Capstone Project — Alfred Geoffrey Lyle

Machine Learning Nanodegree Project — October 2, 2017

Demographic Features Leading to Increased Risk for Patient Mortality

Definition

Project Overview

In 2010 there were 715,000 inpatient hospital deaths in the United States[i]. This accounts for 28.6% of deaths in the United States. Studies show that increased patient age and length of stay contribute to patient mortality[i]. It has also been shown that females have significantly higher hospitality mortality rates than males[ii]. However, when it comes to other patient features such as race/ethnicity, there is no consensus on whether they affect patient mortality.

The 2010 census shows the United States population is growing older as those aged 45 and over increased by 5% in 10 years[iii]. As the population grows older, it can be assumed the number of hospital visits will increase. Thus it will be useful for hospitals to continue to determine which factors lead to increased mortality.

As hospitals transition from pen and paper records to electronic medical records, the concern of the security of patient medical data becomes more relevant. In 2016 there were 374 security breaches in the Health/Medical industry, a 36% increase from 2015[iv]. While data being collected for research purposes is disidentified according to rules and regulations, the data stored in medical records contain much information that can be used to identify and target certain individuals. The challenge is to see what information is useful for hospitals and machine learning algorithms in predicting patient mortality.

Problem Statement

The goal of this project is to use hospital Intensive Care Unit (ICU) data to:

1. Confirm patient age and length of stay can be used to determine patient mortality.
2. Determine if adding features ‘gender’ and ‘insurance’ type will improve the ability to predict patient mortality.

3. See if adding optional features such as ‘ethnicity’, ‘language’, ‘religion’ and ‘marital status’ improve predicting patient mortality.

Metrics

When comparing the classification models, I will be using the Area Under Curve (AUC) metric. As the comparisons will be whether a patient lived or died during their ICU stay is binary (0 for survived, 1 for expired) this metric can be used. When looking at the full data set, there are 15737 patients who died out of 46476 total patients. Thus if I used accuracy to test the models, a benchmark model that predicted all patients survived would have an accuracy of 67%. Using an AUC metric, the baseline false positive to true positive rate would be 0.5 with a perfect model having an AUC score of 1.0. As the overall patient mortality rate changes based on the section of data points being used, using an AUC score will allow me to more easily compare models using different features.

Analysis

Data Exploration

The dataset used in this project is the MIMIC-III (**M**edical **I**nformation **M**art for **I**ntensive **C**are III) database[v]. This database contains deidentified medical data for over forty-thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. Requesting access to this database can be done at https://mimic.physionet.org/gettingstarted/access/. To be granted access to the database, it is necessary to complete the CITI “Data or Specimens Only Research” course.

The MIMIC-III data is contained within different tables in a PostgreSQL database. For this study patient information will be obtained from the ADMISSIONS, PATIENTS, and ICUSTAYS tables. Patient data can be crosslinked between tables using SUBJECT_ID and HADM_ID. Each hospital admission has a unique HADM_ID and each patient has a unique SUBJECT_ID. It is possible for tables to have duplicate SUBJECT_ID, indicating the patient had multiple hospital admissions. From the ADMISSIONS table I will be selecting the SUBJECT_ID, HADM_ID, ADMITTIME, DISCHTIME, DEATHTIME, INSURANCE, LANGUAGE, RELIGION, MARITAL_STATUS, and ETHNICITY columns. From the PATIENTS table I will be selecting the SUBJECT_ID, GENDER, DOB, DOD, and EXPIRE_FLAG columns. From the ICUSTAYS table I will be selecting the SUBJECT_ID, HADM_ID, and LOS columns.

In this data, there is no patient age data. Furthermore, for patients aged 89 years and older, the patient DOB (Date of Birth) was shifted 300 years in order to protect patient confidentiality.

