DS-GA 1001 Term Project
Fall 2019
Team Members: Jonas Aranda, Jordan Chervin, Silas Mann, Lipsa Panda

Prognostic Modeling for Acute Kidney Injury in a Hospital Setting

Business Understanding:

Acute kidney injury (AKI) occurs when the kidneys are not able to filter waste products from the blood and excrete the waste as urine. As a result, the body accumulates waste, fluids, and electrolytes. Dangerous levels of these products require intensive treatment, and AKI's or their complications can be fatal. The global incidence of AKI in adults is almost 22%, with a mortality rate of 24%. Moreover, AKI is associated with 25 - 50% of ICU admissions (Malhotra et al. 2017).

Acute kidney injury has three primary causes: impaired blood flow to the kidneys, damage to the kidneys, or blockage in the kidneys. Risk factors include age, hypertension, diabetes, kidney and liver disease, some cancers, and hospitalization. Patients often present with constitutional symptoms such as fatigue, nausea, dyspnea (shortness of breath), and weakness, and may also present with decreased urine output, fluid retention causing swelling, and an irregular heartbeat. AKI can be staged or classified by serum creatinine levels and urine output levels. AKI's are confirmed via blood and urine tests, urine output measurements, ultrasounds or CT scans, and biopsies. Depending on the severity of the injury, treatments range from intravenous fluids to oral medications to dialysis.

Our business problem is to classify patients who are admitted to hospitals who are likely to be diagnosed with Acute Kidney Injury during their admission. We deliberately chose a condition that is not immediately obvious upon initial presentation, has a variety of risk factors and causes, and requires multiple diagnostic tests to confirm.

Supervised learning can help to address this diagnostic challenge by building a model based on data from previous patients who were diagnosed with AKI, and using this model to classify new patients as either having or not having the diagnosis of interest within a certain threshold of confidence**.** Such a model may reveal that a particular procedure, assessment, or patient characteristic may be more or less informative than previously thought. Deploying such a model into a clinician's practice can provide a faster delivery of diagnostic information, resulting in potentially earlier and less invasive intervention. Expediency of diagnostic

information allows clinicians to take appropriate measures to treat or mitigate the condition in a more timely fashion, before it worsens or further complications arise. Furthermore, this information could also aid in hospital resourcing (nurse and laboratory staffing, dialysis machines, medication supply, etc.).

Data Understanding:

We are utilizing the freely-available MIMIC-III (Medical Information Mart for Intensive Care) dataset (https://mimic.physionet.org), which is comprised of de-identified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. This rich dataset depicts one or more hospital admissions per patient, and is available as a relational database consisting of 26 tables with information such as clinical laboratory results, procedures, ICU stays, prescribed medications, and diagnoses.

Diagnoses in the MIMIC-III dataset are represented as ICD-9 codes, which are a set of alphanumeric codes established by the International Statistical Classification of Diseases and Related Health Problems. ICD-9 codes are used to classify symptoms, findings, diseases, complaints, among others. For diagnosis classification, ICD-9 codes are more granular than required to identify a single binary label for AKI diagnosis, so we merged the diagnoses with the ICD HCC (International Classification of Diseases to Hierarchical Condition Category) Crosswalk dataset. ICD-9 codes and HCC codes are not 1:1, so this dataset matches one or more ICD-9 codes with a single HCC code.

Data Preparation:

Our feature engineering resulted in a pandas Dataframe with 51,113 instances and 205 features. We define an instance to be a single hospital admission, each with a unique id (hadm_id). Our target variable is the HCC code for acute kidney injury, which is 135 (ft_hcc_cd_135). This HCC code corresponds to ICD-9 codes 5845-5849 (acute kidney failure with lesion of tubular necrosis, lesion of renal cortical necrosis, renal medullary necrosis, other specified pathological lesion, and unspecified). Each instance was assigned either a 0 or a 1 in this column, corresponding to a negative or positive diagnosis, respectively. The base rate of instances with the HCC code 135 is 22% (N = 11418).

We reviewed medical literature and researched other prognostic models for AKI in the hospital setting in order to identify a set of features that would be likely predictors. From this review, we narrowed down our feature extraction areas into the following categories:

- Laboratory values (potassium, sodium, hematocrit, creatinine, low blood pH)
- Demographics (gender, age, ethnicity)
- Comorbidities (anemia, hypotension, hypertension, chronic hepatitis, sepsis, HIV/AIDS)
- Concomitant medications (prescriptions that can be nephrotoxins)
- Medical history (prior admission within 30 - 120 days, prior medical or critical care unit ICU stay within 30 days)
- Use of mechanical ventilation

For each of these clinical conditions and cases, we looked at chart and lab data to create one-hot encoded dummies. We created binary variables for gender and ethnicity. For charts data, there are many measurements taken for each observation, so we had to be creative in how we encoded each variable into our final feature set. Therefore, where possible, we also considered time. We merged the admissions set back onto itself to identify readmissions and to encode prior admissions. As another example, we created a series of variables to represent whether a nephrotoxic drug was prescribed to a patient within 12, 24, 36, and 48 hours of their admission. We also applied reference ranges to our variables where appropriate. For instance, there are two features for high blood pressure ("ft_hbp_stg_1" and "ft_hbp_stg_2") to represent the different stages/severity of hypertension. Also, the average value of iron in the blood was not used to determine "anemia"; instead, a dummy variable was encoded for whether there was a lab result during the stay which had iron levels below a clinically relevant reference range.

After we created this raw feature set, we pre-processed the data to prepare it for modeling. Given that our dataset only had 22% positive instances, we **undersampled the negatives** in the data so that we had a more evenly distributed set for the classifier models. We then **split the data into training (70%), validation (20%),** and **test (10%) sets** before doing any additional pre-processing. We split prior to fitting any models so that we would not cross-contaminate the test data with the means from the training data that could potentially result in data leakage. After we split the data, we **replaced nulls** for numeric variables with means and nulls for binary variables with the most frequent value. We then **dropped** any

columns that were **all zeros** for parsimony. Finally, we **scaled** the data by subtracting the mean and dividing by the standard deviation for each column.

To cater our features to our goal of early detection, we focused on features that were either collected at hospital admission or would be relatively easy to collect early in the hospitalization. We identified and disregarded potential sources of leakage in the dataset by excluding variables that were likely obtained or implemented as an intervention *as a result* of the diagnosis, rather than prior to the diagnosis. Although we had access to urine output and considered creating a flag for persistently elevated creatinine, we opted not to use these features, because we suspected that would also constitute data leakage.

