

Hospital Readmission Reduction Application

Team 5

Dipti Pamnani, Jayesh Samyani, Suraj Sharma

December 16, 2016

Abstract

- Alarmingly high risk of readmission in the US.
- Hospital readmissions are expensive and reflect the inadequacies in healthcare system.
- Early identification of patients facing a high risk of readmission can enable healthcare providers to conduct additional investigations and possibly prevent future readmissions.
- This not only improves the quality of care but also reduces the medical expenses on readmission.
- Florida, Texas, California, New-York have the highest readmissions. The medical care provided by the hospitals in these states should thus be improvised by identifying and targeting patients at high risk of readmission.

Introduction

Healthcare is the most critical aspect for anyone across the globe. Health of every individual is hence very important not only for them but also for hospitals, which take charge of curing them. Emergency readmission to hospital is frequently used as a measure of the quality of a hospital because a high proportion of readmissions should be preventable if the preceding care is adequate. Hospitals are required to maintain rates of readmission below a national standard, else they face sanctions and fines. We seek to predict whether a patient discharged from hospital will return back as an emergency admit. We can also estimate the penalty, which is imposed on hospitals if they have more readmission than certain threshold.

Thus, hospitals can take better care for patients which are likely for readmission and save the penalty amount.

Data Description:**Data Source:**

- The data is from the Center for Clinical and Translational Research, Virginia Commonwealth University. This data had been prepared to analyze factors related to readmission as well as other outcomes pertaining to patients with diabetes.

Data Set Information:

- The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks.
- It includes 101,766 instances and over 50 features representing patient and hospital outcomes.
- Information was extracted from the database for encounters that satisfied the following criteria.
- It is an inpatient encounter (a hospital admission).
- It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.
- The length of stay was at least 1 day and at most 14 days.
- Laboratory tests were performed during the encounter.
- Medications were administered during the encounter.
- Attribute Information:

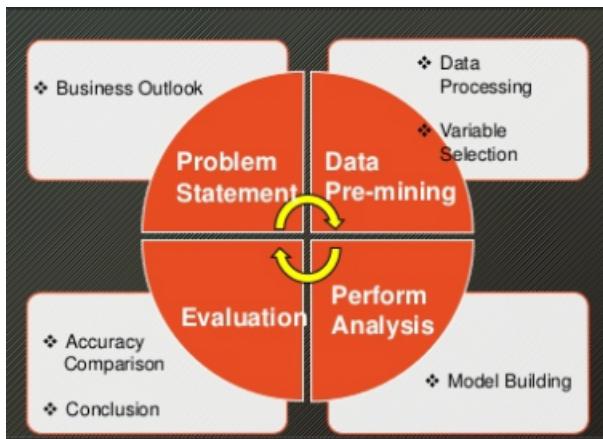
Attribute	Description
Encounter ID	Unique identifier of an encounter
Patient number	Unique identifier of a patient
Race	Values: Caucasian, Asian, African American, Hispanic, and other
Gender	Values: male, female, and unknown/invalid
Age	Grouped in 10-year intervals: 0, 10), 10, 20), ..., 90, 100)
Weight	Weight in pounds.
Admission type	Emergency, urgent, elective, newborn, and not available,etc.
Discharge disposition	Discharged to home, expired, and not available,etc.
Admission source	Physician referral, emergency room, and transfer from a hospital,etc.
Time in hospital	Integer number of days between admission and discharge
Payer code	Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare,

	and self-pay
Medical specialty	Cardiology, internal medicine, family/general practice, and surgeon,etc.
Number of lab procedures	Number of lab tests performed during the encounter
Number of procedures	Number of procedures (other than lab tests) performed during the encounter
Number of medications	Number of distinct generic names administered during the encounter
Number of outpatient visits	Number of outpatient visits of the patient in the year preceding the encounter
Number of emergency visits	Number of emergency visits of the patient in the year preceding the encounter
Number of inpatient visits	Number of inpatient visits of the patient in the year preceding the encounter
Diagnosis 1	The primary diagnosis (coded as first three digits of ICD9); 848 distinct values
Diagnosis 2	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values

Diagnosis 3	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values
Number of diagnoses	Number of diagnoses entered to the system
Glucose serum test result	Indicates the range of the result or if the test was not taken. Values: ">200," ">300," "normal," and "none" if not measured
A1c test result	Indicates the range of the result or if the test was not taken. Values: ">8" if the result was greater than 8%, ">7" if the result was greater than 7% but less than 8%, "normal" if the result was less than 7%, and "none" if not measured.
Change of medications	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: "change" and "no change"
Diabetes medications	Indicates if there was any diabetic medication prescribed. Values: "yes" and "no"
24 features for medications	The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: "up" if the dosage was increased during the encounter, "down" if the dosage was decreased, "steady" if the dosage did

	not change, and “no” if the drug was not prescribed
Readmitted	Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission.
Age Group	Young(0-20), Adult(21-50), Old(51+)
Readmission Status	Yes if readmitted (>30 or <30 days) No if not readmitted
Cost of all procedures	Cost for all the procedures performed for that encounter
Cost of medications	Cost for medication for that encounter
Total cost of Treatment	Total cost of treatment for that encounter

Methodology:



Problem Statement:

- Identify the major factors that contribute to hospital readmission.
- Predict number of readmissions and help reduce the cost of readmission in the US.

Steps:

Data Pre-mining (Data Processing and Variable Selection)

- Categorize Age field into 3 groups (Young, Adult, and Old)
- Re-categorize readmission group (<30, >30 days and No) to 2 groups (Readmitted and Not readmitted)
- Mapping the numeric ids for various fields with their actual description.
- Appending the cost factor to the data so as to analyze the cost of readmission.
- Review each variable's relationship with Readmission.
- Remove irrelevant variables (example: Payer code, Weight, etc.)
- Replace missing values with NULL.

1. Data Ingestion and Wrangling:

- We have **two** input files **diabetic_data.csv** and **IDs_mapping.csv** from the dataset, we found on:
<https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008#>
- We read both csv files and then merged into 1 csv file based on mapping of **Admission type Id**, **Admission Source Id** and **Discharge Dispoistion ID**.
- Categorize Age field into 3 groups (Young, Adult, and Old)
- Re-categorize readmission group (<30, >30 days and No) to 2 groups (Readmitted and Not readmitted)
- Mapping the numeric ids for various fields with their actual description.
- Appending the cost factor to the data so as to analyze the cost of readmission.
- Review each variable's relationship with Readmission.
- Remove irrelevant variables (example: Payer code, Weight, etc.)
- Replace missing values with NULL.

2. Modeling:

- We have a column named readmitted in our dataset which specifies whether the encounter id is readmitted <30(less than days) or >30(after 30 days) or Not.
- We have created readmission status which takes <30 and >30 values of readmitted and gives YES as output value and NO if the readmitted value is NO. Readmission status column has been used to train the classification algorithm.
- Once dataset is ingested and cleaned. We can perform feature selection step.

2.1 Feature Selection

- Using regsubsets function and glm , we came up with 9 most important features which can be used to classify readmission status.

Code:

```
- regfit.fwd=regsubsets(tempdiabdata$readmissionstatus~.,  
                      data=tempdiabdata, nvmax=45,method="forward")  
- F=summary(regfit.fwd)  
- names(F)  
- F$rss  
- F$adjr2  
- coef(regfit.fwd,35)
```

we have used **forward** and **backward** selection method in regsubsets.

- Here are the 9 features

1. Admission type
2. Admission source
3. Discharge disposition
4. Diag1
5. Insulin
6. Number emergency
7. Number inpatient
8. Race
9. Medical Specialty

Following are the screenshots of regsubsets feature selection

The screenshot shows the RStudio interface with several tabs open: Untitled1, k-means.R, initial_clean.R, tempdiabdata, Logistic-Regression.R, data.frame.test, and data.frame.train. The R Script pane displays R code for data cleaning and preparation. The Environment pane shows variables like diabdata (101766 obs.), tempdiabdata (101766 obs.), and values for admissionsource, admissiontype, dischargedisp, and raceCaucasian. The Global Environment pane lists objects like eCap, F, i, max, min, regfit.fwd, remDr, temp, and val. The Help pane shows the glm function documentation. The bottom status bar indicates the date and time as Mon 3:21 PM.

```

201
202
203 tempdiabdata$encounter_id <- as.numeric(tempdiabdata$encounter_id)
204 tempdiabdata$patient_nbr <- as.numeric(tempdiabdata$patient_nbr)
205 tempdiabdata$time_in_hospital <- as.numeric(tempdiabdata$time_in_hospital)
206 tempdiabdata$num_lab_procedures <- as.numeric(tempdiabdata$num_lab_procedures)
207 tempdiabdata$num_procedures <- as.numeric(tempdiabdata$num_procedures)
208 tempdiabdata$num_medications <- as.numeric(tempdiabdata$num_medications)
209 tempdiabdata$num_outpatient <- as.numeric(tempdiabdata$num_outpatient)
210 tempdiabdata$num_emergency <- as.numeric(tempdiabdata$num_emergency)
211 tempdiabdata$num_inpatient <- as.numeric(tempdiabdata$num_inpatient)
212
213 # Create a new column to indicate if the patient was transferred to a federal health care facility
214 tempdiabdata$discharge_disposition_idDischarged/transferred_to_a_federal_health_care_facility <- ifelse(tempdiabdata$discharge_disposition_id == 10, 1, 0)
215
216 # Create a new column to indicate if the admission source is Court/Law Enforcement
217 tempdiabdata$admission_source_idCourt/Law Enforcement <- ifelse(tempdiabdata$admission_source_id == 1, 1, 0)
218
219 # Create a new column to indicate if the admission source is Transfer from critial access hospital
220 tempdiabdata$admission_source_idTransfer_from_critial_access_hospital <- ifelse(tempdiabdata$admission_source_id == 2, 1, 0)
221
222 # Create a new column to indicate if the admission source is Normal Delivery
223 tempdiabdata$admission_source_idNormal_Delivery <- ifelse(tempdiabdata$admission_source_id == 3, 1, 0)
224
225 # Save the cleaned dataset
226 save(tempdiabdata, file = "dataset_diabetes.RData")

```

> coef(regfit.fwd, 35)

	(Intercept)	encounter_id	-2.570383e-10	raceCaucasian	4.072802e-02	admission_type_idElective	-5.692982e-02	admission_type_idNULL	4.461000e-02	dischargeDisposition_idExpired	-4.777039e-01	dischargeDisposition_idNULL	-7.583264e-02	dischargeDisposition_idDischarged/transferred to a federal health care facility.	-2.665416e-01	admissionSource_idCourt/Law Enforcement	-1.280504e-01	admissionSource_idNot Available	-2.298245e-01	admissionSource_idTransfer from critial access hospital	-2.180162e-01	admissionSource_idNormal Delivery	-4.051074e-01
--	-------------	--------------	---------------	---------------	--------------	---------------------------	---------------	-----------------------	--------------	--------------------------------	---------------	-----------------------------	---------------	--	---------------	---	---------------	---------------------------------	---------------	---	---------------	-----------------------------------	---------------

The screenshot shows an RStudio interface with the following details:

- File Menu:** RStudio, File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Window, Help.
- Toolbar:** Go to file/folder, Addins -.
- Left Panel:** Untitled1, K-means.R, initial_clean.R, tempdiabdata, Logistic-Regression.R, data.frame.test, data.frame.train.
- Right Panel:**
 - Sharing Screenshot:** A link to your screenshot has been copied to your clipboard.
 - Data:**
 - tempdiabdata: 101766 obs. of 51 variables
 - tempdiabdata: 101766 obs. of 50 variables
 - Values:**
 - admissionsource: chr [1:25] "Physician Referral" "Clinic..."
 - admissiontype: chr [1:8] "Emergency" "Urgent" "Elective"
 - dischargedisp: chr [1:30] "Discharged to home" "Discha..."
 - eCap: List of 1
 - F: Large summary.regsubsets (8 elements, 24.4 Mb)
 - i: 101767
 - max: "90"
 - min: "80"
 - regfit.fwd: Large regsubsets (29 elements, 24.4 Mb)
 - remDr: Environment
 - temp: "80-90"
 - val: 3L
- Console:**

```

201
202
203 tempdiabdata$encounter_id <- as.numeric(tempdiabdata$encounter_id)
204 tempdiabdata$patient_nbr <- as.numeric(tempdiabdata$patient_nbr)
205 tempdiabdata$time_in_hospital <- as.numeric(tempdiabdata$time_in_hospital)
206 tempdiabdata$num_lab_procedures <- as.numeric(tempdiabdata$num_lab_procedures)
207 tempdiabdata$num_procedures <- as.numeric(tempdiabdata$num_procedures)
208 tempdiabdata$num_medications <- as.numeric(tempdiabdata$num_medications)
209 tempdiabdata$number_outpatient <- as.numeric(tempdiabdata$number_outpatient)
210 tempdiabdata$number_emergency <- as.numeric(tempdiabdata$number_emergency)
211 tempdiabdata$number_inpatient <- as.numeric(tempdiabdata$number_inpatient)
224:1 (No Level)

Console ~/Desktop/ADS/Final Project/dataset_diabetes/ >
      diag_1136
      -2.652553e-01
      diag_1142
      7.437774e-02
      diag_2600
      5.442330e-02
      diag_3601
      -1.282983e-01
      chlorpropamideSteady
      -2.466148e-02
      glimepirideNo
      -8.962739e-03
      acetohexamideSteady
      4.996433e-01
      discharge_disposition_idDischarged/transferred to a Critical Access Hospital (CAH).
      0.000000e+00
      discharge_disposition_idDischarged/transferred to another Type of Health Care Institution not Defined Elsewhere
      0.000000e+00
      diag_3930
      -4.327641e-01
      diag_3E861
      -5.107205e-01
  
```
- Message Area:**

Warning message:
In log(vr) : NaNs produced
- Bottom Bar:** Shows various application icons.

Code:

```
fit1 <- glm(readmissionstatus ~., data = data.frame.train,family = binomial(link="logit"))
```

```
data.frame.test$predict <- predict(fit1,newdata=data.frame.test)
```

Following are the screenshots of glm feature selection

```

admission_source_id Not Available
admission_source_id Transfer from critial access hospital
admission_source_idNormal Delivery
admission_source_id Sick Baby
admission_source_id Extramural Birth
admission_source_idNULL
admission_source_id Not Mapped
admission_source_id Transfer from hospital inpt/same fac reslt in a sep claim
admission_source_id Transfer from Ambulatory Surgery Center
time_in_hospital
num_lab_procedures
num_procedures
num_medications
number_outpatient
number_emergency
number_inpatient
number_diagnoses
max_glu_serum>300
max_glu_serumNone
max_glu_serumNorm
glipizideNo
glipizideSteady
glipizideUp
insulinNo
insulinSteady
insulinUp
changeNo
diabetesMedYes
AgeGroupOld
AgeGroupYoung
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

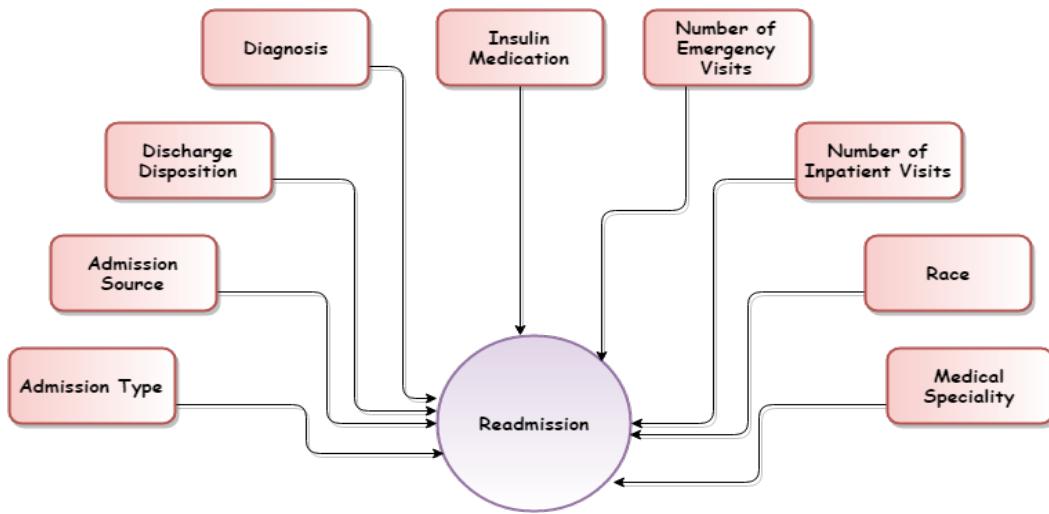
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 48618  on 69369  degrees of freedom
Residual deviance: 45890  on 69298  degrees of freedom

```

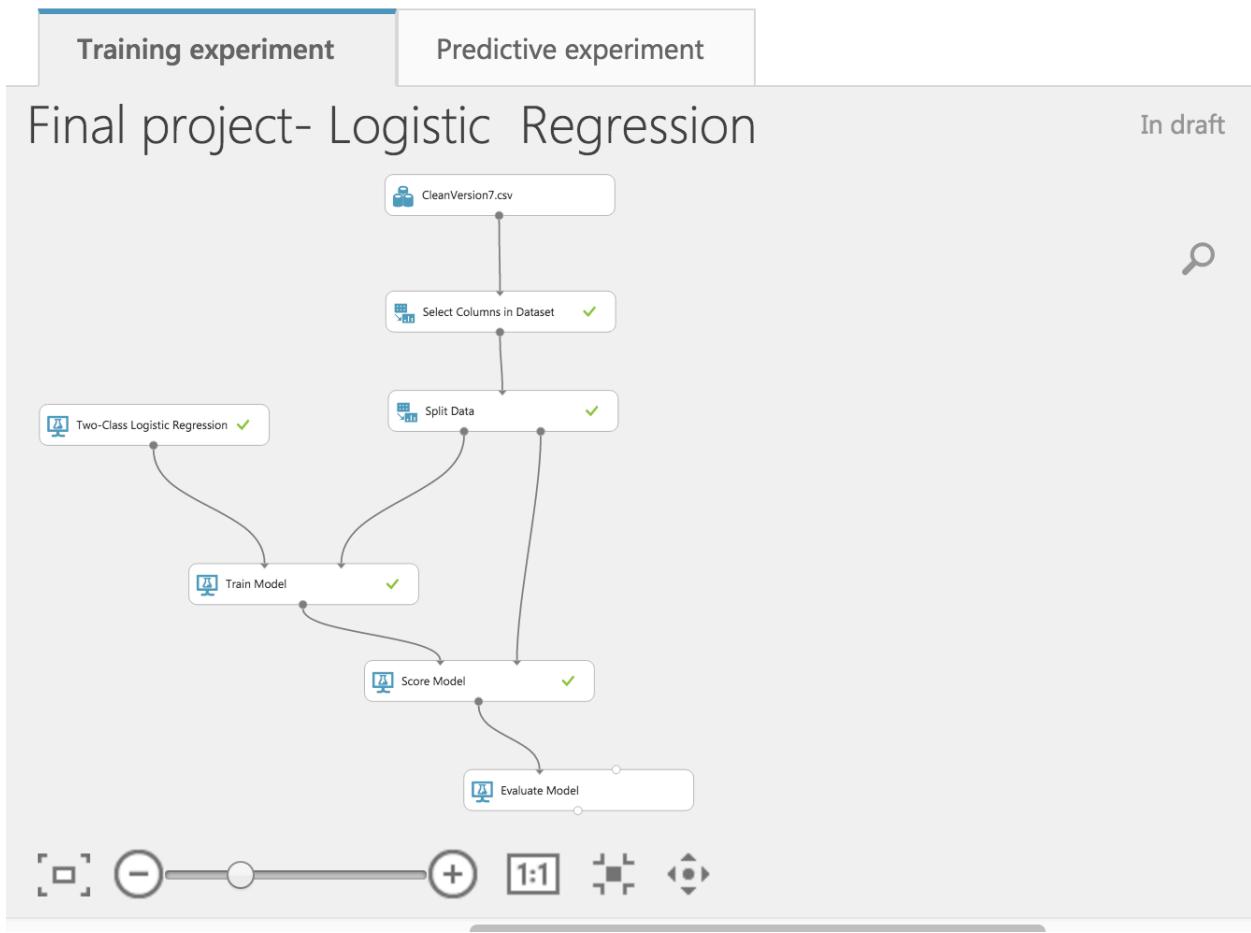
Perform Analysis

- Input variables:
- Total sample size:
- Partition:
- Build models from:

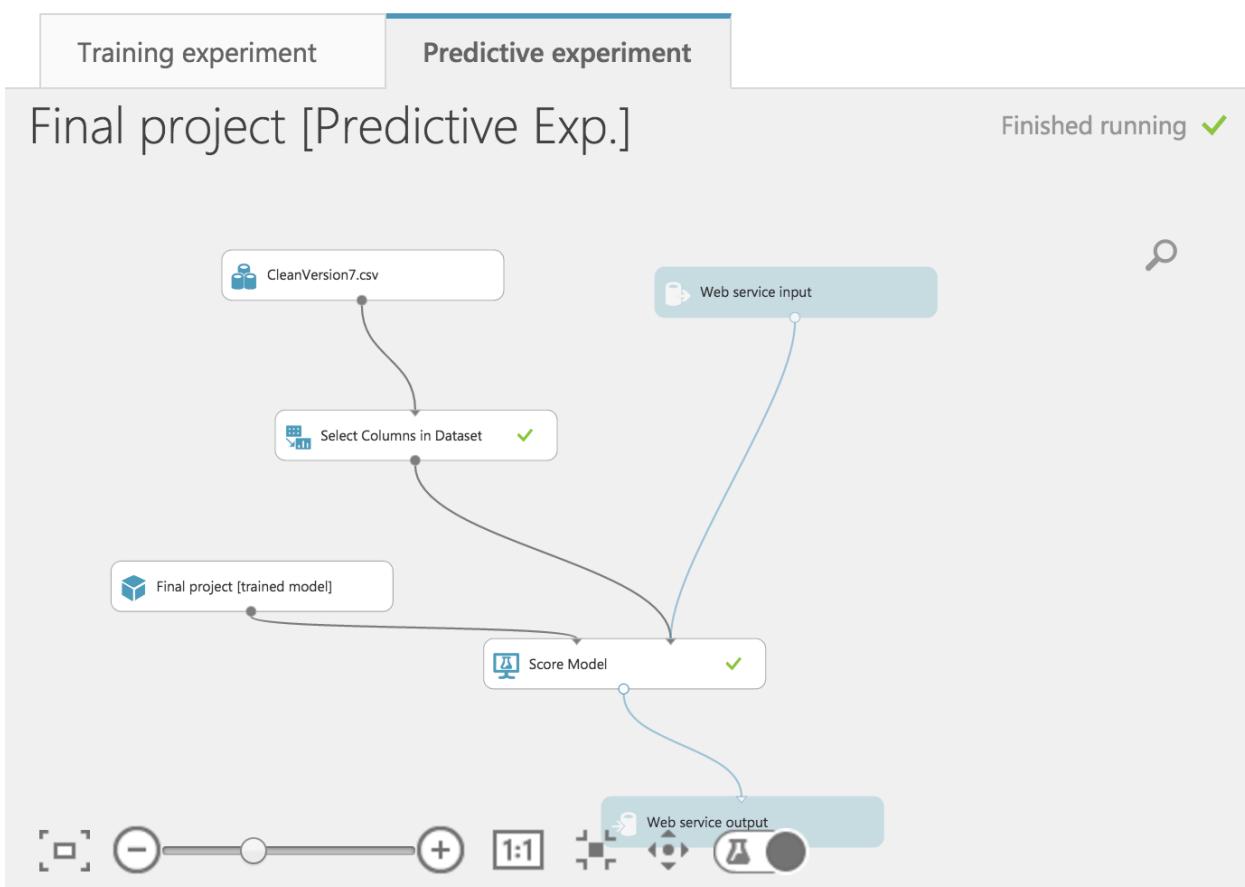


CLASSIFICATION:

Logistic regression

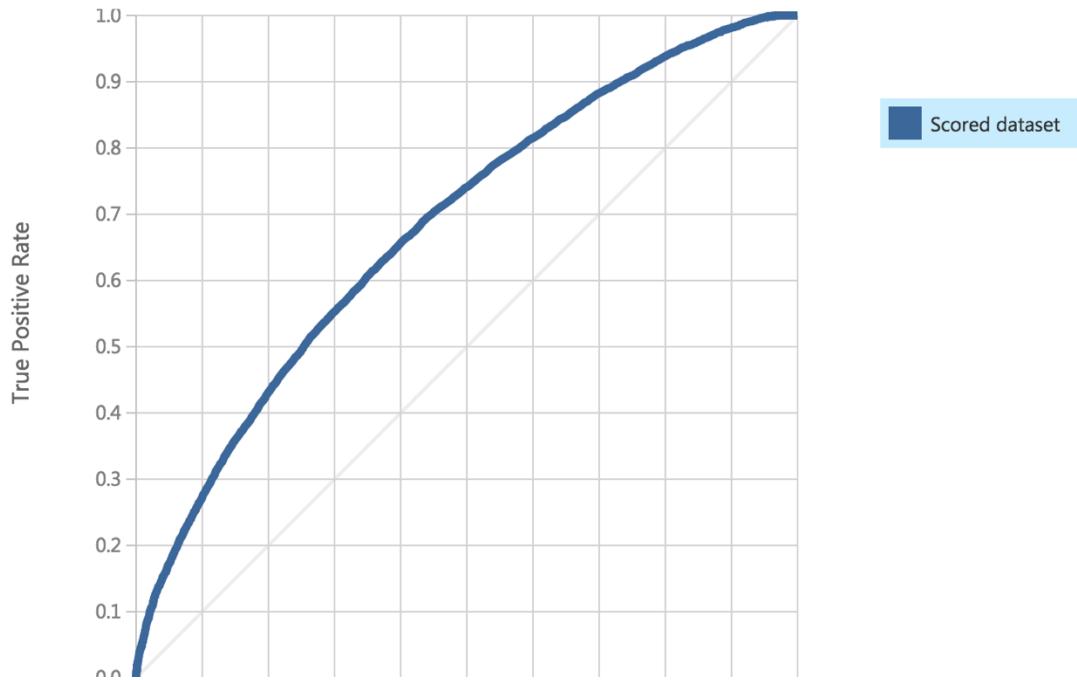


Model deployed on web:



ROC Curve and Accuracy :

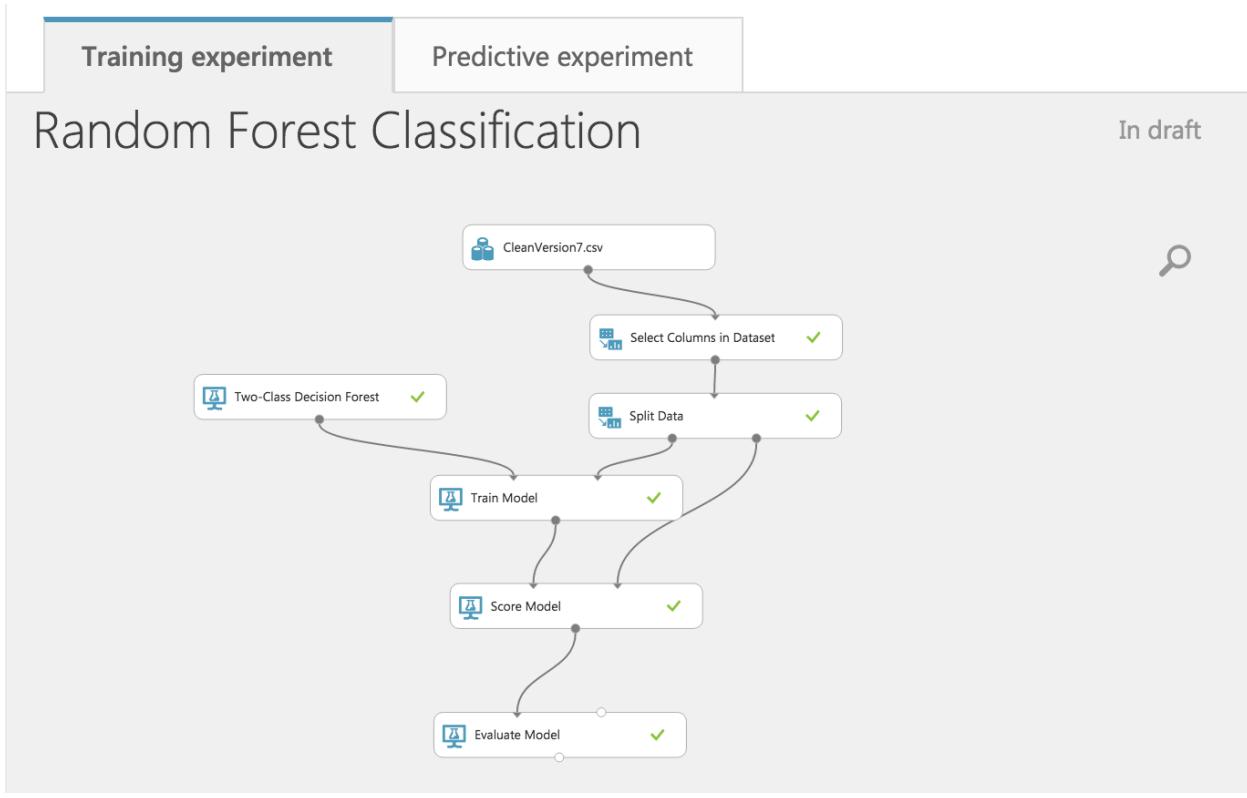
Final project- Logistic Regression ➤ Evaluate Model ➤ Evaluation results



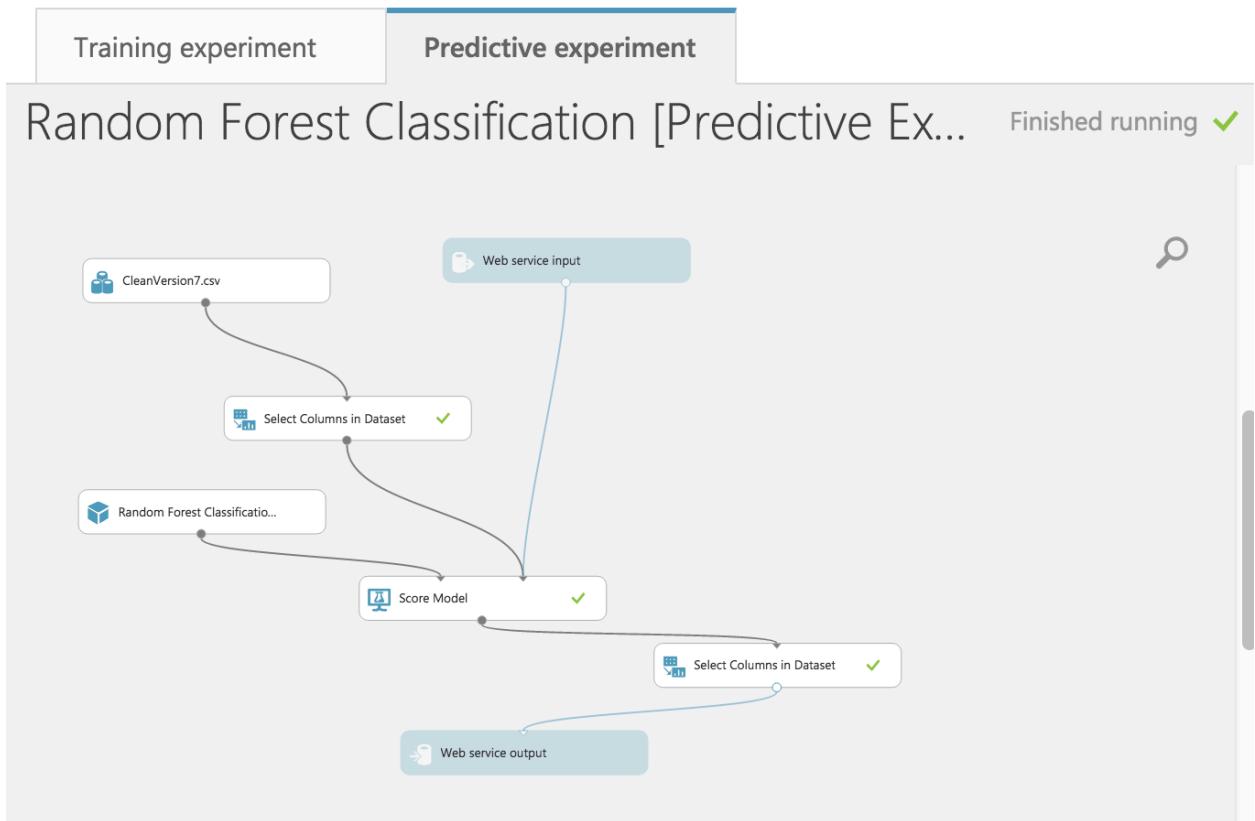
Final project- Logistic Regression ➤ Evaluate Model ➤ Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
5658	6019	0.633	0.631	0.5	0.682
False Positive	True Negative	Recall	F1 Score		
3315	10449	0.485	0.548		
Positive Label	Negative Label				
YES	NO				

Random Forest

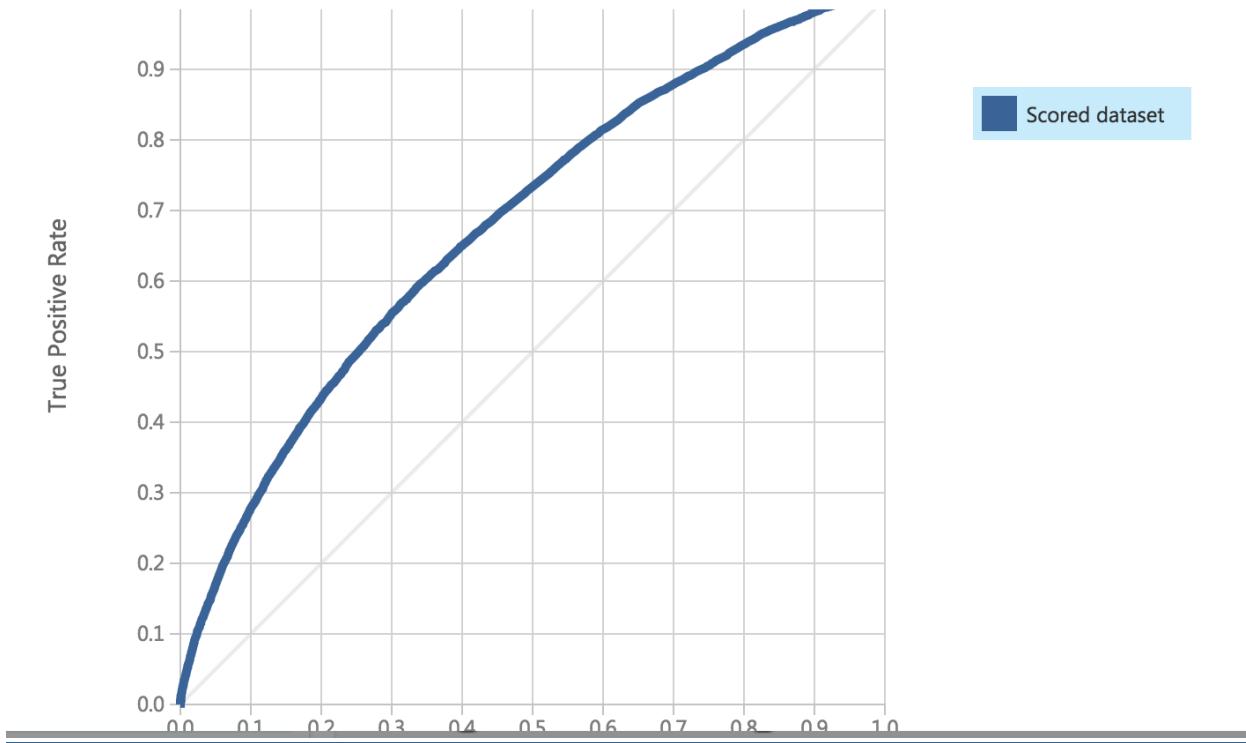


- Model deployed on web:



ROC and Accuracy:

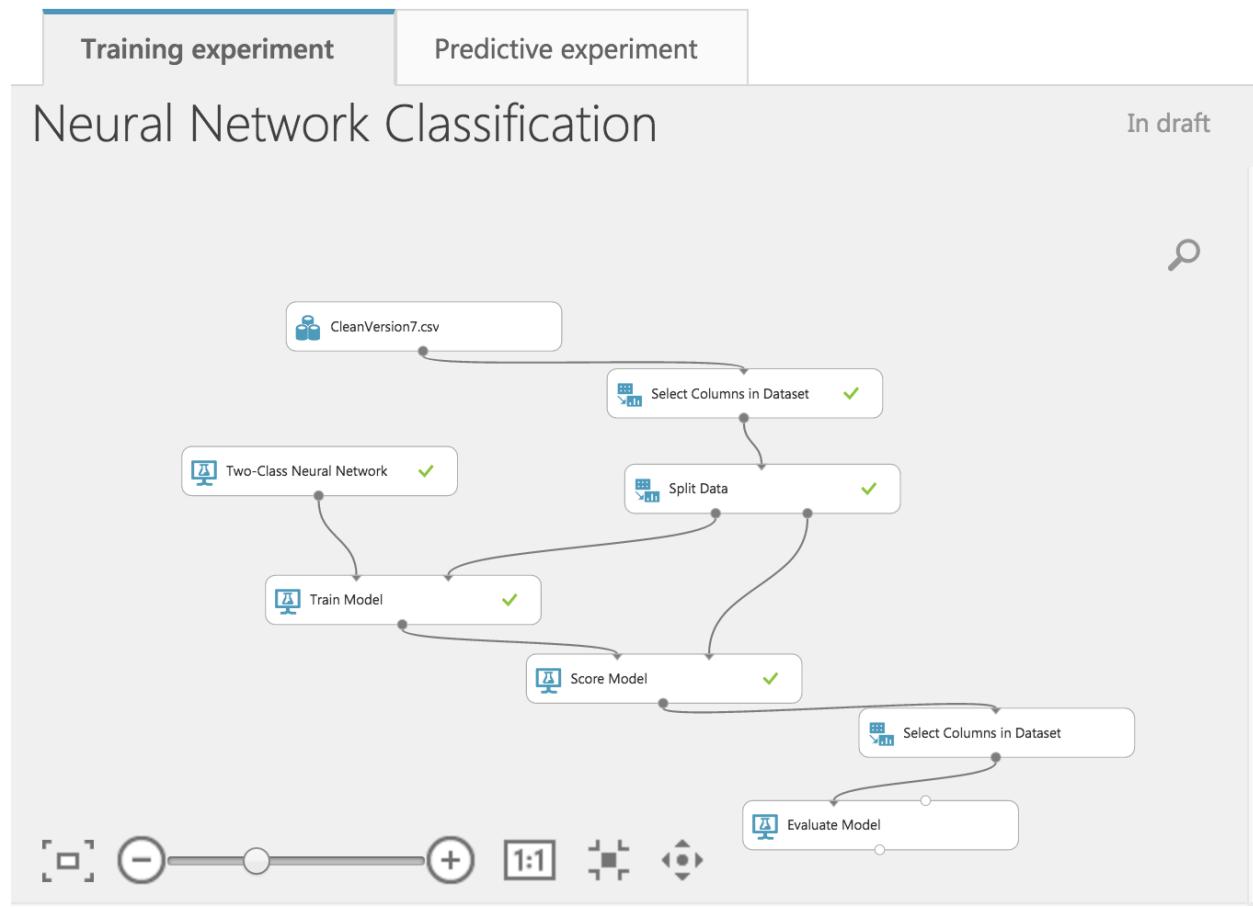
Random Forest Classification ➤ Evaluate Model ➤ Evaluation results



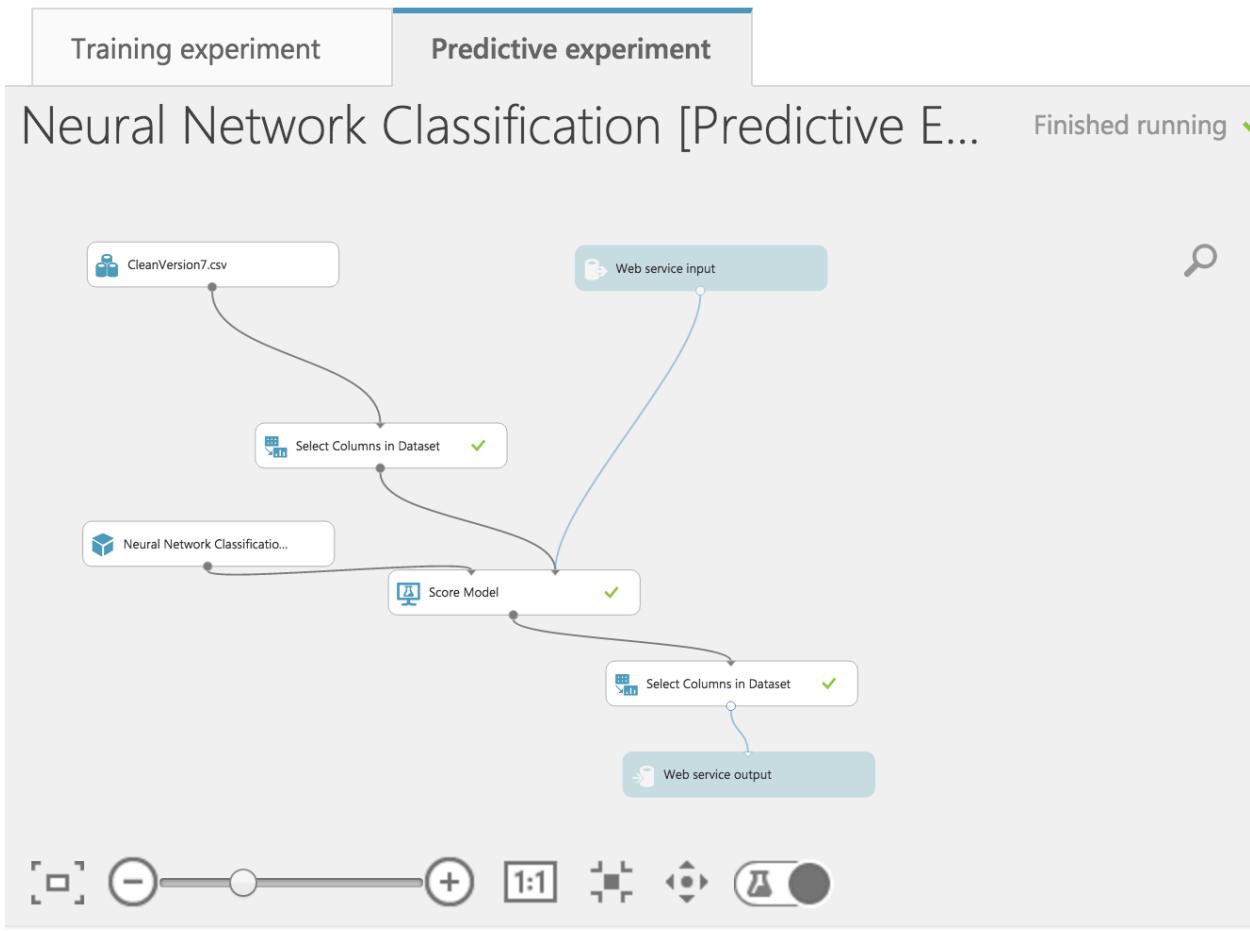
Random Forest Classification ➤ Evaluate Model ➤ Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
7194	6899	0.633	0.625	0.5	0.680
False Positive	True Negative	Recall	F1 Score		
4313	12124	0.510	0.562		
Positive Label	Negative Label				
YES	NO				

Neural Networks:

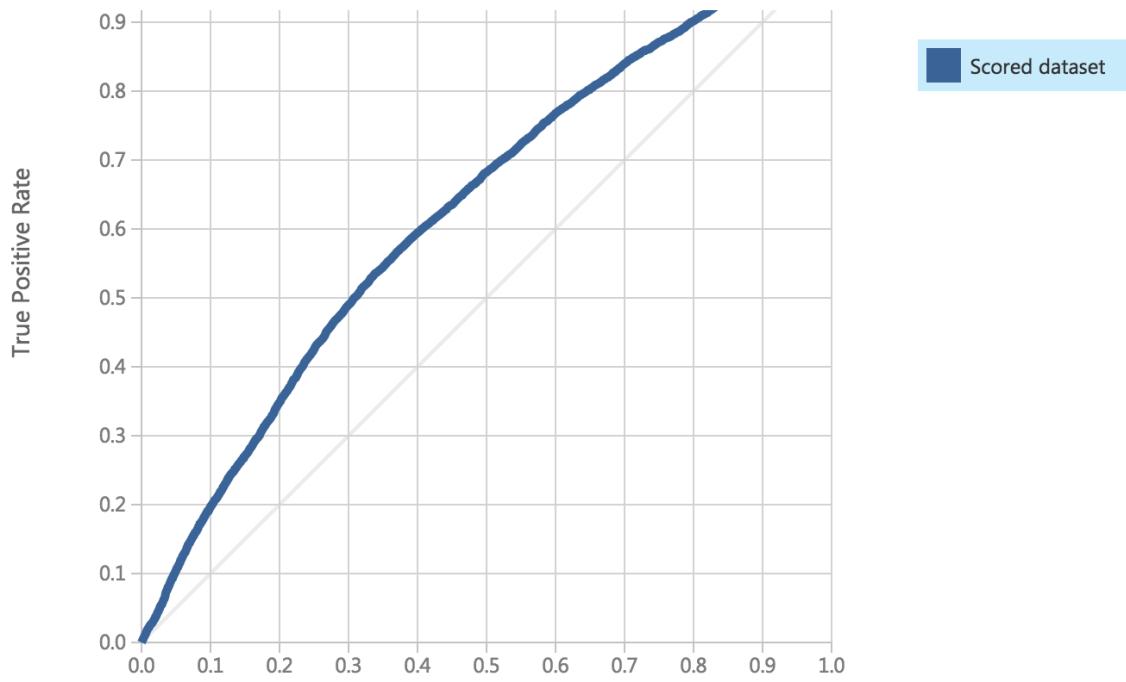


- **Model deployed on web:**



ROC Curve and Accuracy:

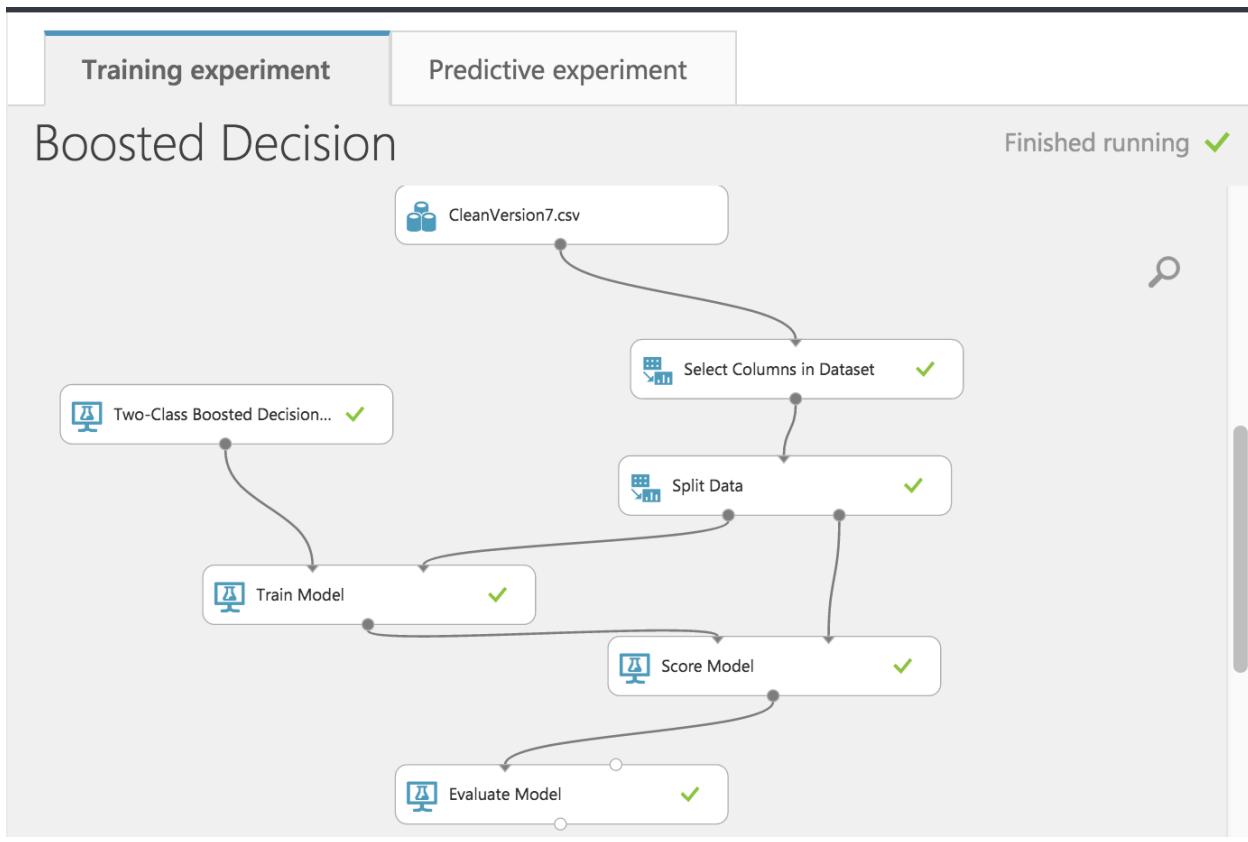
Neural Network Classification ➤ Evaluate Model ➤ Evaluation results



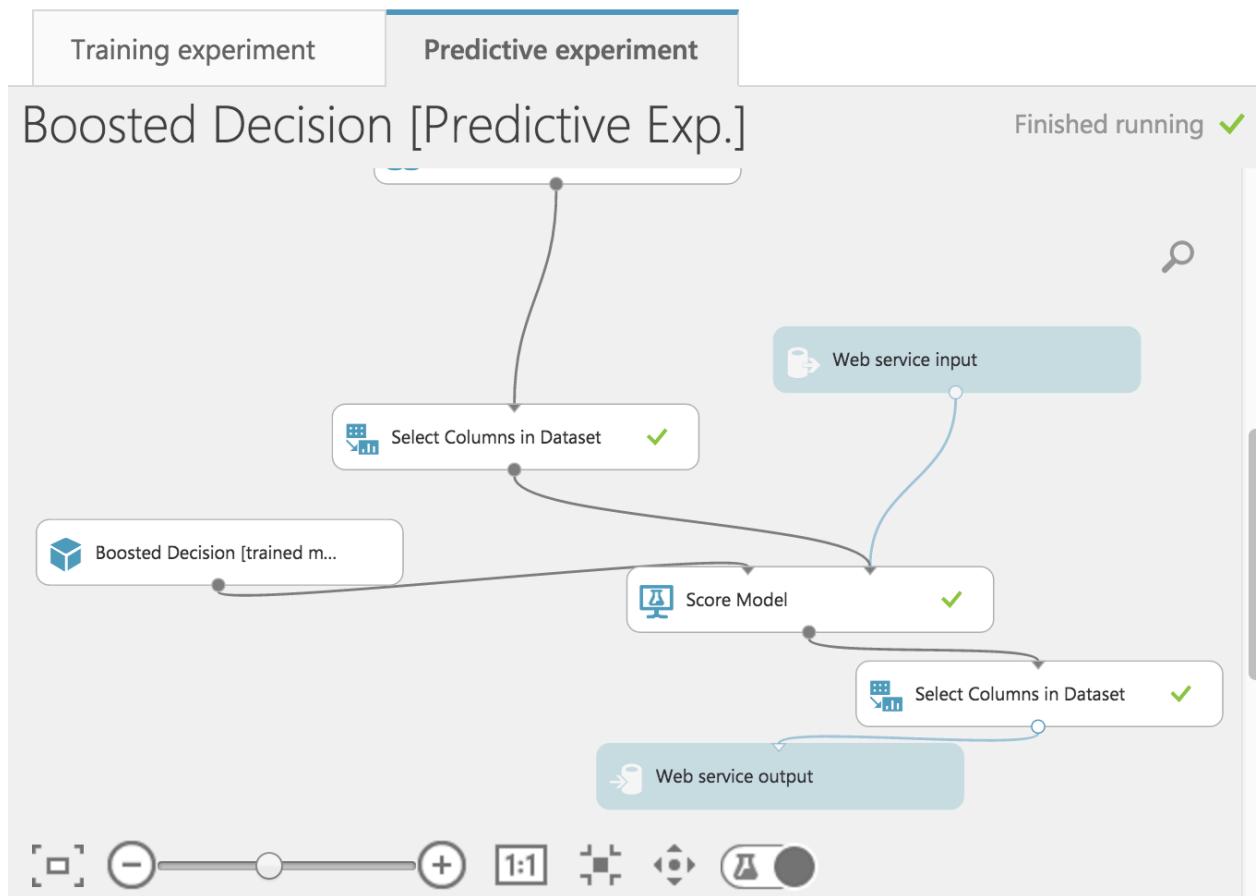
Neural Network Classification ➤ Evaluate Model ➤ Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
6688	7405	0.603	0.586	0.5	0.629
False Positive	True Negative	Recall	F1 Score		
4728	11709	0.475	0.524		
Positive Label	Negative Label				
YES	NO				

Boosted Decision Tree

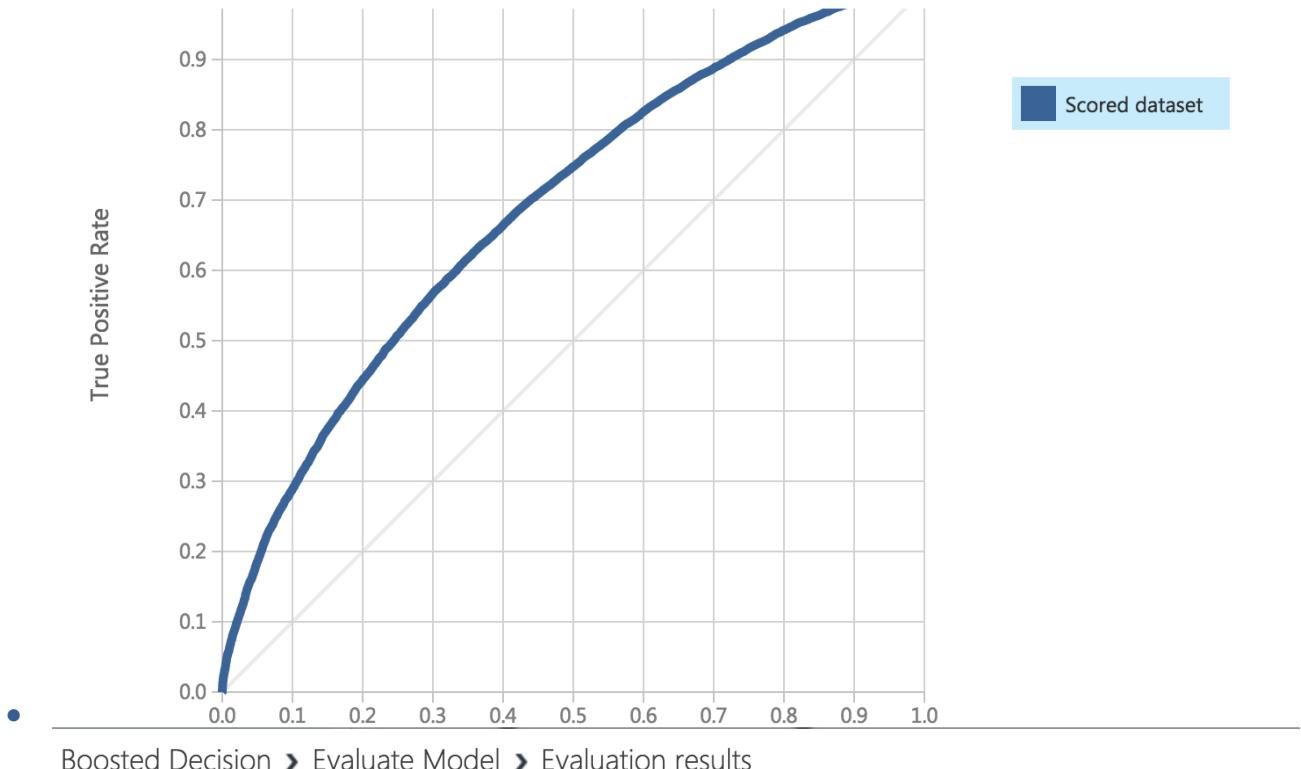


- **Model deployed on web:**



ROC Curve and Accuracy:

Boosted Decision ➤ Evaluate Model ➤ Evaluation results



Boosted Decision ➤ Evaluate Model ➤ Evaluation results

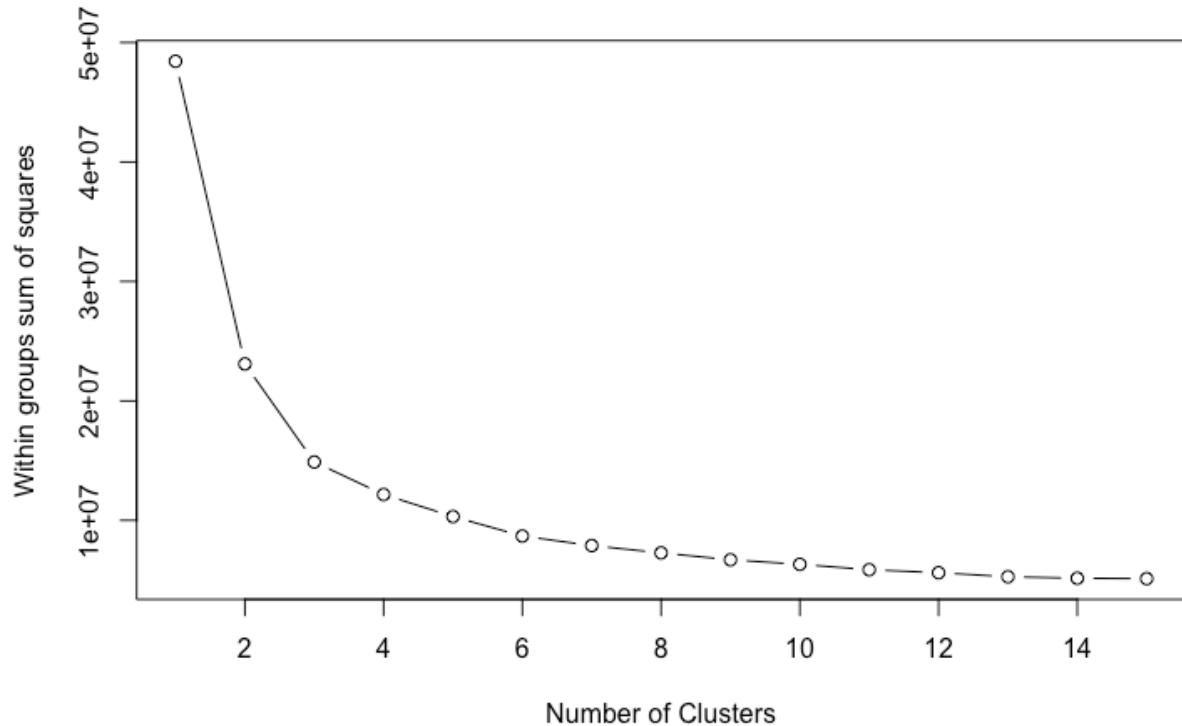
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
12473	11037	0.638	0.628	0.5	0.690
False Positive	True Negative	Recall	F1 Score		
7380	19993	0.531	0.575		
Positive Label	Negative Label				
YES	NO				