Exploratory Visualization

When exploring the data it is important to determine the distribution of the target variable, patient ICU survival. When looking at patient survival rates, the MIMIC-III data has similar rates compared to those found in previously published studies. Figure 1-1 shows the female mortality rate is slightly higher than the male mortality rate. Interestingly, patient mortality increases in patients aged 65 years and older; however, patient mortality is lower in those 85 years and older compared to patients aged 75-84 years (Figure 1-2). Initial observations of the data show there are differences among the groups of patients. Therefore, it is likely that a supervised learning algorithm will be able to create a decision boundary that will provide predictive power in determining which patients have a higher mortality risk.

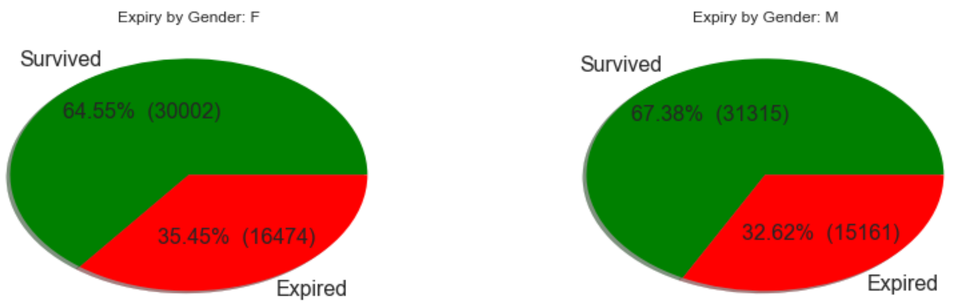


Figure 1-1: Mortality rates by gender

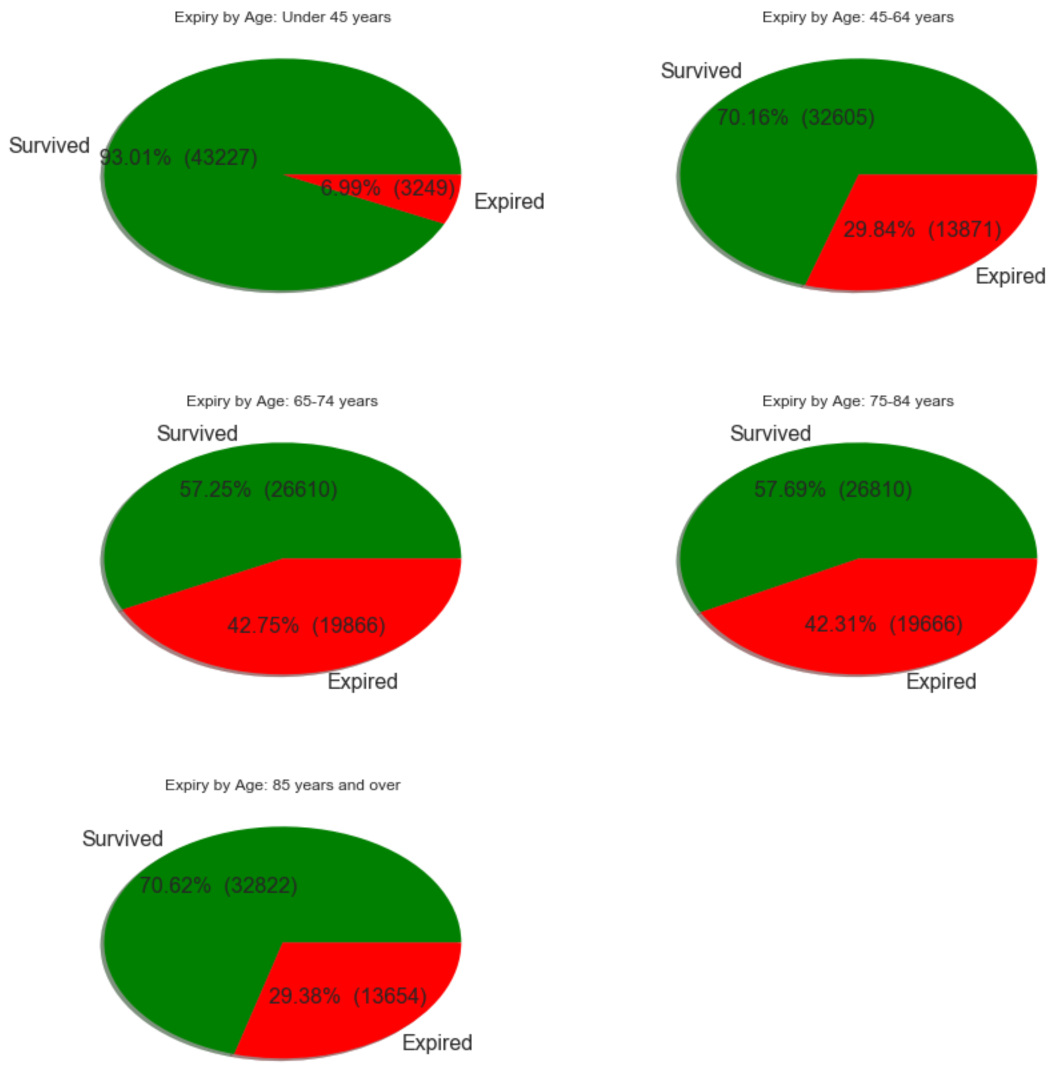


Figure 1-2: Mortality rates by age

Algorithms and Techniques

There were a variety of supervised learning algorithms used classify the data. As I did not know which algorithm would be best at classifying the data, I used GaussianNB, DecisionTreeClassifier, AdaBoostClassifier, SVM and RandomForestClassifier. These models are compared against each other using the AUC scorer. These algorithms were made available through the Sci-Kit Learn module[vi].

Benchmark Model

The benchmark model is created from using the two features the CDC agree affect patient mortality, patient age and LOS. The model will be the algorithm with the highest AUC score. A similar study on predicting 30 day hospital readmissions was hosted on Kaggle. This competition contained data including patient age, gender, and length of stay. The top team created a model with an AUC score of 0.68469 and the 10th place team had an AUC score of 0.64823[vii]. The benchmark model should have an AUC score in this range.