Modeling:

We initially chose a support-vector machine (SVM) model as our baseline and wanted to test a linear model and a tree-based model against it for comparison. We believed it would be good to test two different types of models: one with traditionally low bias and high variance, like decision trees, as well as one with high bias and low variance, like a logistic regression, so that we could find an optimal solution.

We hypothesized that a support-vector machine (SVM) model would perform well given the high dimensionality of our feature space. However, we discovered that the SVM was very inefficient (initial tests with scaled input data and default settings took ~7.5 hours to run). This was likely due to the SVM model's sensitivity to noise or overlap of feature space in relation to our target variable, increasing complexity of the calculations required to fit the model. Ultimately, while we have a good deal of dimensionality to our feature space, we also have a relatively large number of samples, so we did not find an advantage by using SVM.

Next, we pivoted to a **decision tree as our baseline model**. Although decision trees can also be sensitive to noise in the data, they are simple, easily-interpretable, and allow us to better understand the features with the highest information gain. Our baseline decision tree was instantiated using criterion='entropy' and otherwise default parameters, and was applied to the entire set of features. The AUC for this baseline model on the validation set was 75.58%. We then tested variations in the 'min_sample_split_size' and 'min_leaf_size' hyperparameters in order to determine the optimal values, which we found to be leaf size of 75 and a split size of 275 **(Appendix Figure 4)**. These parameters brought the AUC up to 89.68%. An additional

decision tree using these parameters and only the top 25 most important features yielded AUC of 89.29%, as shown in **Figure 1**.

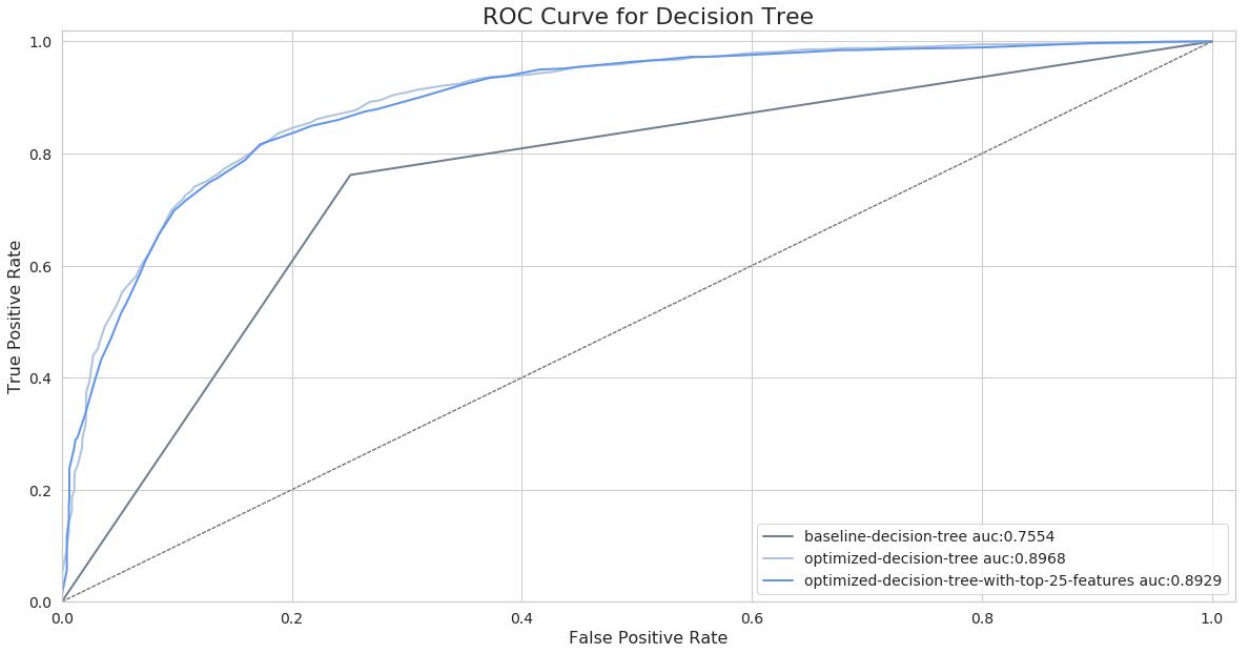

***Figure 1**: ROC curves for three decision trees. The baseline decision tree uses all features and default parameters. The optimized decision tree uses all the features and the optimal parameter set. The fully-optimized decision tree uses the optimal parameter set and the top 25 features that were used in the fit on the training data.*

Initially, we compared hyperparameter selection on model receiving operator characteristic performance within a model family. We created each model on the training set, performed feature selection on the training set, validated the model (i.e. compared different AUCs, Precision and Recall scores) on the validation set within each family and across families and then ran the final model on the test dataset.

For our purposes, model selection is based on model family, hyperparameter selection, and feature selection. To that effect, we not only varied the hyperparameters within the model families, but we also tried varying the approach to refining features for each model. Our initial raw engineering set, which included over 200 features, is sufficient as a starting point, but our goal in this process was to build an **effective and parsimonious model**. If we could stack the final feature set with tests that are performed on hospital admission and the features that are considered "most important," we could develop a model that is more realistic and can be deployed in production. Our four methods of feature selection are:

1. Domain Knowledge
    a. Conducted a literature review for features used in other prognostic models with correlation matrix for validation
2. PCA with features from the literature and PCA (unsupervised) with all the features
3. SelectKBest feature selection from scikit learn using unscaled data and the chi-sq test
4. Decision Tree
    a. Use the feature importance to determine which features are most important

For the first method, we looked at other prognostic models and a meta-analysis on prognostic models in the hospital/ICU setting. A set of 5 ICUs in Belgium used a machine-learning based AKIPredictor (Fletchet et al. 2019), and in the U.S., a simple discrete-time logistic regression model was found to be quite successful using readily available laboratory data. (Simonov et al. 2019). Comorbidities, vitals, and lab results were all commonly used features among those models and in other literature. In particular, we found that some of the most important features were age, creatinine increase within 48 hours (from a previous test), gender, and other comorbidities. We also looked at the average length of ICU stay if the patient was admitted within 30 days. **Table 1** shows the list of features we selected. A correlation heat map of these features is listed in the **Appendix (Figure 1)**.

