Prediction of Hospital Admission after Initial Triage for Patients Presenting to Emergency Department

DS5559: Big Data Analytics (Summer 2020) - Final Project Report

Thomas Hartka(trh6u), Alicia Doan(ad2ew), Michael Langmayr(ml8vp)

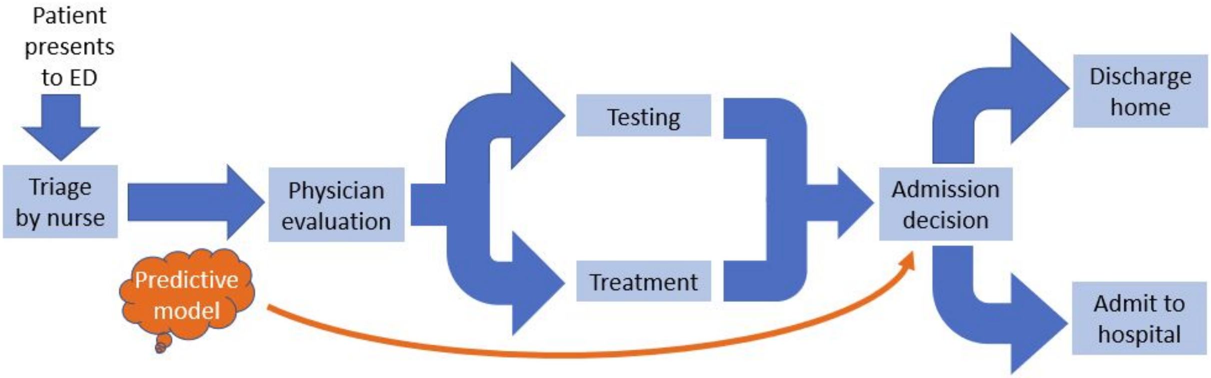
ABSTRACT

After evaluation and treatment in the Emergency Department (ED), patients are either discharged home or admitted to the hospital for further management. This decision is usually made near the end of the patient's stay in the ED. However, an in-patient room could be prepared or discharge planning facilitated if we could accurately predict which patients require admission earlier in their stay. For this project we obtained data from a national sample of ED visits from 2007-2017. We examined the ability of data available after triage to predict hospital admission. The variables examined included demographic information, vital signs, pre-existing health conditions, and the reason for the visit. We experimented with logistic regression, ridge regression, lasso regression, random forest, and a linear support vector machine to find the model with the best performance. We found that all models performed approximately equally well, as long as the model compensated for the class imbalance in the training data. For models that used all the available variables, the AUROCs were 0.825-0.836 and F1 scores were 0.416-0.423. We conclude that it is feasible to predict hospital admissions and that any of these models would be appropriate.

INTRODUCTION

The large majority of all patients that are admitted to in-patient wards in the hospital come through the Emergency Department (ED). Patients who present to the ED are first evaluated by a nurse in triage. That nurse usually measures vital signs (heart rate, blood pressure, respiratory rate, temperature, oxygen saturation, and pain score if applicable) and determines the reason that the patient is seeking care. The list of known medical problems and current medication list is also updated in the medical record at that time. The patient is then sent to the waiting room until a treatment room is available. Once at a treatment room, a physician evaluates the patient then orders tests and treatments. After the evaluation is complete and test results are available, the physician determines if the patient requires admission to an in-patient ward of the hospital.

Once a patient is determined to need in-patient admission, a bed must be made ready, the in-patient physicians need to evaluate the patient, and the ED nurses must hand off care to the ward nurses. This process could be accelerated if the need for admission could be determined earlier in the patients visit. However, determining which patients require admission versus those who can be safely managed as an outpatient is a difficult decision, which typically requires years of experience to perform accurately. Medical analytics has the potential to assist making these types of decisions through using large amounts of patient data to make predictions.