Methodology

Data Preprocessing

I imported the Psycopg2 module to retrieve the data from the PostgreSQL database. Then a cursor was created to access the target database and select the needed columns from it. All rows were fetched and put into a Pandas DataFrame.

```
#Import necessary modules

import psycopg2
import psycopg2.extras

#Access relevant fields from ADMISSIONS table
curr2 = conn.cursor()
try:
    curr2.execute("SELECT SUBJECT_ID, HADM_ID, ADMITTIME, DISCHTIME, DEATHTIME, 
INSURANCE, LANGUAGE, RELIGION, MARITAL_STATUS, ETHNICITY from ADMISSIONS")
except:
    print "I cannot SELECT from ADMISSIONS"

#Get column names for Pandas DataFrame
admcolnames = [desc[0] for desc in curr2.description]
#Create Pandas DataFrame
dfadm = pd.DataFrame(curr2.fetchall(), columns=admcolnames)
print dfadm.head() #Get data sample
print len(dfadm.index)
curr2.close()
```

As there was no patient age category, the patient age was created by subtracting the patient's DOB from their admission time. Thus patients were grouped according to the CDC groups in their study. Patient LOS (Length of Stay) was also grouped similar to the CDC groups. Grouping these numerical values this way also helps us deal with any outliers in the data.

The data from the different tables were merged together on the SUBJECT_ID column. Duplicates in the HADM_ID and SUBJECT_ID columns were dropped. Furthermore, many columns had rows with missing, 'UNOBTAINABLE', 'UNKNOWN (DEFAULT)' or NaN values. When using these features for classification these rows were dropped.

One-hot encoding was used to transform the non-numerical features into dummy variables.