Literature Feature
ft_age
ft_creatinine_increase_within_48
ft_avg_hematocrit
ft_hcc_cd_2_sepsis
ft_avg_icu_los_within_30
ft_baseline_creatinine
ft_gender
ft_nephrotoxin_diuretic_rx_within_24
ft_nephrotoxin_ibuprofen_rx_within_24
ft_low_blood_ph_within_12_hrs
ft_high_potassium
ft_hcc_cd_19_dbtes_wo_comp
ft_hcc_cd_29_chronic_hepatitis
ft_hcc_cd_85_chf
ft_hcc_cd_136_ckd_stg_5

***Table 1**: Features selected from the literature*

We also performed Principal Component Analysis (PCA) to determine whether reducing the number of features to principal components would improve the efficacy of the logistic regression model. PCA is typically used to extract information from a high-dimensionality feature space (such as ours) and project it into a lower-dimensional subspace. Principal components are based on (and ordered) by subspaces that can explain more of the variation in the data. We selected an N of 5 components and tried PCA with all the 204 features as well as just the features selected from the literature.

Interestingly, the PCA developed using the features from the literature was able to capture 13% variance in the first component. By contrast, the first component in the PCA on all of the data explained 5% of variance. We know that some of the features from the literature were linearly correlated (like presence of Chronic Kidney Disease in stage 3 and the average baseline creatinine) so we might find that when the high-dimensionality subspace is a subset of features that are somewhat linearly correlated, the principal components are subspaces that have more in common with those features. A 3D graph of the first three components in the PCA analysis is shown in the **Appendix (Figure 3)**.

We used Select-K-Best to score each feature against the target variable using a chi-sq test. This method is appropriate for binary data since chi-sq test is for nominal data. However, it is not appropriate for the continuous variables in the data (such as 'ft_avg_hematocrit' and 'ft_baseline_creat_gt_1'). This method selected those numeric features, but since they were important based on the literature, we did not remove them from the final list of features that were selected using this method.

Finally, using the decision tree with optimal leaf size and split size values, we selected the top 25 features that were considered important using "entropy" as our measure of information gain. After we identified the top 25 features from the optimized decision tree, we re-ran the SVM model on the validation set. We thought that by reducing the feature set to just 25, the noise would be reduced enough to generate a more separable dataset, and thus yield better results than our first test. We wanted our model to be flexible when handling outliers, so we tried increasing the margin, though as we decrease the value of the regularization weight 'C', we risk increasing the mis-classfication rate. To optimize 'C', we chose a value that kept error reasonably low while maintaining a high AUC. When increasing C, AUC converged at about .857 beginning at around C=1E-2 **Appendix (Figure 6)**.

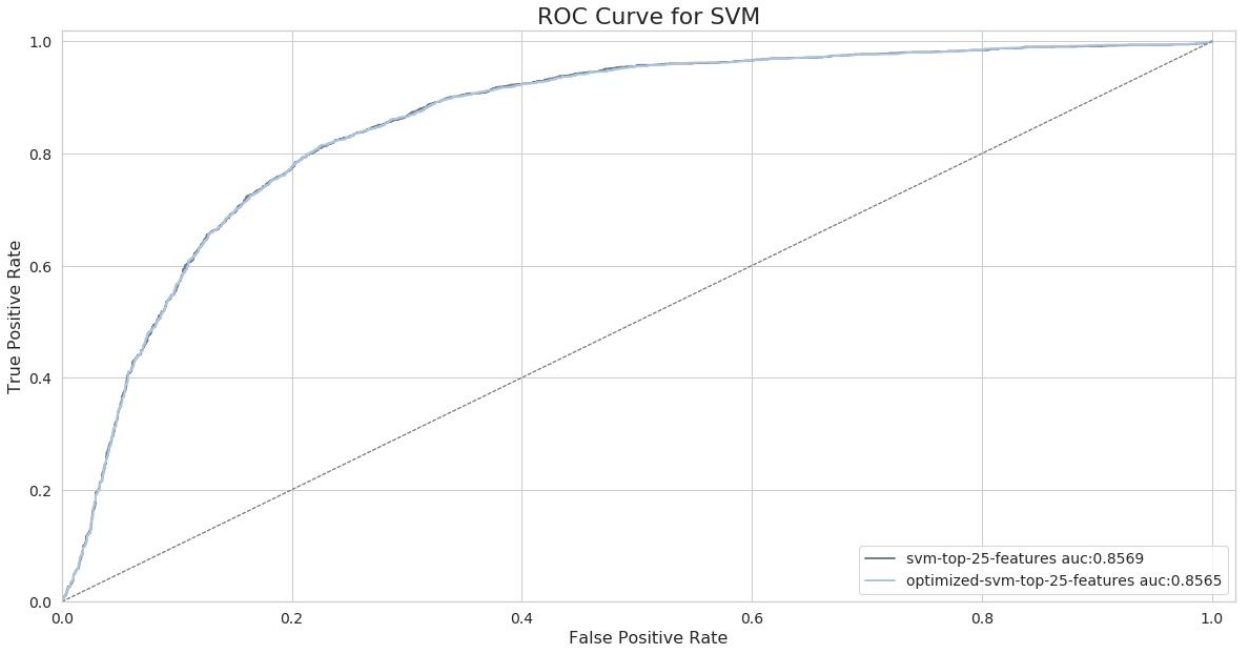

***Figure 2**: ROC curves for the baseline and the SVM based on the 25 features and with C optimization.*

We also evaluated the different feature sets by comparing the ROC curve of a logistic regression model with default parameters and all the features with the same logistic regression model fitted with each of the different feature sets. This is shown in **Figure 3.**