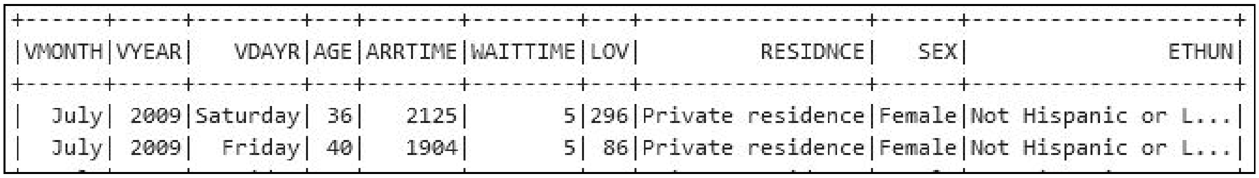
The objective of this project was to create a model to predict the need for hospital admission using only the data available when a patient is triaged.

DATA DESCRIPTION

Our data is from the National Hospital Ambulatory Medical Care Survey (NHAMCS). This is a stratified sample of data gathered from Emergency Departments (EDs) from around the United States collected by the CDC. Data from the years 2007-2017 are publically available at:

https://www.cdc.gov/nchs/ahcd/datasets_documentation_related.htm

The data is provided in tabular format, with one table for each year. Each row represents a patient encounter and each column is a variable associated with the encounter. The values for categorical and numeric variables were all in string format. An example of the data format with first 10 columns is shown below:



```
+------+-----+--------+---+-------+--------+---+-----------------+------+--------------------+
|VMONTH|VYEAR|   VDAYR|AGE|ARRTIME|WAITTIME|LOV|         RESIDNCE|   SEX|               ETHUN|
+------+-----+--------+---+-------+--------+---+-----------------+------+--------------------+
|  July| 2009|Saturday| 36|   2125|       5|296|Private residence|Female|Not Hispanic or L...|
|  July| 2009|  Friday| 40|   1904|       5| 86|Private residence|Female|Not Hispanic or L...|
```

The data collected in NHAMCS about each patient’s demographic data (age, sex, residence, ethicity, and race), known pre-existing health conditions, and vital signs. Regarding the ED visit, the data includes the tests that were run, the medications administered, and the ultimate disposition of the patient. The disposition might be admission to the local hospital, transfer to another hospital, or discharge home.

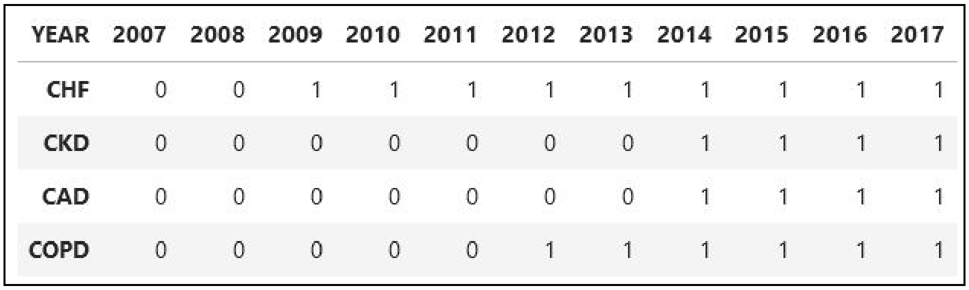
METHODS

Data import and preprocessing

Combining data by year

We first imported each year from NHAMCS data from 2007-2017 from CSV files into PySpark and concatenated the data into one dataframe (‘010-Combine_data.ipynb’). The combined data was then output into a parquet data structure.

The data collected for NHAMCS by the CDC changes slightly from year to year. We therefore analyzed which variables were present in each year (‘100-Determine_data_by_year.ipynb’). Below is an example of the data for various diseases available for different years with 0 indicating the data was not collected:



YEAR	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
CHF	0	0	1	1	1	1	1	1	1	1	1
CKD	0	0	0	0	0	0	0	1	1	1	1
CAD	0	0	0	0	0	0	0	1	1	1	1
COPD	0	0	0	0	0	1	1	1	1	1	1

We found that many of the variables regarding pre-existing illnesses that we thought would have high predictive value were only recorded after 2014, such as congestive heart failure (CHF) and chronic kidney disease (CKD). We therefore limited our analysis to 2014-2017.