Implementation

The implementation process was carried out in three phases:

1. Training and Testing Phase
2. Evaluation Phase
3. Data Gathering Phase

In the training and testing phase, a classifier was fit to the data, and then ran against test data. The AUC score and accuracy were input into a dictionary with the classifier name as the key.

```
def train_results(learner, X_train, y_train, X_test, y_test):
    '''
    inputs:
        -learner: type of learning algorithm to be compared
        -X_train: features training set
        -y_train: expiry training set
        -X_test: features test set
        -y_test: expiry training set
    '''

    results = {}

    #Fit the learner to the training data
    learner = learner.fit(X_train, y_train)

    #Get the predictions on the test set,
    predictions_test = learner.predict(X_test)

    #Compute accuracy on test set
    results['acc_test'] = accuracy_score(y_test, predictions_test)

    #Compute AUC score on the test set
    results['auc_test'] = roc_auc_score(y_test, predictions_test)

    # Return the results
    return results
```

To ensure the AUC score would be valid over multiple trials, the classifiers were trained and tested multiple times with the data randomly split. The AUC scores were then averaged and the classifier with the highest average AUC score was identified.

```
def best_classifier(features, expiry, trials, classifiers):
    '''''
        -features: A dataframe containing the features for training
        -expiry: Target variable, binary patient death or survival
        -trials:  An integer denoting the number of random iterations
        -classifiers:  A list of supervised learning classifiers
    '''

    #Run through multiple times using random states
    results = {}
    for clf_name in classifiers:
        results[clf_name] = {}
    for trial in range(trials):
        state = random.randint(0,429496729)
        #print "Trial {}: Random State is {}".format(trial+1, state)

        #Create test and train samples
        X_train, X_test, y_train, y_test = train_test_split(features, expiry,
                                                 random_state=random.randint(0,429496729))

        #  Initialize the models
        clf_A = GaussianNB()
        clf_B = tree.DecisionTreeClassifier(random_state=state)
        clf_C = RandomForestClassifier(random_state =state)
        clf_D = AdaBoostClassifier(random_state = state)
        #clf_E = SVC(random_state = state)

        # Collect results on the learners
        for clf in [clf_A, clf_B, clf_C, clf_D]:
            clf_name = clf.__class__.__name__
            results[clf_name][trial] = \
            train_results(clf, X_train, y_train, X_test, y_test)

    #Find average AUC scores
    average_scores = {'accuracy': {}, 'auc': {}}
    #print results
    for clf in results:
        auc_scores = 0
        accuracy_scores = 0
        for trial in results[clf]:
            auc_scores+= results[clf][trial]['auc_test']
            accuracy_scores += results[clf][trial]['acc_test']
        average_scores['auc'][clf] = auc_scores/num_trials
        average_scores['accuracy'][clf] = accuracy_scores/num_trials

    # Print final AUC scores
    for clf in results:
        print "{} has an average AUC test score of {:.6f}".format(clf, average_scores['auc'][c
lf])
        print "{} has an average accuracy test score of {:.6f}".format(clf, average_scores['ac
curacy'][clf])
    #Print classifier with max AUC score
    max_auc_clf = max(average_scores['auc'].iterkeys(), key=(lambda key: average_scores['auc']
[key]))
    print "{} has highest AUC test score of {:.6f}".format(max_auc_clf, average_scores['auc'][
max_auc_clf])
    max_accuracy_clf = max(average_scores['accuracy'].iterkeys(), key=(lambda key: average_sco
res['accuracy'][key]))
    print "{} has highest accuracy score of {:.6f}".format(max_accuracy_clf, average_scores['a
ccuracy'][max_accuracy_clf])
```

With the classifier with the highest AUC score identified. It was then necessary to evaluate the classifiers using different feature sets. A function was created to run quickly through the data and evaluate how the classifiers scored using different feature combinations.

```
def classifier_for_features(df, features, trials):
    '''''
    df - The Pandas DataFrame with the data for training and testing
    features - A list of column features the classifier will use for training
    trials - An integer for the number of random trials the best classifier function will run
    '''

    #Preprocess feature data
    input_features_raw = df[features]
    input_features = pd.get_dummies(input_features_raw)

    #Get target variable
    expiry = df['expire_flag']

    #Assign dictionary name to features
    features_key = ' '.join(features)
    if df.name not in best_classifiers:
        best_classifiers[df.name] = {features_key: []}
    else:
        if features_key not in best_classifiers[df.name]:
            best_classifiers[df.name][features_key] = []

    #Add features to list for future data collection
    if features not in previous_features:
        previous_features.append(features)


    #Run best_classifier function
    best_classifiers[df.name][features_key] = \
    best_classifiers[df.name][features_key] + best_classifier(input_features, expiry, trials,
clf_name_list)
```