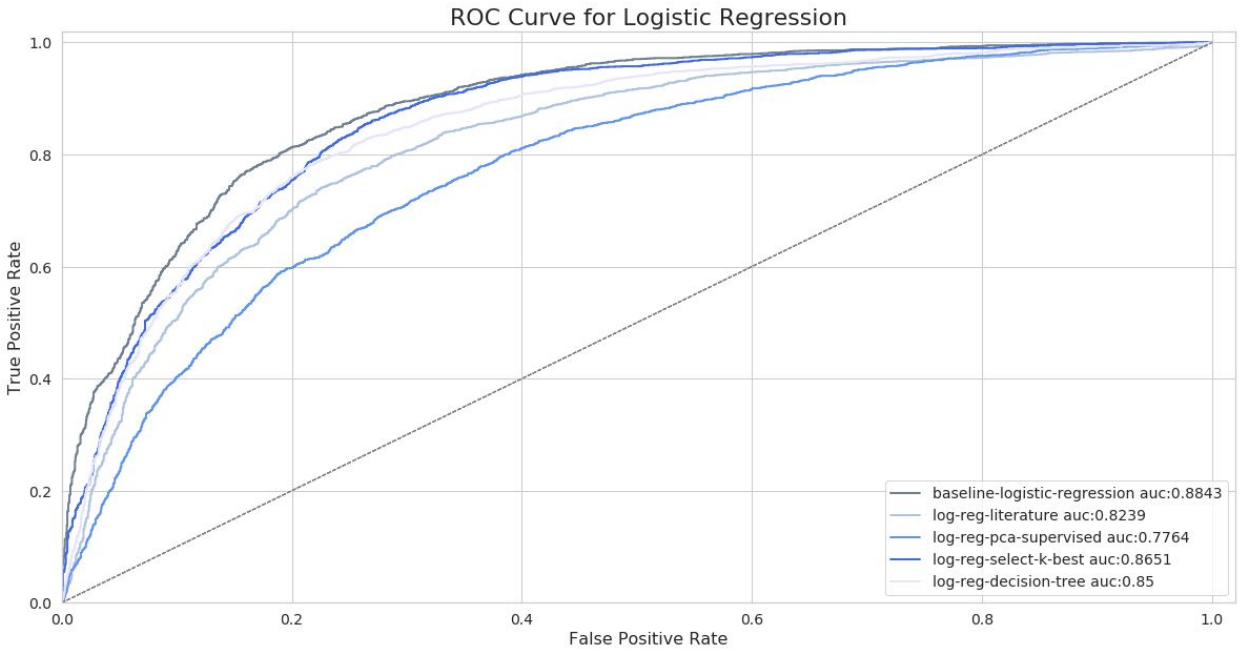

***Figure 3**: ROC Curve for a default logistic regression with different features.*

We also performed some hyperparameter selection on the logistic regression model by varying the penalty, the C, and the solver. We used a GridSearchCV which performed 200 fits for 40 candidate models with the default feature set. This resulted in a "best logistic regression" classifier for which we measured the AUC.

In summary, our competing approaches to the baseline decision tree were hyperparameter optimized decision trees, optimized decision tree with a subset of features, logistic regression with different methods of feature selection, hyperparameter optimized logistic regression with the default features and hyperparameter optimized SVM. In the next section, we will document our evaluation metric and how we compared each model using the best objective function.

Evaluation:

A false positive in our model represents a patient who would be misdiagnosed as having an AKI or is erroneously treated for one. Financial consequences of a false positive affect both the patient and the hospital. A patient may be responsible for the costs of tests and treatments, and the hospital may be mis-allocating resources. Clinical consequences include subjecting a patient to unnecessary procedures or interventions. Moreover, the patient's true diagnosis could be missed if a clinician proceeds using a false positive result.

However, a false negative represents a patient who has AKI but is not diagnosed with one. Given the severity of AKI and its complications, the risk of a false positive is greater than a false negative. Missed AKI diagnosis could result in a delay in treatment, and ultimately could cause real clinical harm to the patient. Complications of AKI include fluid buildup in the lungs, chest pain due to pericardial inflammation, muscle weakness, permanent kidney damage requiring a mechanical filtration device or transplant, or even death. The benefits of early detection and intervention outweigh the costs of managing complications that arise from an undiagnosed AKI.

For these reasons, our goal is to minimize our false negative rate, thereby **optimizing model recall**. When choosing the best algorithm, we also used the ROC curve to calculate the AUC for each model. We wanted to use the **AUC metric** since it was a commonly used metric in model evaluation in the literature. We wanted to maximize the recall and AUC, but also **limit** both the **number of features** (and thus the number of tests required at hospital admission)

and **model complexity**. A summary of the AUC, precision, and recall for our various models is presented in **Table 2**.

label	auc	precision	recall
optimized-decision-tree	0.8968	0.8148	0.8350
optimized-decision-tree-with-top-25-features	0.8929	0.8237	0.8164
grid-searched-log-reg	0.8845	0.7890	0.8279
baseline-logistic-regression	0.8843	0.7905	0.8257
log-reg-select-k-best	0.8651	0.7639	0.8430
svm-top-25-features	0.8569	0.8023	0.7523
optimized-svm-top-25-features	0.8565	0.8025	0.7514
log-reg-decision-tree	0.8491	0.7889	0.7559
log-reg-literature	0.8239	0.7736	0.7141
log-reg-pca-supervised	0.7764	0.7142	0.6710
baseline-decision-tree	0.7525	0.7491	0.7541

Table 2: *AUC, precision and recall scores for various models.*

The best models on the validation set were the models that used all the features available. These were the **optimized decision tree, the grid-searched logistic regression, baseline logistic regression, and the optimized SVM** with AUCs of 0.8968, 0.8845, 0.8843, and 0.8569 respectively.

Given that we may not have all these features available to us at the time of deployment, we wanted to choose a parsimonious alternative. **After feature selection, the model that performed the best was the optimized decision tree with the top 25 features with an AUC of 0.8929 and a precision, recall of 0.8237, 0.8164.** The highest-performing features in this model were the baseline creatinine, and the increase in creatinine. The next best was the logistic regression using selectkbest with an AUC of 0.8651 and a precision, recall of 0.7639, 0.8430.

Notably, the **PCA-based logistic regression was better than baseline but not as performant as the other feature sets**. We believe that this has to do with the dimensionality reduction and the fact that we may be losing information that would be otherwise helpful to the model. We also picked only 5 components which may not have been enough to explain all the variance in the data.

Also interestingly, the AUC for the logistic regression on the features from the literature was 0.8239 with a precision, recall of 0.7736, 0.7141. This is almost exactly consistent with the

mean AUC of 0.80 reported in the meta-analyses we found. We suspect that the additional features we created provided additional information (since we were working with a very rich dataset) and allowed us to train a model that outperformed the literature.

Given the models we had, we decided it would be prudent to **deploy the optimized decision tree with the top 25 features that were originally selected on the trained data.** This is an easy model to explain and it outperformed every other iteration based on AUC though it was very similar to the baseline and grid-search optimized logistic regression. **Table 3** shows the metrics for our final test dataset.

```
metrics for optimized-decision-tree-with-top-25-features
--------------------
auc: 0.8949070952096015
precision: 0.8470806302131604
recall: 0.7899740708729472
```

***Table 3**: Evaluation metrics on the final model chosen*

On the other hand, since the final model selected above had **higher AUC** values but **lower recall than the logistic regression with the selectkbest features**, we can argue that we might also consider deploying that model. Since the second had higher recall but lower precision, that model could be used in a scenario where it would be better to predict a false positive than a false negative.