Variable selection

This model is intended to be used to predict admission immediately after a patient is triaged. Therefore, only variables that could be known at that point in the patient’s visit were included in our models. For instance, variables were not included that contained data about which tests were ordered or which medications were administered. The variables that were considered in the models were the following:

- **Demographics**: Age, Sex, Type of residence
- **Arrival**: Time of day, Day of the week, Year of visit
- **Reason for visit**: Up to five reasons
- **Vitals signs**: Heart rate, Respiratory rate, Blood pressure, Temperature, Oxygen saturation, Pain score
- **Pre-existing health conditions**: Alzheimer's disease, Asthma, Coronary artery disease, Cancer, Cerebrovascular disease, Congestive heart failure, Chronic kidney disease , Chronic obstructive pulmonary disease, Depression, Diabetes, HIV, End-stage renal

disease/dialysis, Alcohol abuse, History of pulmonary embolism, Hypertension, Hyperlipidemia, Obesity, Obstructive sleep apnea, Osteoporosis, Substance abuse, Total number of chronic diseases
- **Injury data**: Was visit injury related, Did the injury occur >72 hours prior to visit

Preprocessing data

The data for each column was initially stored as strings, so we then turned to convert the data into an appropriate format (‘200-Preprocessing’).

- The AGE variable was converted to an integer and the special values were assigned an integer value ('Under one year' -> 0, ‘93 years and over' -> 93, and ‘100 years and over' -> 100).
- SEX was changed to SEXMALE, with 1 indicating males and 0 indicating females.
- The ARRTIME variable gave the time of initial presentation to the ED in the form XX:XX [24-hr time]. This was converted to ARRTIMEMIN, which represented the minutes after midnight.
- For vital signs, the values were converted to integers or floats as appropriate.
- Each comorbid is in a separate column and were originally recorded as ‘Yes’ or ‘No’. These variables were each converted to 1 (‘Yes’) or 0 (‘No’).

Outcome variable

The outcome of interest was hospital admission. For the purposes of this study, a patient was considered to be admitted if they were admitted to an in-patient ward, admitted to an observation area, transferred to another hospital, or transferred to a psychiatric facility. These outcomes were in separate columns in the initial data (ADMITHOS, OBSHOS, TRANSOTH, and TRANSPSYCH). We therefore created a composite variable called ADM_OUTCOME that was positive (1) if any of those outcome variables were recorded as ‘Yes’.

Feature Engineering

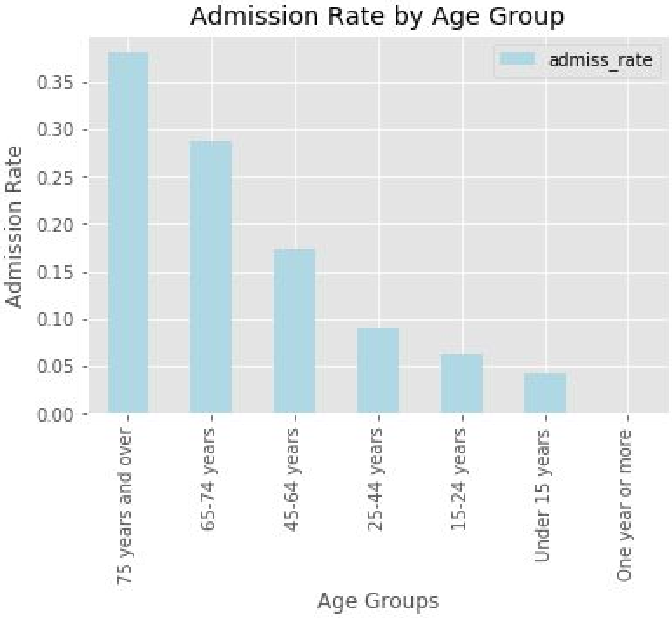
We used historical data (2007-2013) to determine the average admission rate based on the primary reason for visit (RFV1). Averages were only computed for reason for visits that occurred more than five times. These average rates of admission were then joined to the 2014-2017 data based on RFV1. The baseline admission rate was used for patients where the values of RFV1 did not appear in the historical data at least five times. These predicted admission rates based on RFV1 were stored as RFV1_admit_rate.