Refinement

When running best_classifier function, I found that the training and testing of the SVM added over 60 seconds per trial. As each set of features was run with 10 trials in order to reduce the randomness of the AUC score, adding the SVM as a potential classifier would add over an hour to the total processing time. Also, preliminary results showed the SVM was actually one of the weakest classifiers. Therefore, the SVM was dropped as a potential classifier during testing and training.

Results

Model Evaluation and Validation

The dfmimic DataFrame, which includes all 46476 patient records, gives the following scores for non-optional feature combinations.

dfmimic		
age los gender insurance	GaussianNB	0.713391
age los insurance	GaussianNB	0.712831
age los	GaussianNB	0.702392
age los gender	GaussianNB	0.696713

Table 1-1: dfmimic DataFrame AUC scores

This table proves the initial assumption that patient age and LOS affect patient mortality. Of greater interest is how the non-optional features improve the AUC score. When the gender features are added to the classifier, it performs worse in predicting patient mortality. However, gender combined with the insurance features performs better than the baseline model and the best of the four combinations. This shows that using all non-optional features should aid hospitals in predicting patients with higher mortality risk.

dfmimic		
age gender insurance	GaussianNB	0.710751
age insurance	GaussianNB	0.710572
age gender	GaussianNB	0.690434

Table 1-2: dfmimic DataFrame AUC scores with 'los removed

While the AUC scores drop a little when LOS is removed, it does not appear to drop significantly. Thus, while it could be useful to know how long a patient has been in the ICU and tailoring care to this patient accordingly, it is not a necessary feature for predicting patient mortality.

The top classifiers and AUC scores for each DataFrame/feature combination is represented in Table 1-1. The scores are presented in order from highest to lowest. DataFrames are named by which data subsets have been cleaned of NaN and missing values (e=ethnicity, r=religion, ms=martial_status, l=language; ex. df_er uses 'ethnicity' and 'religion' features).

DataFrame	Features	Classifier	AUC_score
df_ethnicity	age los gender insurance	GaussianNB	0.713590
dfmimic	age los gender insurance	GaussianNB	0.713391
df_er	age los gender insurance	GaussianNB	0.698317
df_religion	age los gender insurance	GaussianNB	0.698027
df_marital_status	age los gender insurance	GaussianNB	0.666114
df_msr	age los gender insurance	GaussianNB	0.661452
df_ems	age los insurance	RandomForestClassifier	0.661383
df_msl	age los insurance	GaussianNB	0.656994
df_language	age los gender insurance	GaussianNB	0.655570
df_erms	age los insurance	GaussianNB	0.653822
df_msrl	age los insurance	GaussianNB	0.653046
df_elms	age los insurance	GaussianNB	0.652262
df_rl	age los insurance	GaussianNB	0.651184
df_el	age los insurance	GaussianNB	0.650352
df_emsrl	age los gender insurance	GaussianNB	0.648243

Table 1-3: Best AUC score for DataFrame by column features

From the table, I can see that only two combinations of non-optional features, (age los gender insurance, age los insurance) give the highest AUC scores for each DataFrame regardless of optional features (ethnicity, religion, marital_status, language) presence. The top 6 classifiers use all four of the non-optional features.

To validate the best model (Dataframe is df_ethnicity, contains all patient records except for those that had NaN or missing values for ethnicity value; column features used by the classifier are 'age', 'los', 'gender' and 'insurance') I ran the classifier multiple times and output the AUC score. As the classifier randomly groups patients into training and testing samples, and the starting position of the classifier is randomized each time, if the AUC score is significantly different from the multiple trials it would show this model is not reliable. The results from 10 validation trials are as show in Table 1-4.