CLUSTERING

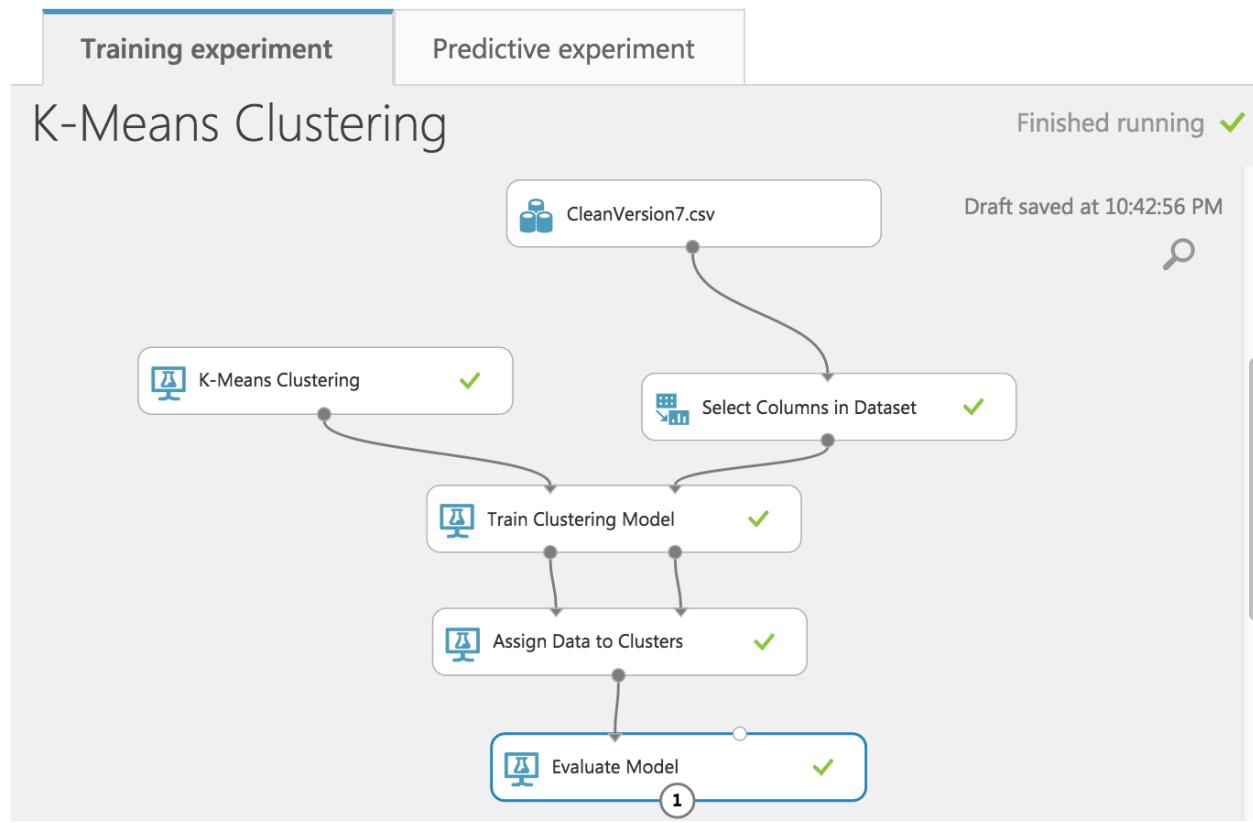
Deciding number of clusters based on elbow graph

K-means Code:

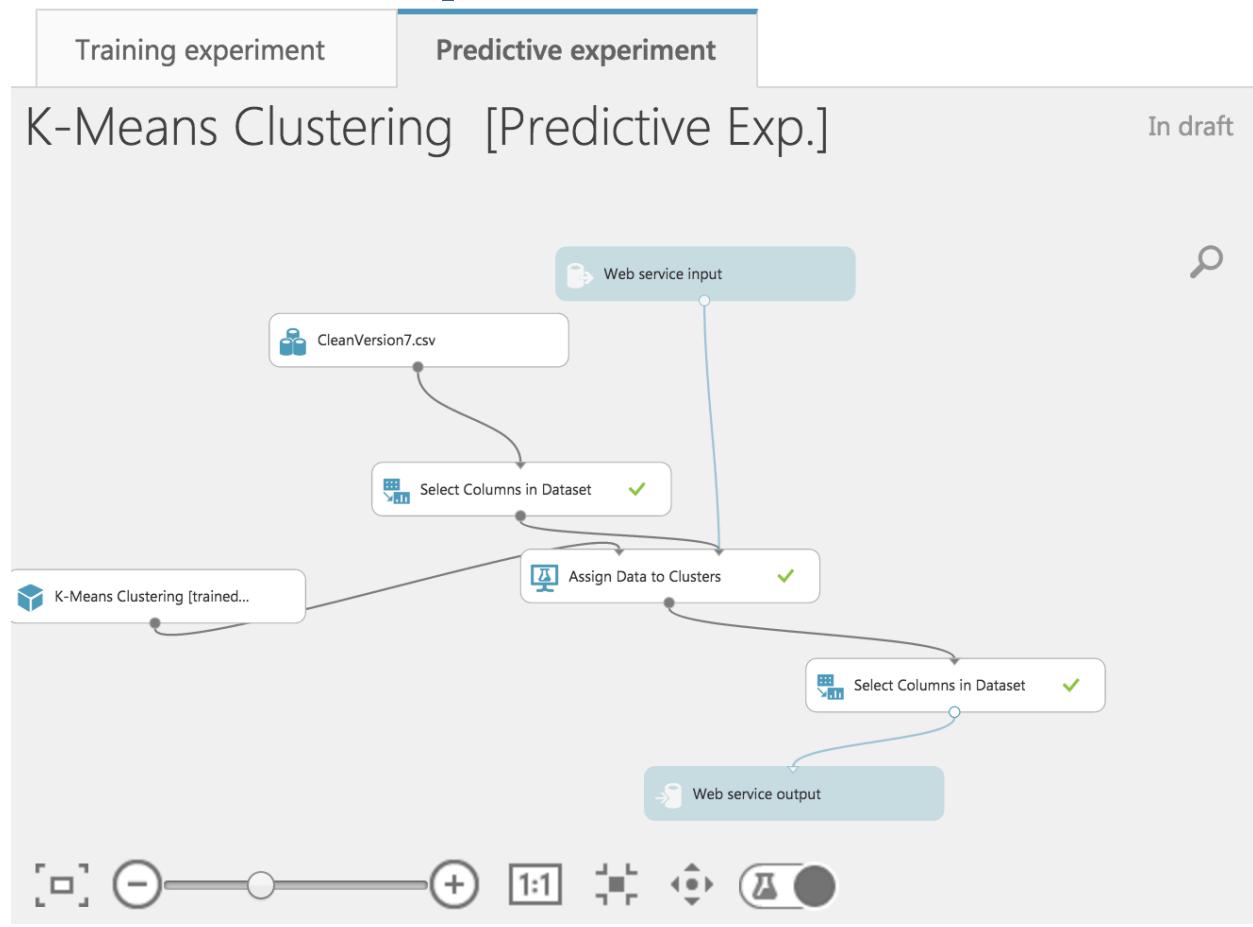
```
kmeansdata <- tempdiabdata[,c(10,13,14,15,16,17,18,22)]  
  
km.out <- kmeans(kmeansdata,5,nstart=10)  
kmeansdata$clustertagkmeans <- km.out$cluster  
km.out$size  
# plot(df2, col=(km.out$cluster), main="K-mean result with  
k=3")  
  
wss <- (nrow(kmeansdata)-1)*sum(apply(kmeansdata,2,var))  
for (i in 2:15) wss[i] <-  
sum(kmeans(kmeansdata,centers=i)$withinss)  
plot(1:15, wss, type="b", xlab="Number of Clusters",  
ylab="Within groups sum of squares")#Scatterplot
```



K Means:



Model deployed on web:



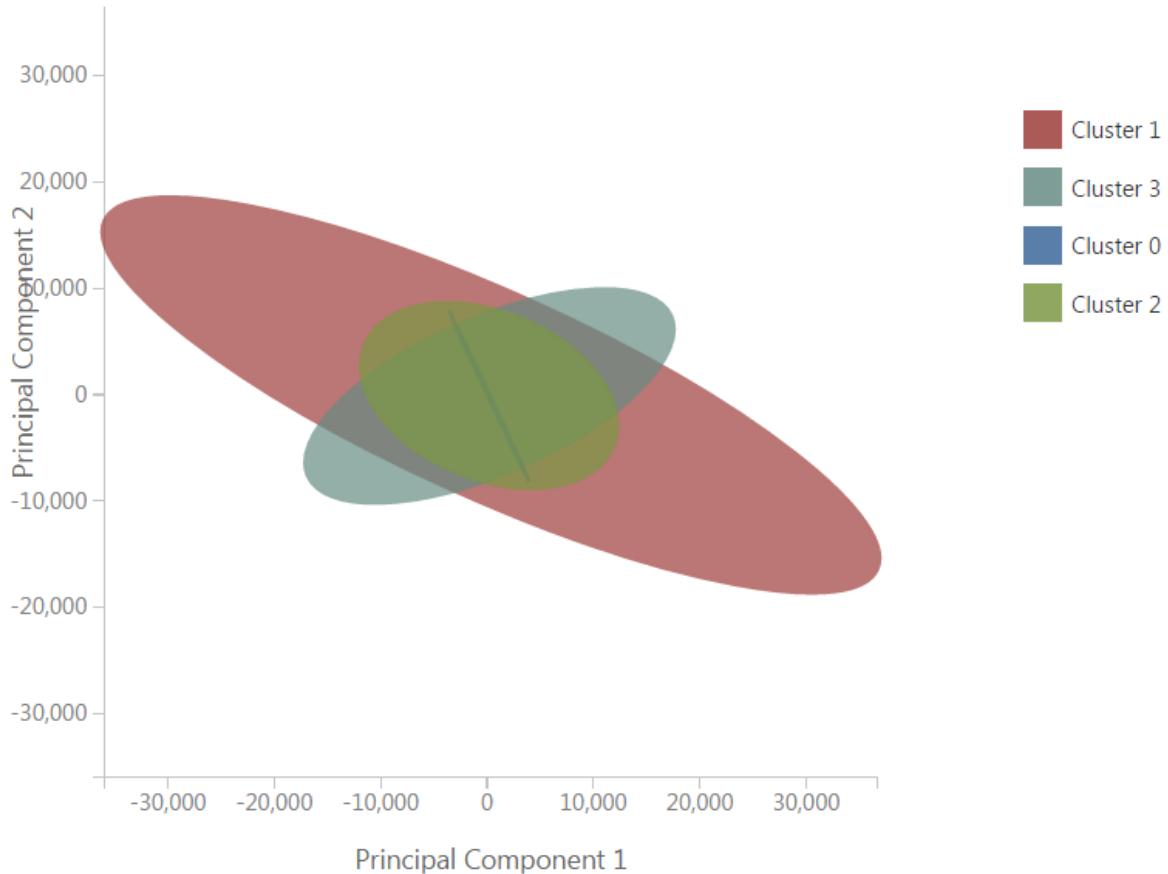
Evaluate Model:

K-Means Clustering ➤ Evaluate Model ➤ Evaluation results

rows	columns	Combined Evaluation	18.801793	101766	75.87267
5	5	9.535167	26.040193	16283	64.607828
		9.641087	21.860685	17408	63.401558
		11.922712	15.728204	34066	54.247286
		8.916847	16.849157	34009	75.87267
		8.881709			

- **Cluster Plot:**

K-Means Clustering ➤ Assign to Clusters ➤ Results dataset



Architecture:

- We have used Spring MVC architecture.
- Model is our Azure model.
- Controller is Spring Controller which will access data from model using API URL and API key of Azure model.
- View is our HTML form, which requests various parameter values from user.
- Using controllers, we have pulled the response from model which in our case is Azure model using Request-response URL and primary key generated by Azure. Controller passes this data on view(HTML form) so that user can access the results based on technique they choose.