Data splitting / sampling

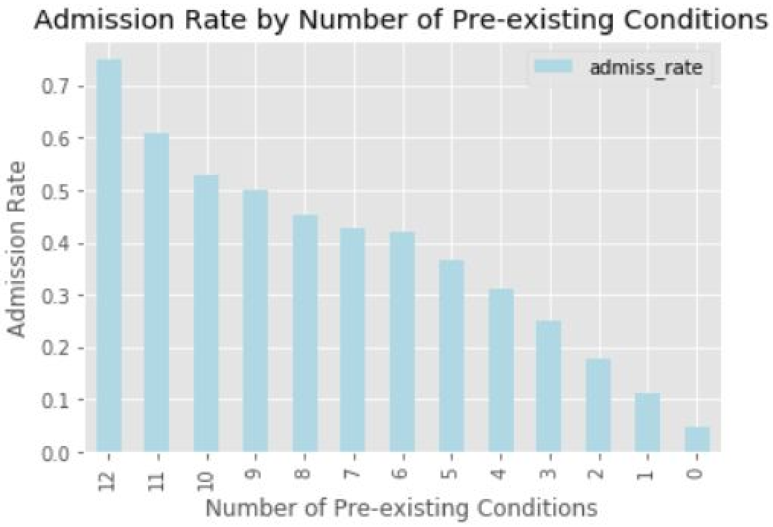
The data was randomly separated into training and testing data using a 80%/20% split. The splitting was performed using PySpark function ‘randomSplit’. The same seed was used when training and testing each model to ensure uniformity. Cross-validation with K=5 with the training data to tune hyperparameters.

Exploratory data analysis

Our outcome of interest was which patients will be admitted to the hospital, so we plotted the percentage of patients admitted by age group.



The plot shows that there is a strong correlation between age and rate of admission with those 75 years and older having over a 35% chance of being admitted. We also plotted the rate of admission based on the number of underlying health conditions.



This shows that the number of pre-existing health conditions is also closely tied to admission rate.

Model construction

For all models, a pipeline was constructed and then fit on the training data. The model from the pipeline was then used to make predictions on the testing data. Model metrics were then assessed based on these predictions.

Logistic Regression (base model)

Our base model used only the age of the patient, sex, and total number of pre-existing health problems in a logistic regression model. The base model was initially produced without adjusting for class imbalance in our data. The data was not scaled since this does not typically benefit logistic regression. A second model was then evaluated using the class weighting feature in the pyspark.ml.classification.LogisticRegression package. The classes were weighted by the inverse of portion of the class occurrences. For example, cases with which the patient was admitted were weighted as 0.9 if the admission rate was 10%.

Logistic Regression (all variables)

The next model we tested included patient variables and engineered features. This added flags for individual health conditions, place of residence, vital signs, arrival time and date, injury data, reason for visit, and historical admission rate of reason for visit. Weighted outcomes were used to compensate for the class imbalance.

Ridge and Lasso Regression

Ridge and lasso regression were then performed separately using all variables. These used the same pyspark.ml.classification.LogisticRegression package that we used for logistic regression, however the elasticNetParameter was changed. The data was scaled before ridge and lasso regressions. Cross-validation using the training data was used to tune the regularization hyperparameter with K=5. The area under the receiver operator curve (AUROC) was used to determine the best regularization parameter. The best model was then evaluated using the testing data.

Random Forest

We then build random forest models using all variables with the pyspark.ml.classification.RandomForestClassifier package. Cross-validation was used to determine the optimal number of trees and tree depth. Unfortunately, outcome weighing was not available in the version of PySpark we are using (version: 2.4.5). We found very few positive predictions, therefore down-sampling of negatives was implemented in the training data. After down-sampling there were an equal number of positives and negatives for training. The testing data remained imbalanced and in its original form.