The model solves the business problem by helping providers to identify patients with AKI who may have otherwise traversed the hospital system or would not be identified in a timely manner to receive the intervention they need. This is most useful for patients who may not exhibit the classic signs/symptoms of AKI, or for patients who present with signs/symptoms that are common to many possible diagnoses. The model can accelerate the triage process and support the decision-making processes of the providers. As a result, earlier intervention may reduce the risk of complications or progression. A study conducted in Belgium in 2019 compared machine learning models to physician prediction in early diagnosis of AKI in the ICU. The study used a classification prediction model, available online, which can be used to predict AKI at various points in the clinical course of a patient. There was no statistical difference between the model and physician classification at the time of initial diagnosis, but on average the physicians tended to overestimate the risk of AKI, suggesting that there is real value in such a model as a supplement to physician decision making.

For further exploration we would like to implement a feature for the use of contrast dyes as a predictor of AKI. Unfortunately we were not able to encode it, but it would be useful in a future iteration to predict contrast-dye-induced AKI, a subset of acute kidney injuries.

Deployment:

Ideally, the results of our data mining solution can be deployed by integrating our fitted model with electronic medical record (EMR) systems. Every patient at a hospital is represented by a unique medical record number (MRN) and has an associated medical record which contains all of their current and previous medical records at a particular institution (and outside institutions if requested). Integration with the medical record can allow a system to extract relevant data points to feed into our model, and then flag a patient for a possible AKI diagnosis. The model could be re-run after new data is added to the patient's medical record, or at regular intervals, to include the accumulation and trend of the patient's stored data.

However, EMR systems are built by privately-owned companies and are not always conducive to deploying new software. More realistically we could deploy the model to a website, wherein providers can manually enter the relevant data points. If several critical features could be identified manually during hospital admission, these could be used as a trigger for providers to run the data through the online interface. We found an example of such an approach online. A limitation with this implementation is that it requires a clinician to enter information in the first place. It is not practical to screen every patient for AKI, nor is it realistic that doctors will do this themselves. This task may be delegated to a nurse or intern. Therefore, additional hospital resources (staffing and time) may be required to complement the model deployment.

Once in production, we would want to monitor false positives and false negatives, then re-evaluate what caused the model to make those predictions. Regardless of the prediction, a clinician will always make a final diagnosis that can be used as feedback to periodically re-fit the model. Typically, software tools in clinical settings must go through a rigorous review process that may not allow for a dynamically-updated implementation. Furthermore, manual review of these inconsistencies could allow for the discovery of new features (or the elimination of features) that may be better representative of AKI, therefore improving the model. It would also be important to see whether clinicians actually use these results in order to diagnose early. If the providers already make a diagnosis, then our model didn't achieve our original goal.

Medical providers need to be aware of the limitations of the model and understand that it is not intended to replace their diagnostic judgement, but facilitate it. They also need to understand how false negatives and positives are defined, and what consequences each carries. It is not realistic to assume our model will ever be 100% accurate, even with additional refining and tuning. Ultimately, further testing is needed to define a 'successful' model. For example, the model may be less accurate than a provided, but it may still be valuable if on average it can identify possible AKI earlier than providers.

The obvious ethical issues are associated with the false positive/false negative rate, and the consequences for both the hospital and patient when these circumstances arise. If a patient is at known risk for AKI, his/her provider can inform the patient that this tool is being implemented to supplement the decision-making process. Other ethical considerations may arise based on the demographics of the training data for the model compared to the demographics of patients at the hospital where our model is deployed. Initially, we included many race features in the training data, given the differences in racial frequencies of AKI diagnoses in the literature. However, we did not identify race as having significant information gain. Even without race as a feature in our final set, race and other protected classes may be represented as proxy variables within the data. It is important to track and monitor our model's results to ensure that clinicians within hospitals do not unknowingly treat protected groups differently, and that all patients are receiving unbiased medical care.

A risk of our proposed plan is how to handle protected health information and protect patient privacy. Systems integration engineering would have to be implemented to ensure that our model, when integrated with the EMR, is in compliance with HIPAA (the Health Insurance Portability and Accountability Act).

Contributions

Lipsa	Initial Dataset Research, Literature Review, Feature Engineering, Data Pre-Processing, PCA, Feature Selection, Logistic Regression modeling, Writeup
Silas	Literature Review, Target Identification, SVM modeling, Write-up
Jordan	Literature Review, Feature Selection, Decision Tree modeling, Write-up
Jonas	Literature Review, Feature Engineering, Write-up

References

Journal Papers:

1. "Machine learning versus physicians' prediction of acute kidney injury in critically ill adults: a prospective evaluation of the AKIpredictor", Fletchet, Falini, et al. Critical Care, (2019) 23:282. https://ccforum.biomedcentral.com/articles/10.1186/s13054-019-2563-x
2. "Systematic review of prognostic prediction models for acute kidney injury (AKI) in general hospital populations" Hodgson LE, Sarnowski A, Roderick PJ, et al. BMJ Open 2017;7:e016591. doi:10.1136/ bmjopen-2017-016591
3. "A risk prediction score for acute kidney injury in the intensive care unit" Malhotra, Kashani, et al. Nephrol Dial Transplant. t (2017) 32: 814–822. doi: 10.1093/ndt/gfx026
4. "A simple real-time model for predicting acute kidney injury in hospitalized patients in the US: A descriptive modeling study" Simonov, Ugwuowo, et al. (2019) PLoS Med 16(7): e1002861. https://doi.org/10.1371/journal. Pmed.1002861
5. "World Incidence of AKI: A Meta-Analysis" Susantitaphong, Cruz, et al. Clinical Journal of American Society of Nephrology. (2013) 8 (9) 1482-1493; DOI: https://doi.org/10.2215/CJN.00710113
6. "Section 2: AKI Definition." Kidney international supplements vol. 2,1 (2012): 19-36. doi:10.1038/kisup.2011.32

Websites:

1. http://www.akipredictor.com/en/aki_predictor/
2. https://www.mayoclinic.org/diseases-conditions/kidney-failure/diagnosis-treatment/drc-20369053

Appendix

Figure 1: Correlation heatmap of features selected through the literature review.

