 spam and 500 ham), I created several feature sets to train a Naive Bayes classifier. I ran the classifier with the following adjustments:

- I began with a feature set on the top 2000 most frequent words of the corpus input. Initially I did not choose to filter the data for punctuation or stop words as I assumed that the repetitive occurrence of either of those is sometimes indicative of spam emails. This classifier thus uses the most common occurrences of words to determine if a corpus is spam or ham.
- I adjusted the size of the feature set containing the most frequent words, using 1000, 500, and 100 instead of 2000.

- I varied the number of files used to collect the top words, using 250, 100, 25, and 5 from each corpus instead of 500 (maintaining the feature set at 2000).
- I removed stopwords and negation while maintaining feature set size and file count.
- I used NLTK's bigram finder to identify the top 500 bigrams and used this as a feature set instead of the most frequent words.
- I tested my theory that spam emails often throw in excessive punctuation and wrote a function to identify word punctuation ratio per piece of text and trained the classifier on that.
- I used part of speech tagging to do the same process.
- I removed stopwords from the text in addition to using bigrams instead of word frequencies to train the data.
- I tried using the word frequency instead of the boolean value to train and test the data.

Outcome: The mean accuracy on my first run using only the top 2000 most common words was 95.4% with high precision on spam and less precision on ham. This result was a spam filter that occasionally (8% of the time) sends your real email to your spam folder. One way this could be improved is to prioritize high ham precision and lower spam precision, so as to avoid tossing any important emails aside at the risk of having a few spam emails enter your inbox. The following changes were unable to increase the accuracy significantly above 95%. The decreased feature set size from the top 2000 most frequent words to 1000, 500, and 100 words yielded mean accuracies of 95.3%, 95.6%, and 95.4%, remaining steady despite the decrease, with varying levels of sensitivity and specificity. The decreased input file size from 500 each to 250, then 100, 25, and 5 each resulted in a decrease in overall accuracy, at 94.2%, 88.0%, 72.0%, and 40.0% respectively. After removing any stopwords and negation, accuracy remained at 95.0%. Using NLTK's bigram finder to identify the top 500 bigrams did not change the accuracy, yielding 95.0%. Word to punctuation ratio proved unrelated to status as spam, with an accuracy of around 52.9%. Parts of speech tagging yielded little to no change in the accuracy at 95.3%. Removing stop words in addition to training on bigrams also yielded 95.2%. Using word frequency instead of the boolean value was unable to bring the accuracy over 95%, at 94.5%.

Reflection: This project demonstrates proficiency in the Applied Data Science Program learning outcomes through its comprehensive approach to spam email detection. The project involved managing a substantial dataset, demonstrating the ability to handle large volumes of data effectively. By implementing a Naive Bayes classifier, various predictive models were tested using different feature sets, including frequent words, bigrams, and word-to-punctuation ratios. This demonstrates the application of data science techniques to develop effective tools for practical applications, such as improving email filtering accuracy. The use of Python and the NLTK package for data processing, feature set creation, model training, and validation highlights proficiency in using programming languages to build and evaluate machine learning models.

IV. IST707 Applied Machine Learning

Course Scope: This course explores industry-standard machine learning techniques and algorithms, emphasizing model development, optimization, and real-world applications. Students gain practical experience with modern data science tools to analyze data mining needs, apply algorithms effectively, and communicate insights through data storytelling. Key topics include machine learning methods, R and RStudio, data preparation, association rule mining, classification, clustering, and evaluation techniques.

Project Goal: Build a model to diagnose a patient with a disease from the presence of any combination of 133 symptoms.

Collaborators: Victoria Haley and Gustavo Gyotoku

Data: Our data is an artificial set sourced from Kaggle, with two CSV files, one training and one testing. Each table includes 133 columns, 132 of these columns are symptoms and the last column is a prognosis. These symptoms are mapped to 42 different diseases. Testing data includes 4290 patients and testing data includes 42 patients.

Method: We first cleaned and investigated the data for any anomalies or NA values. Once satisfied, we built a handful of models using the training data. Models included decision trees, Naive Bayes, Random Forest, and K-Nearest Neighbors (KNN), sourced from R libraries e1017, caret, randomForest, and rpart. We decided to first utilize the decision tree classifier to identify the variables that would have the most impact on the test dataset. The idea was that these symptoms would be useful in increasing the overall model accuracy. The “varImp” function from the “rpart” library reported the top most impactful variables by a numeric value. These variables were saved to an object which was then fed to the remaining classifiers Naive Bayes, Random Forest and KNN.