Linear Support Vector Machine

Finally, we built a linear support vector machine(SVM) model using all variables with the pyspark.ml.classification.LinearSVC package. Cross-validation was used to find the optimal regularization parameter. Due to the time to train each model, only 10% of the data was used for initial tuning. The outcome weighting was used to compensate for the class imbalance in the outcome.

Model evaluation

Each of the models was evaluated using a confusion table, accuracy, precision, recall, F1 score, sensitivity, specificity, and AUROC. AUROC was chosen to be the primary evaluation metric since it is used most commonly in the medical literature. The F1 score was used as the secondary evaluation metric. Accuracy was considered to be the least significant evaluation metric because of the class imbalance of the outcome. Training time was recorded in seconds and was the time needed to train the model after the best hyperparameters were found. This does not take into account the time needed for tuning.

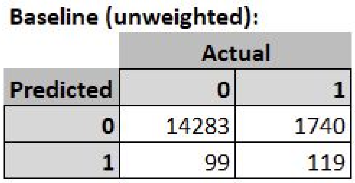
RESULTS

There were 305,897 patients in the NHAMCS data 2007-2017. When we narrowed our window to 2014-2017 to ensure all the comorbidity data was present, we were left with 81,081 patient observations. Of these patients, 9,308 were admitted (11.5% admission rate). The evaluation metrics followed by the confusion matrices are shown for each model.

Comparison of Model Performance

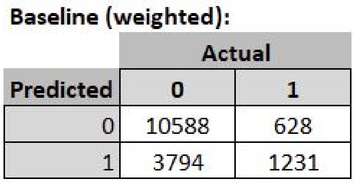
	Baseline	**Baseline (weighted)**	**All vars LR**	**Optimized Ridge**	**Optimized Lasso**	**Optimized RF**	**Optimized SVM**
Model Type	Logistic regression	Logistic regression	Logistic regression	Ridge regression	Lasso regression	Random forest	Linear Support Vector Machine
Variables	Age, Sex, Num of conditions	Age, Sex, Num of conditions	All	All	All	All	All
Hyper-parameters	None	None	None	Regulariz-ation	Regulariz-ation	Num trees, Max depth	Regulariz-ation
Optimal param value	N/A	N/A	N/A	0.001	0.0001	Trees=200, Depth=10	0.0
Class Imbalance comp	None	Outcome weighting	Outcome weighting	Outcome weighting	Outcome weighting	Down-sampled negatives	Outcome weighting
Accuracy	0.887	0.723	0.770	0.770	0.774	0.759	0.777
Precision	0.546	0.245	0.295	0.295	0.298	0.288	0.299
Recall	0.064	0.662	0.724	0.726	0.724	0.752	0.708
F1	0.115	0.358	0.419	0.420	0.423	0.416	0.421
Sensitivity	0.064	0.662	0.724	0.726	0.724	0.752	0.708
Specificity	0.891	0.944	0.956	0.956	0.956	0.959	0.954
AUC	0.749	0.750	0.826	0.826	0.825	0.834	0.825
Training time (sec)	12	12	34	163	176	444	1127

Confusion matrices



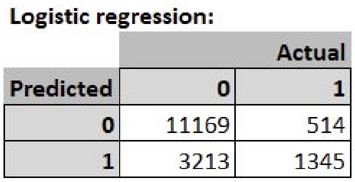
Baseline (unweighted):

	Actual	
Predicted	0	1
0	14283	1740
1	99	119



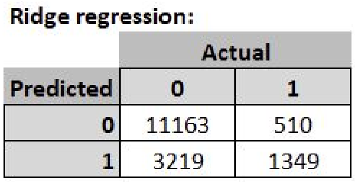
Baseline (weighted):

	Actual	
Predicted	0	1
0	10588	628
1	3794	1231



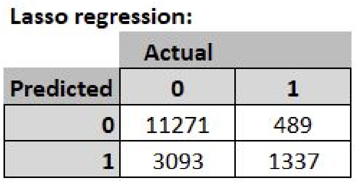
Logistic regression:

	Actual	
Predicted	0	1
0	11169	514
1	3213	1345



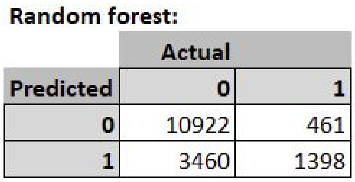
Ridge regression:

	Actual	
Predicted	0	1
0	11163	510
1	3219	1349



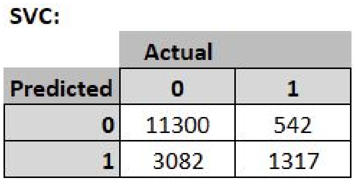
Lasso regression:

	Actual	
Predicted	0	1
0	11271	489
1	3093	1337



Random forest:

	Actual	
Predicted	0	1
0	10922	461
1	3460	1398



SVC:

	Actual	
Predicted	0	1
0	11300	542
1	3082	1317

DISCUSSION

This project showed that it is feasible to predict the need for hospital admission in patients presenting to the Emergency Department using only data available at the time of triage. We found that the inclusion of more detailed information significantly improved the accuracy of our predictions. When the complete set of variables was included, all of the models performed approximately equivalently based on AUROC and F1 score. The models showed good specificity (ability to correctly classify patients who were discharged) and moderate sensitivity (classifying patients who were admitted).

We found the performance of logistic, lasso, and ridge regression models to be almost identical on several different metrics. We used cross-validation to tune the regularization parameters, which found that very small regularization parameters performed the best. This explains why all these models had similar metrics, since lasso and ridge are approximately the same as logistic regression when the regularization parameter approaches zero. It also indicates that all of the variables contributed to the accuracy of predictions, since ridge and lasso rely on shrinkage.

Our tests showed that compensating for the class imbalance in the data was extremely important. The accuracy was actually the highest in our baseline model without outcome weighting, but this was because of the high number of true negatives since most of the predictions were negative. The regression and SVC functions had a parameter to include outcome weights. This weighing significantly improved the F1 score of our baseline model, while keeping the AUROC approximate the same. The random forest function did not have outcome weighting in the version of PySpark we are using, so we instead down-sampled the negatives in the training data. This also worked well to compensate for the class imbalance, but we were training with a reduced data set. It would be interesting to test the performance of random forest in PySpark 3.0 because it now includes outcome weighing.

Based on our results, all of the methods appear to be appropriately predicting hospital admission. If we had to choose one model, we would select random forest as our champion because it had a slightly higher AUROC.

Future Work

The next steps for this project are to perform additional feature engineering, experiment using additional models, and test on prospectively collected data.

Adding infection severity scores as features could improve the results. These include the systemic inflammatory response syndrome (SIRS) and sequential organ failure assessment (SOFA) scores. Additionally, we only used the primary reason for visit (RFV1) for prediction. We could try to combine RFV2-RFV5 using a naive bayesian approach to improve that feature.

It would also be interesting to experiment with other models. Deep learning has been described using PySpark that might be able to take advantage of additional interactions between variables. Another possibility would be to use an ensemble method which combined the prediction from the models that we previously trained.

Before using this tool clinically, it would need to be prospectively validated. This could be performed on subsequent years of data from NHAMCS when they become available. However, it is possible that although these models perform well on aggregate data, they might not perform well at individual hospitals since patient distributions are not equal. For instance, the University of Virginia Hospital sees much sicker patients than average and the admission rate is ~20%. It may be that individual models would be needed for each hospital or class of hospital (academic/community, urban/rural, etc.).

CONCLUSIONS

The project used PySpark to build a model which predicts the need for hospital admission in patients presenting to the Emergency Department using data from a national database. We demonstrated feasibility of predicting admission using only data available at the time of triage. We found that logistic regression, ridge regression, lasso regression, random forest, and linear support vector machine models all performed similarly. The next steps in this work are to perform additional feature engineering, model optimization, and prospective validation.