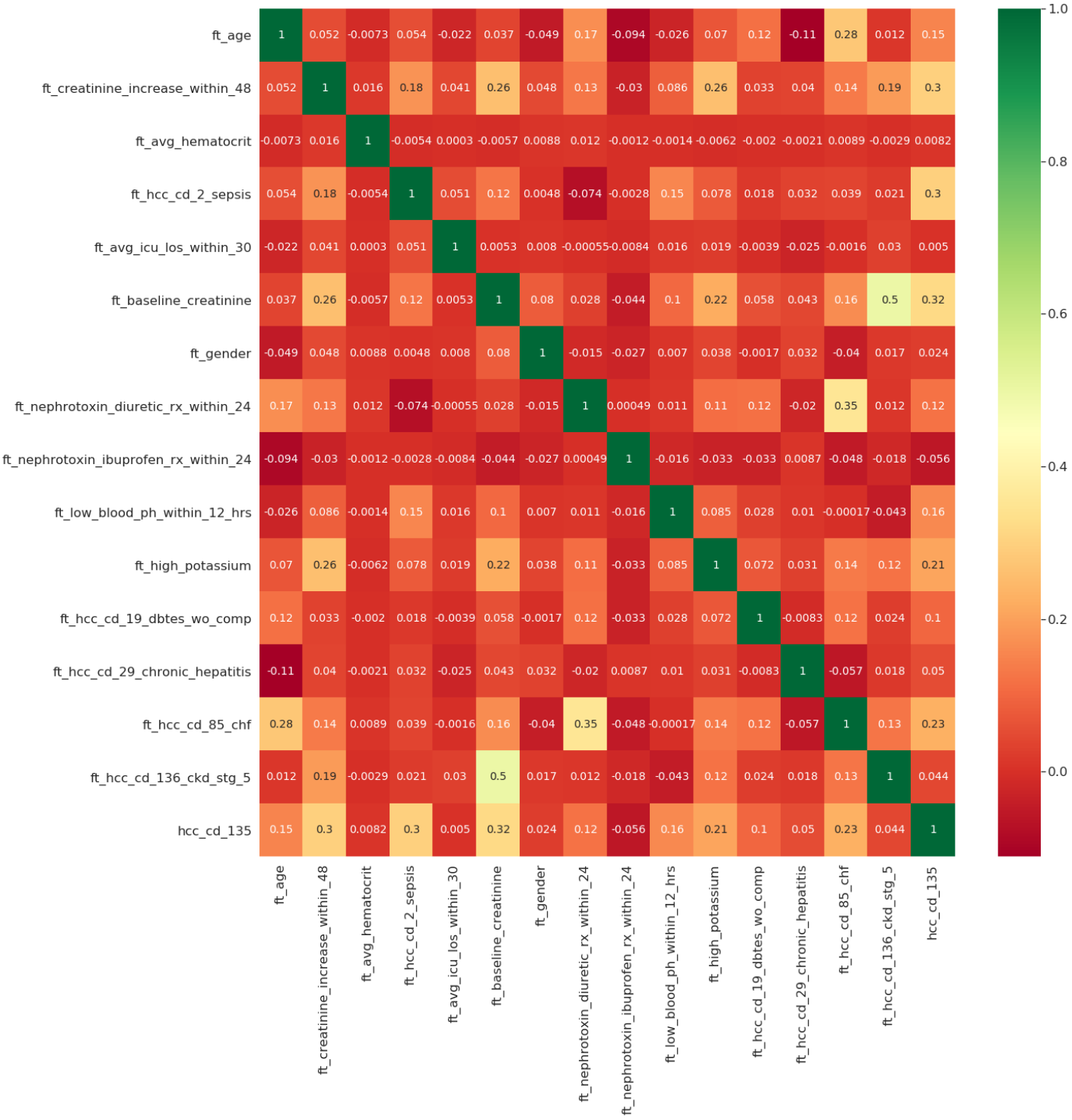

Figure 2: 25 important features used in the optimized decision tree with feature selection.