Trial 0	0.713665
Trial 1	0.711882
Trial 2	0.713592
Trial 3	0.711102
Trial 4	0.711697
Trial 5	0.713888
Trial 6	0.713485
Trial 7	0.711997
Trial 8	0.714496
Trial 9	0.711447
Range	0.003394
SD	0.001214
Mean	0.712725

Table 1-4: df_ethnicity validation AUC scores

Performing a one sample t-test for significance gives a two-tailed P-value of 0.0508, which is not quite significant.

Justification

In order to show this model is significantly better than the baseline model (uses the dfmimic Dataframe with all patient records included; column features used are 'age' and 'los'), I ran the validation function on the baseline model (results in Table 1-5) and performed another a one sample t-test.

Trial 0	0.699678
Trial 1	0.700335
Trial 2	0.699831
Trial 3	0.696994
Trial 4	0.695212

Trial 5	0.698583
Trial 6	0.70152
Trial 7	0.700173
Trial 8	0.698266
Trial 9	0.699134
Range	0.006308
SD	0.001818
Mean	0.698973

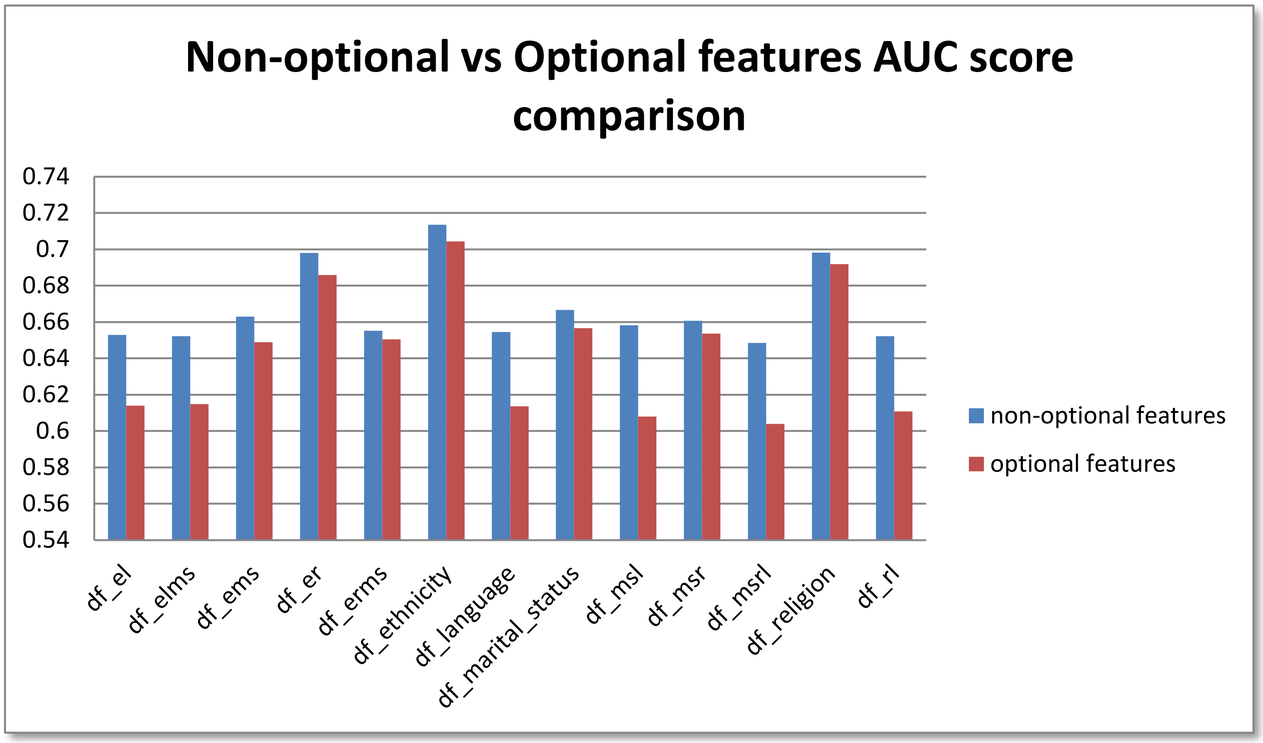
Table 1-4: dfmimic baseline validation AUC scores

The one sample t-test gives a P-value of less than 0.0001, which is considered extremely significant.