Outcome: The decision tree algorithm yielded a prediction accuracy of 74%. While this approach demonstrated promising results, it may be sensitive to data imbalance and prone to overfitting. Further fine-tuning could enhance its diagnostic capabilities. The Naive Bayes algorithm exhibited a considerably higher accuracy of 89%. This technique's ability to handle large and complex datasets, along with its assumption of feature independence, contributed to its success in predicting diagnoses. Employing the Random Forest algorithm resulted in a remarkable accuracy of 93%. Random Forest demonstrated significant potential in accurately classifying and predicting diagnoses, leveraging the power of an ensemble of decision trees. Among the machine learning techniques employed, KNN achieved the highest accuracy of 99%. The exceptional performance of KNN in diagnosing various conditions can be attributed to its ability to find the most similar instances in the dataset and make predictions based on local patterns. These results underscore the effectiveness of machine learning techniques in diagnosing medical conditions based on the provided dataset. Notably, KNN emerged as the most accurate method, closely followed by Random Forest and Naive Bayes. The Decision Tree approach, while demonstrating potential, may benefit from further refinement to improve its diagnostic accuracy. These findings highlight the value of leveraging machine learning algorithms in healthcare settings to enhance diagnostic timeliness and improve patient outcomes.

Reflection: This project demonstrates proficiency in the program learning outcomes by employing advanced machine learning techniques to diagnose diseases based on symptom data. Using R and libraries such as e1071, caret, randomForest, and rpart, we processed a dataset containing 133 symptoms mapped to 42 diseases and implemented decision trees, Naive Bayes, Random Forest, and KNN models to a high yield of accuracy. The findings were effectively communicated through figures and explained in a presentation, showcasing our ability to effectively communicate findings. This project not only highlights technical proficiency in model development and optimization but also underscores the program's emphasis on applying data science methodologies to practical domains.

V. IST659 Database Administration Concepts and Database Management

Course Scope: This course in database administration covers fundamental database management system models and their applications. Students learn data management techniques for relational databases, schema design, and SQL query languages using Microsoft Access and SQL Server.

Project Goal: Design and implement a functional inventory system with a database.

Data & Tables: Inspired by the organization of a lab-inventory software, I decided to create my own library-style system for personal use. The database includes several key tables supporting an inventory management system. The **shelves** table lists shelf names like 'avila', 'shell', and 'pismo', while the bins table details 25 bins labeled from 'A1' to 'E5', representing storage units within the shelves. The **actions** table defines transaction types such as 'new addition', 'checked out', and 'returned', and the categories table categorizes items into groups like 'tech equipment', 'surf accessories', 'art supplies', 'tools', and 'misc'. The **users** table lists usernames such as 'nadia', 'anton', 'davos', and 'dante', representing individuals who interact with the system. The **items** table records item details, including names, associated category IDs, shelf IDs, and bin IDs, with examples like 'surf wax', 'hdmi cable', and 'paintbrushes'. The **checkout_history** table logs transactions with data on transaction dates, item IDs, action IDs, and user IDs, documenting the addition, checkout, and return of items. These tables work together to track items, their locations, and transaction histories within the personal inventory system.

Views: all_things_and_homes displays item details with location, **checkout_history_recent** shows the latest checkout transactions, **all_things_history** offers a complete checkout history, and **items_inventory** combines item and checkout details. The views **items_inventory_cat_keys** and **items_inventory_item_keys** extend the inventory view with keyword search functionality based on category and item names, respectively.

Procedures & Functions: The procedures handle key inventory actions: **checkout_item** and **return_item** update the checkout history for items, ensuring correct handling of checked-out and returned items with error handling for invalid actions. **add_item** inserts new items into the inventory with specified details, and **remove_item** deletes items based on their name, ensuring data integrity. The functions **search_categories** and **search_items** provide keyword search capabilities, returning relevant inventory rows based on category and item name keywords.

Error Handling: Error handling is implemented with specific error codes: **Error 50100** for invalid checkout or return actions, **Error 50101** for user registration or spelling issues, and **Error 50102** for problems with adding or removing items.

Reflection: This project involved designing and implementing a functional inventory system that effectively manages data through various stages, including collection, storage, and access. Utilizing technologies like Microsoft Access and SQL Server, I created a tool to support robust data structures and efficient inventory management. Understanding the inner workings of such data management practices is invaluable for engaging with data in various data science roles across industries. The project highlighted my ability to apply ethical principles in data management, ensure data integrity, and effectively communicate insights derived from the system's operation.

VI. Conclusion

My four projects span a wide range of skills and applications within the field of data science. In IST718, I differentiate between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) using genetic expression data, demonstrating proficiency in large dataset analysis, predictive modeling, and biomedical informatics. IST664 focuses on spam email detection, showcasing expertise in text data processing, machine learning classification techniques in Python, and ethical considerations in data usage. IST707 involves diagnosing patients based on symptoms, highlighting my skills in machine learning model development, evaluation using R markdown, and application in healthcare analytics. IST659 Database Administration Concepts and Database Management features the creation of an inventory database, emphasizing my proficiency in SQL, database design, procedures, views, and ensuring data integrity. These projects collectively illustrate my ability to apply data science methodologies across diverse domains, preparing me for roles that require comprehensive data analysis and strategic decision-making skills.

VII. References

Golub, T R et al. “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.” Science (New York, N.Y.) vol. 286,5439 (1999): 531-7.

KAUSHIL268 (2020). Disease Prediction Using Machine Learning [Data set]. Kaggle. https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning/data

V. Metsis, I. Androutsopoulos and G. Paliouras, "Spam Filtering with Naive Bayes - Which Naive Bayes?". Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 2006.