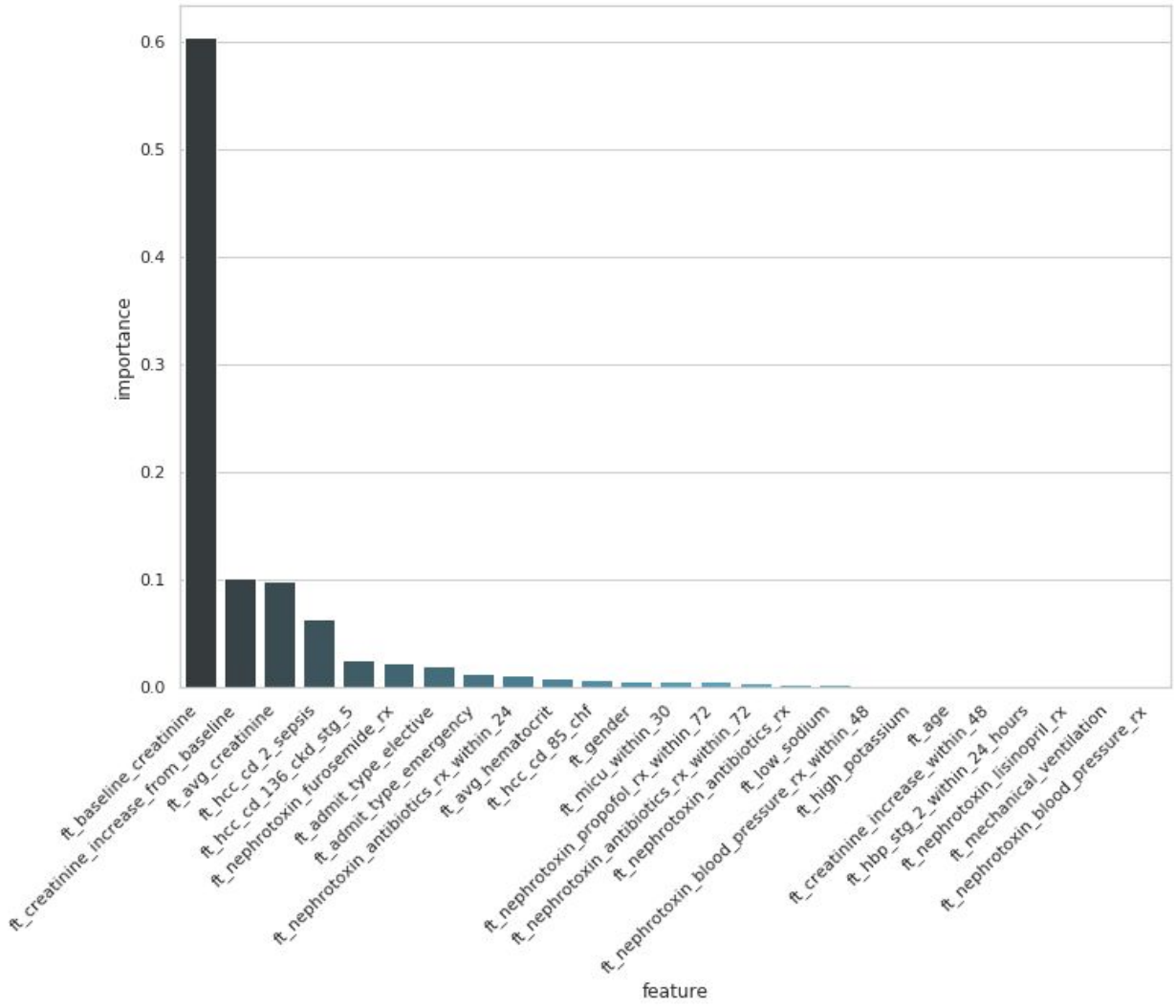

Figure 3: Visualization of PCA in just the first three dimensions. Light blue indicates a negative sample and dark blue indicates a positive sample. We can see here that the dark blue is relegated to low values on the first dimension, low values on the second dimension and somewhat higher values on the third.