Conclusion

Free-Form Visualization

One interesting aspect is how adding optional demographic features to the classifiers consistently dropped the AUC score.



This appears to show that these demographic features are not useful in helping a learning algorithm predict patient mortality. While these features may be important in the medical field for disease diagnosis and patient care, which are important to the patient, they do not seem to affect patient outcomes. The best predictors use patient age and length of stay in combination with patient gender and type of insurance.

Reflection

This project had many interesting challenges that gave me insight into the many different aspects of applying machine learning to problem. As the MIMIC-III database contained a wide range of different medical information, I found the data exploration aspect to be particularly challenging. With so many possible features to choose, it was difficult to narrow the scope of the project. The data preprocessing was also necessary in order to remove groups containing too few records. One interesting aspect is how classifiers using the DataFrame df_ethnicity, which contained over 8000 fewer medical records than the dfmimic DataFrame, had higher AUC scores than the dfmimic DataFrame. However, using 'ethnicity' as a feature led to the classifiers having lower AUC scores. This seems to suggest the algorithms are being affected by the 'curse of dimensionality'. While this reduced DataFrame may have cleaned up difficult to classify patients, the extra features made it more difficult to consistently predict patient mortality. Indeed, this may be the reason adding all the non-optional features consistently led to lower AUC scores. A final caveat is that while a model can be created to predict mortality quite well, it is very specific in its scope. This model should only be used by the Beth Israel Deaconess Medical Center as it specifically trains itself only using data from that hospital. While it would be very easy to continue to update this model to predict increased patient mortality at Beth Israel Deaconess Medical Center, it has limited usefulness in other hospitals or even on a national scale.

Improvement

While the MIMIC-III database is one of the largest publicly available databases for patient medical records, I feel that machine learning algorithms working with a much larger data set would be able to produce a better model. If the scope of the project were expanded to the national scale for a country like the United States of America, there would be approximately 4 million data points for one year's worth of data[viii]. This is about 100x more data than that contained in the MIMIC-III database. A model using more data would likely have a higher AUC score, and would be a better determinant of what role demographic features like ethnicity and marital status have in predicting patient mortality.

[i] Margaret Jean Hall, Ph.D.; Shaleah Levant, M.P.H.; and Carol J. DeFrances, Ph.D., "Trends in Inpatient Hospital Deaths: National Hospital Discharge Survey, 2000–2010", *Centers for Disease Control and Prevention*, March 2013, https://www.cdc.gov/nchs/products/databriefs/db118.htm

[ii] Becker, Edmund R., and Ali Rahimi. “Disparities in Race/ethnicity and Gender in in-Hospital Mortality Rates for Coronary Artery Bypass Surgery Patients.” *Journal of the National Medical Association* 98.11 (2006): 1729–1739. Print.
[iii] https://www.census.gov/prod/cen2010/briefs/c2010br-03.pdf
[iv] http://www.idtheftcenter.org/images/breach/Overview2005to2016Finalv2.pdf
[v] MIMIC-III, a freely accessible critical care database. Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. Scientific Data (2016). DOI: 10.1038/sdata.2016.35. Available from: http://www.nature.com/articles/sdata201635
[vi] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
[vii] “Private Leaderboard - Predicting 30 Day Hospital Readmissions”, *Kaggle Inc.*, April 2015, https://inclass.kaggle.com/c/predicting-30-day-hospital-readmissions/leaderboard/private
[viii] George Washington University. "Nearly 50 percent increase in ICU admissions in U.S., new study says." ScienceDaily. ScienceDaily, 14 May 2013. <www.sciencedaily.com/releases/2013/05/130514212946.htm>.