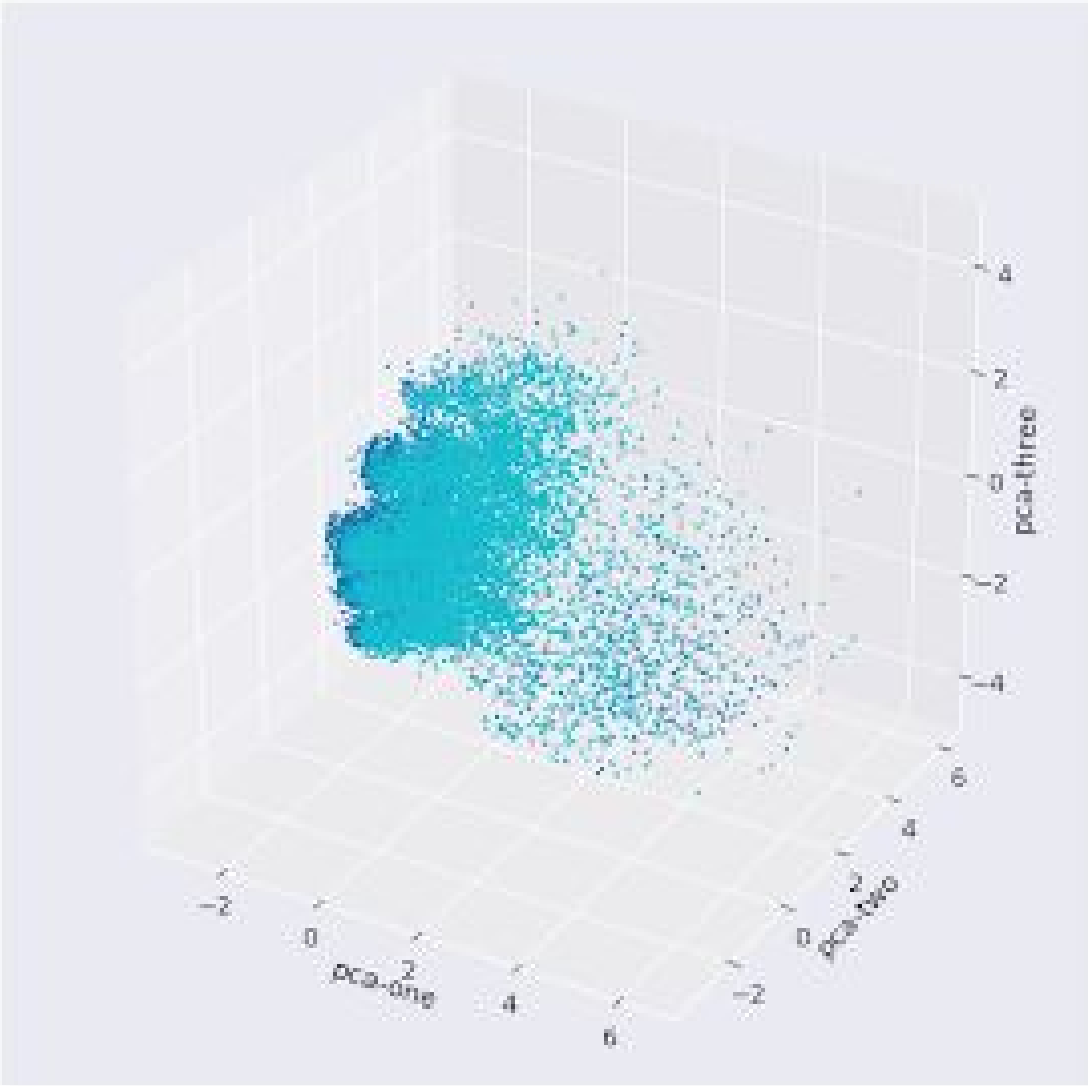

Figure 4: AUCs of 10 decision trees varying leaf and split size

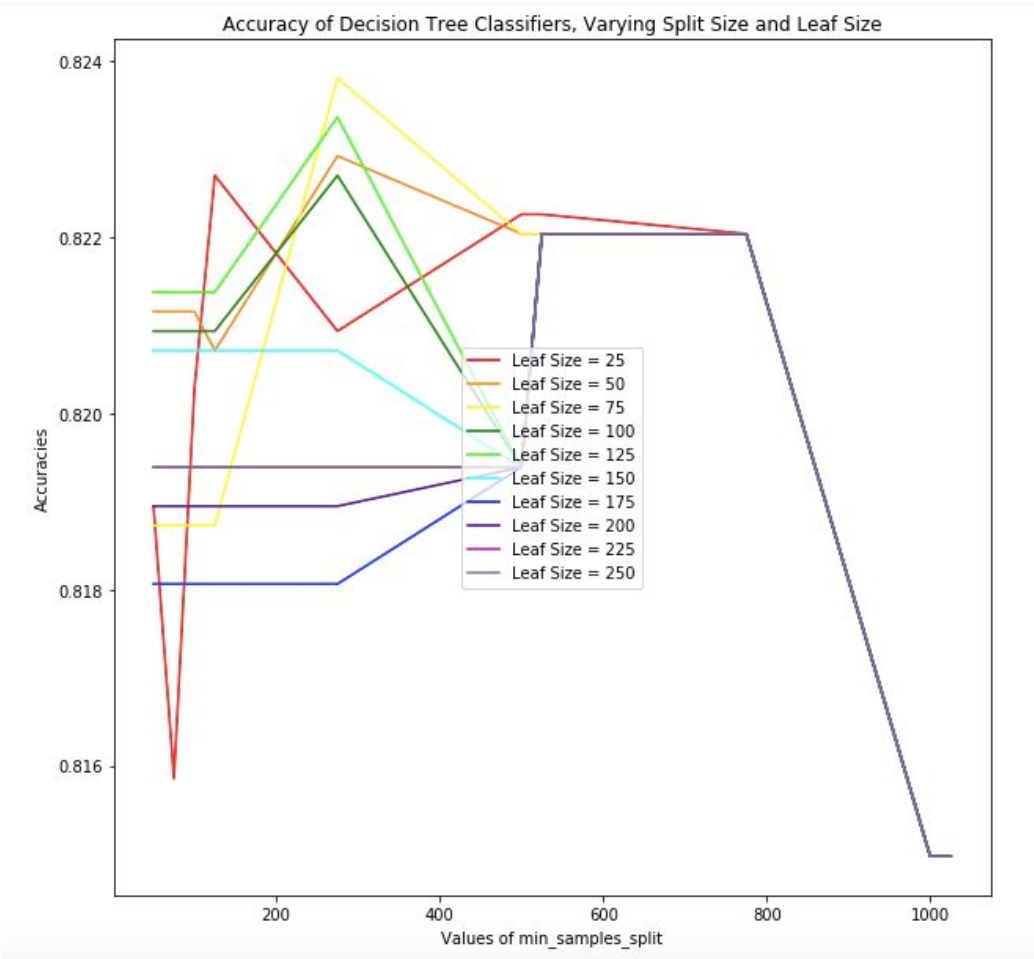

Figure 5: ROC Curves for seven SVMs varying in regularization weight (25 features)

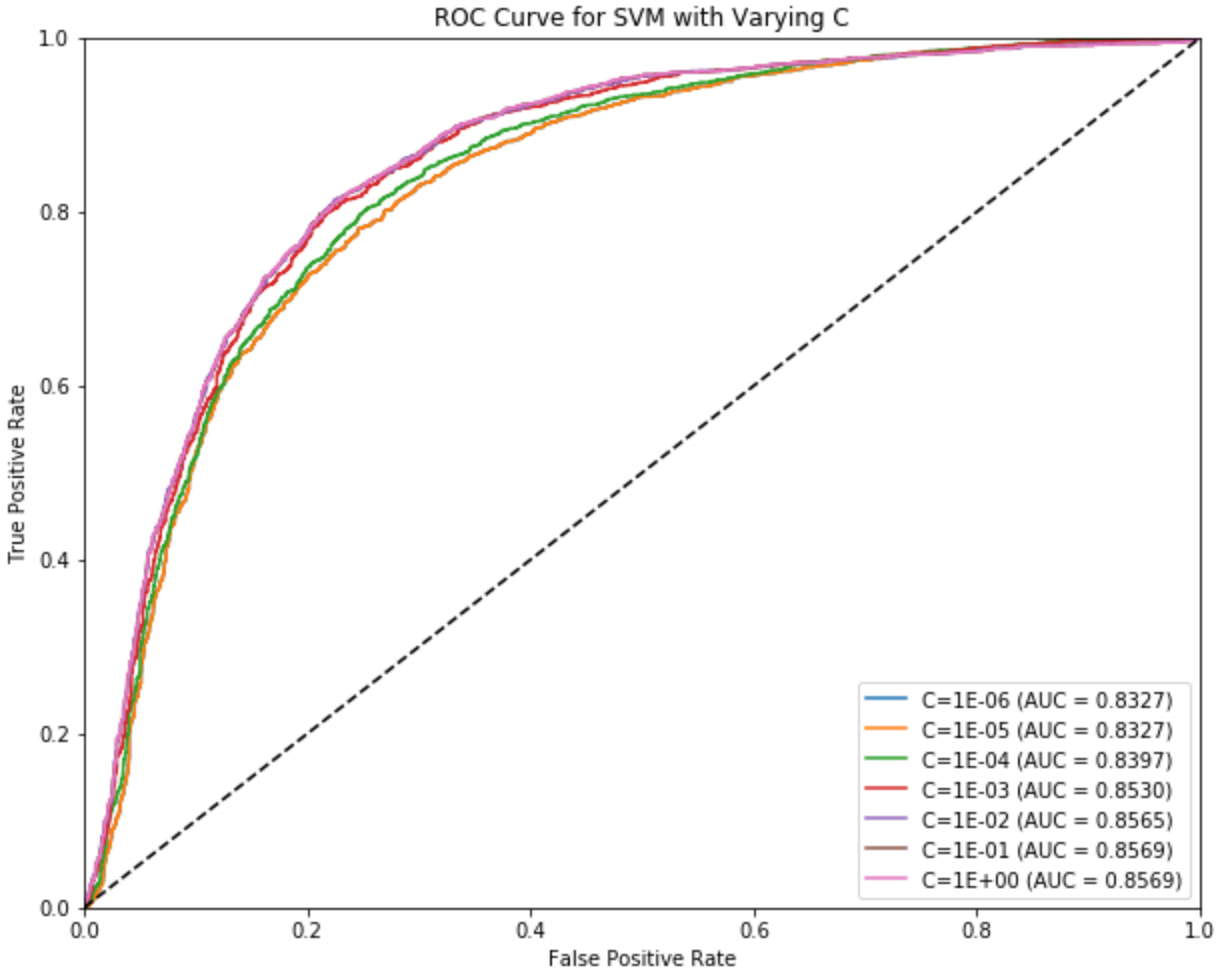

Figure 6. AUCs for seven SVMs varying in regularization weight (25 features)