API Information in Azure:

Once all models are deployed as a service, retrieve the api url and key from the consume web service page.

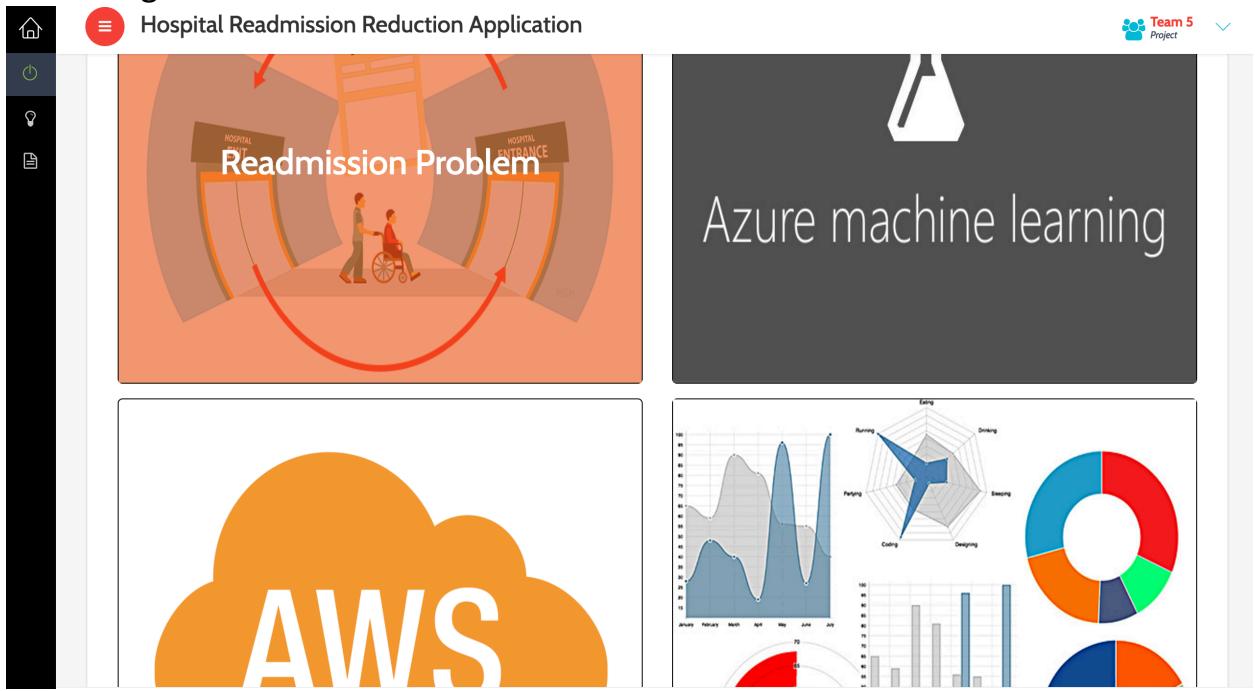
Basic consumption info

Want to see how to consume this information? [Check out this easy tutorial.](#)

Primary Key	njNoWTN5tDsV70Wjs1k5fmR5fJmLxgzHjek7nx1OOEdzdiyvguzFPP4HZG49j9tboDKTQwf1O+n5sKqTlrH/ZA==	
Secondary Key	6JQUyB3WzFslkqU+AJdOphN+nEjArkepcYBAwMqgsqRIWAeKIPKFe5aQk238M1thZGC30RaQ3C/rHiw1TIA==	
Request-Response	https://ussouthcentral.services.azureml.net/workspaces/8efb18bffe094a0990ccd0fd4251d124/services/26cb9f487c2346c48d9a7b218acc5b0/execute?api-version=2.0&format=swagger	
Batch Requests	https://ussouthcentral.services.azureml.net/workspaces/8efb18bffe094a0990ccd0fd4251d124/services/26cb9f487c2346c48d9a7b218acc5b0/jobs?api-version=2.0	

Web Interface:

As seen below our home screen has three selection options for machine learning techniques to choose namely Regression, Classification and Clustering



Input Form:

- Classification Real time Input

The screenshot shows a web-based application titled "Hospital Readmission Reduction Application". On the left, there is a vertical sidebar with icons for Home, Power, Location, and File. The main content area has a title "Enter Input for Classification". Below the title are five input fields with the following values:

- Race: Caucasian
- Admission Type: Emergency
- Discharge Disposition: Discharged to home
- Admission Source: Physician Referral
- Diagnosis 1: 648

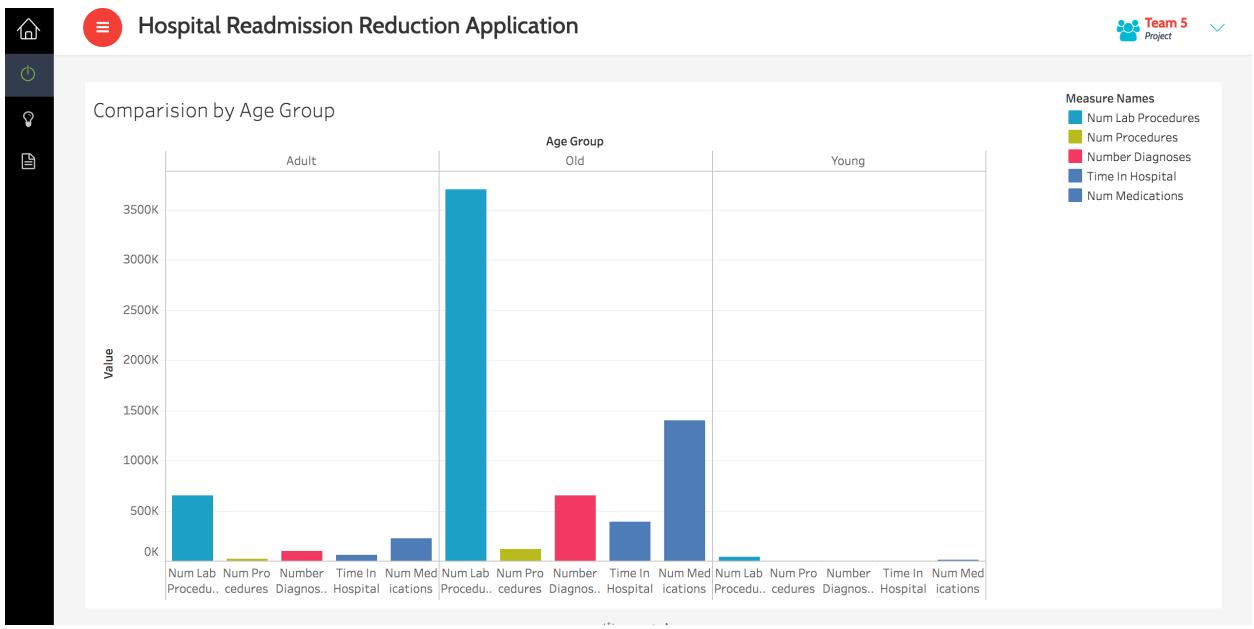
In the top right corner, there is a "Team 5 Project" logo.

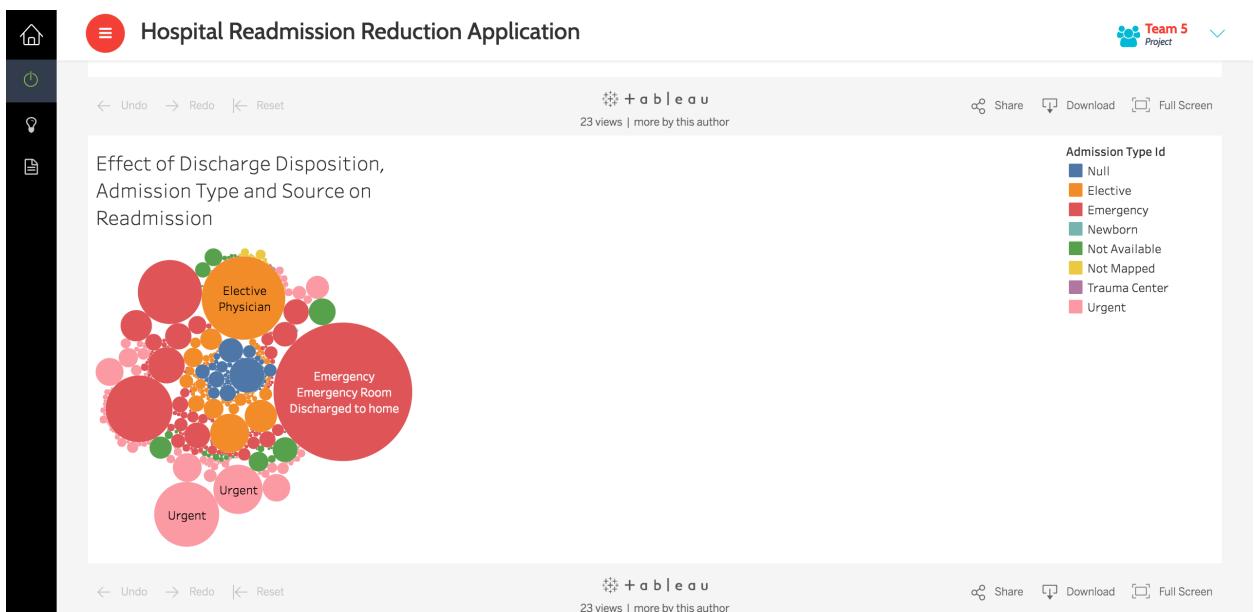
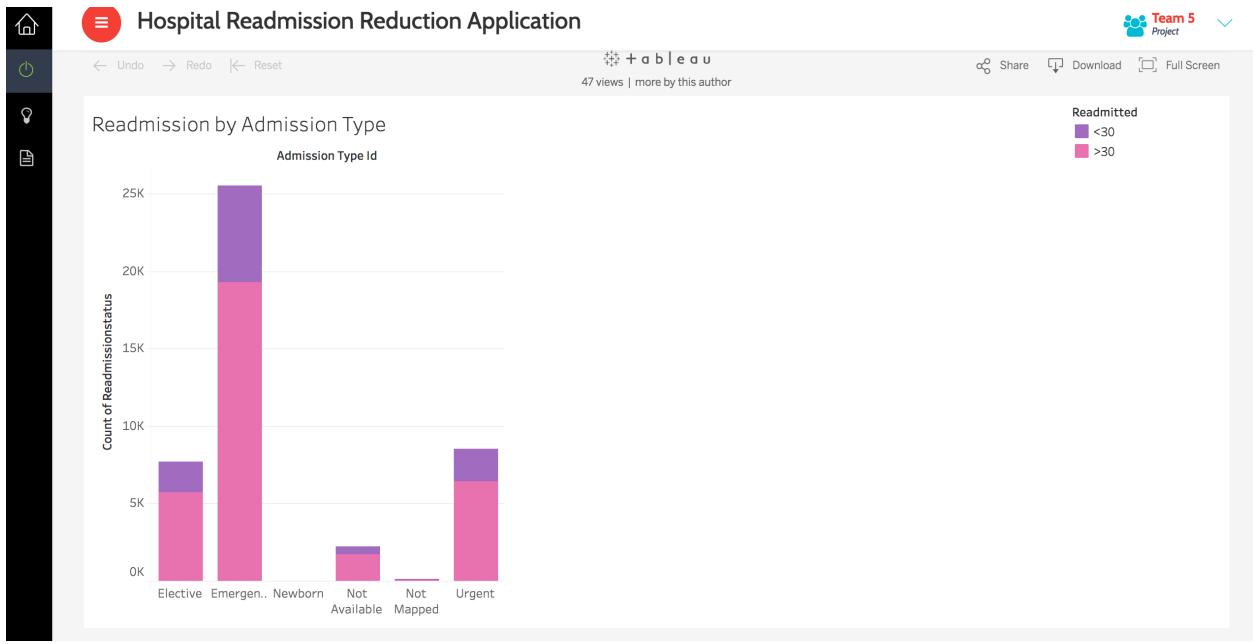
Input Form:

- Classification Batch Input

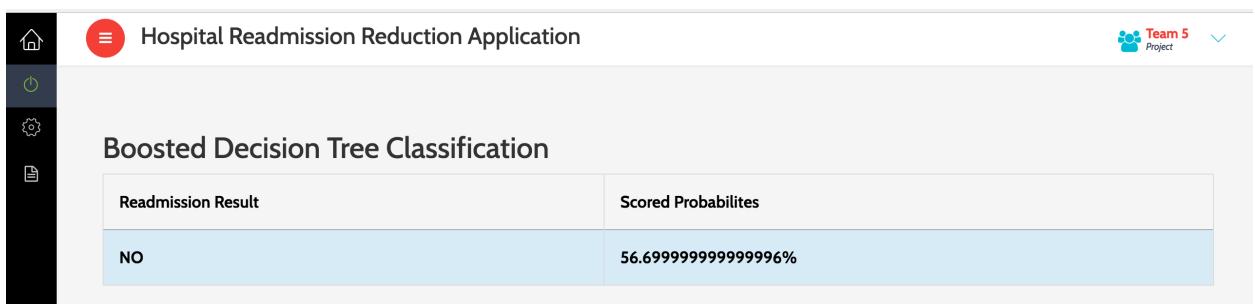
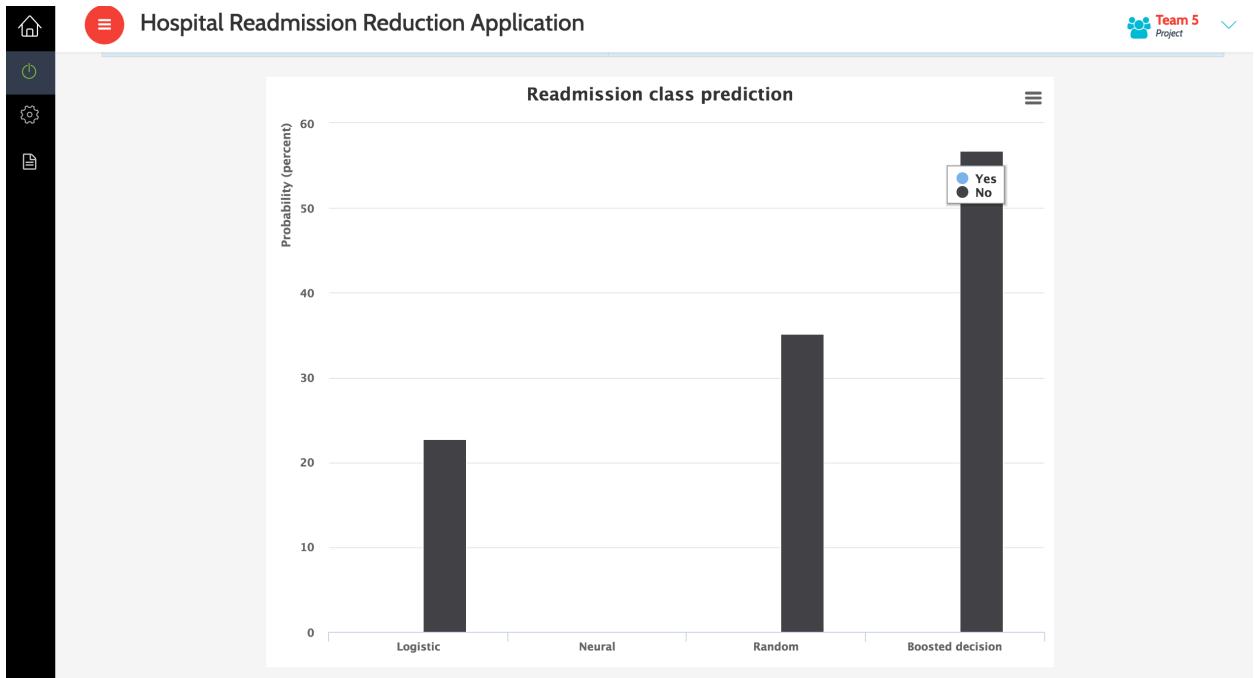
The screenshot shows a web application interface titled "Hospital Readmission Reduction Application". On the left is a vertical sidebar with icons for Home, Power, Settings, and a Document. The main content area has a title "Enter Input File for Batch Prediction". It contains two input fields: "Name:" with a placeholder "Enter file name" and "File:" with a placeholder "Choose File No file chosen". A blue "Submit" button is at the bottom.

Visualizations:





- **Classification Output:**



- We have four modules on our Website.
 1. Overview
 2. Azure ML
 3. Amazon Web service
 4. Visualization

1. Overview.

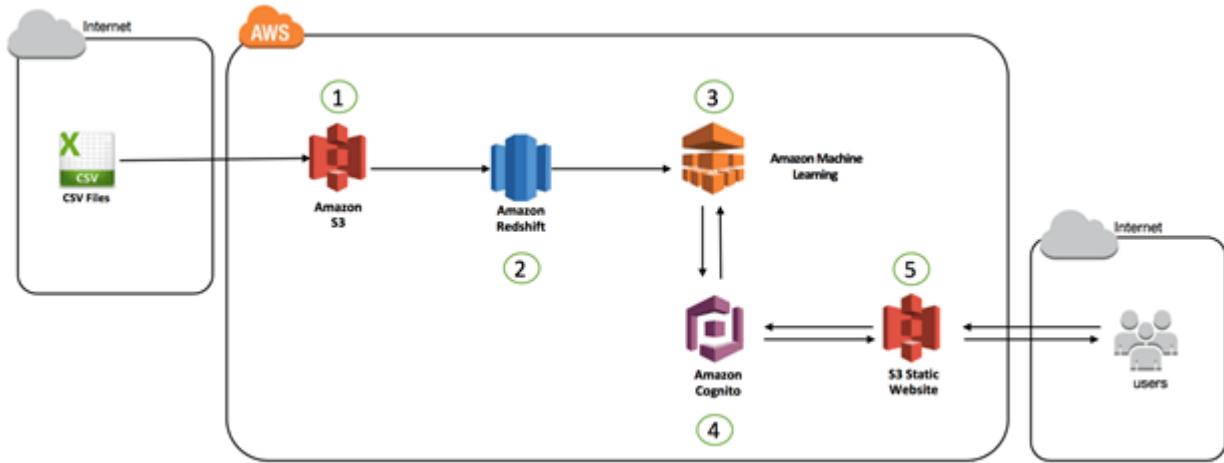
This module shows the overview problem of applying penalty which on most of hospitals due to increased number of readmissions.

2. Azure ML

- We have created a web site which accepts the parameters for classification of readmission status that consumes REST API service to call the web services deployed on Azure machine learning studio.
- The web site consists of an html form that takes user input for the features selected to run the algorithms in azure.
- We get back output value of readmission of 4 algorithms with their probabilities.
- Following are 4 algorithms used for classification
 1. Logistic Regression
 2. Random Forest
 3. Neural Network
 4. Boosted Decision Tree
- These features are passed through the REST API in json format.
- We have used java in eclipse to create this web application.
- Once the call to azure machine learning studio is made and the models are run as required, the result of the scored label is sent back to the application as a rest response which is deserialized.
- Scored Label and Scored Probability is displayed for classification .
- Scored Label is displayed for prediction.

- **K-means** algorithm is used for clustering.
- Assignment to which cluster means to which doctor that patient should report that has high probability of readmission.

3. Amazon web service



1. The first step is to get the data into Amazon S3, the object storage service from AWS.
2. Amazon Redshift acts as the database for the huge amounts of structured clinical data. The data is loaded into Amazon Redshift tables and is massaged to make it more meaningful as a data source for an ML model.
3. A binary classification ML model is created using Amazon ML, with Amazon Redshift as the data source. A real-time endpoint is also created to allow real-time querying for the ML model.
4. Amazon Cognito is used for secure federated access to the Amazon ML real-time endpoint.
5. A static web site is created on S3. This website hosts the end user facing application using which one can query the Amazon ML endpoint in real time.

4. Visualization

- We have deployed the tableau sheets on tableau server.
- Using javascript code we have integrated sheets in our website.

Chart -1 (Comparison by Age Group) :

- The factors such as number of lab procedures, number of procedures, number of diagnoses, time in hospital and number of medications when analyzed with respect to the age group, suggests that patients under the Age Group - Old (51+) are more likely to have these as a major factor or are more vulnerable to readmission as compared to the Young and Adult age Group

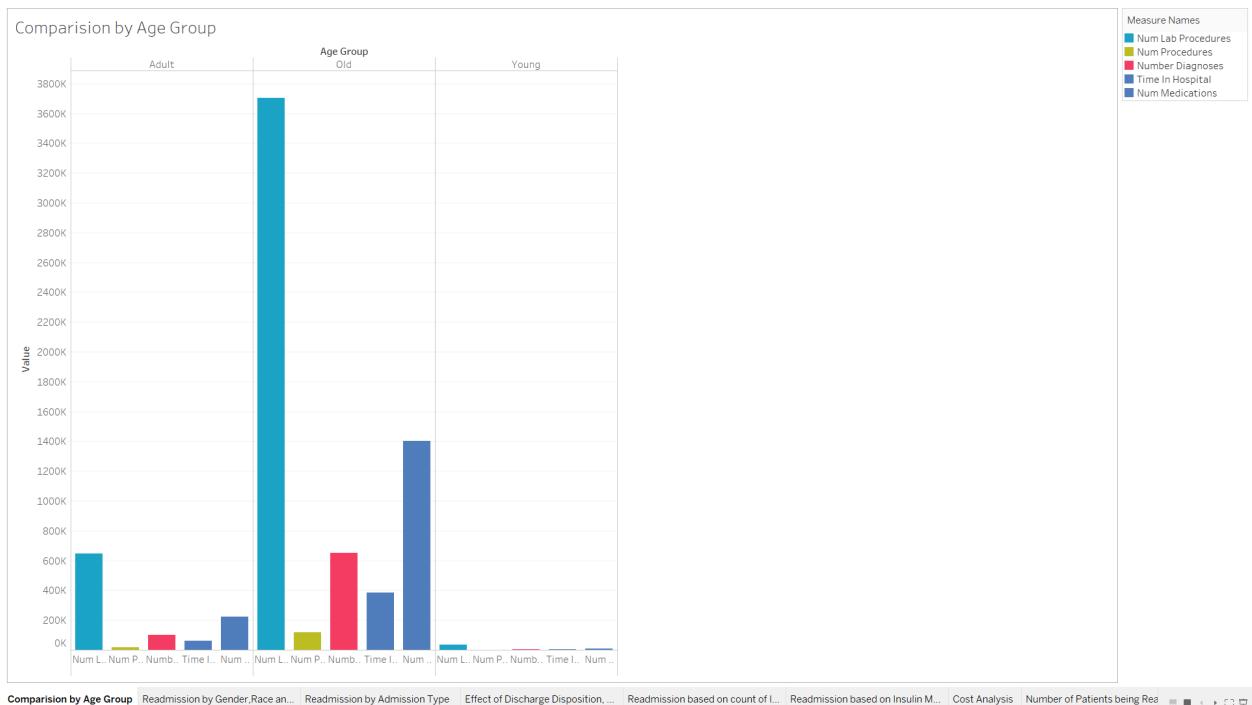


Chart -2 (Readmission by Gender, Race and Age Group):



From the above chart it is clear that Old Female patients who are Caucasian have the highest count of readmissions as compared to others. Old Male belonging to Caucasian race are the second most likely set of patients to get readmitted.

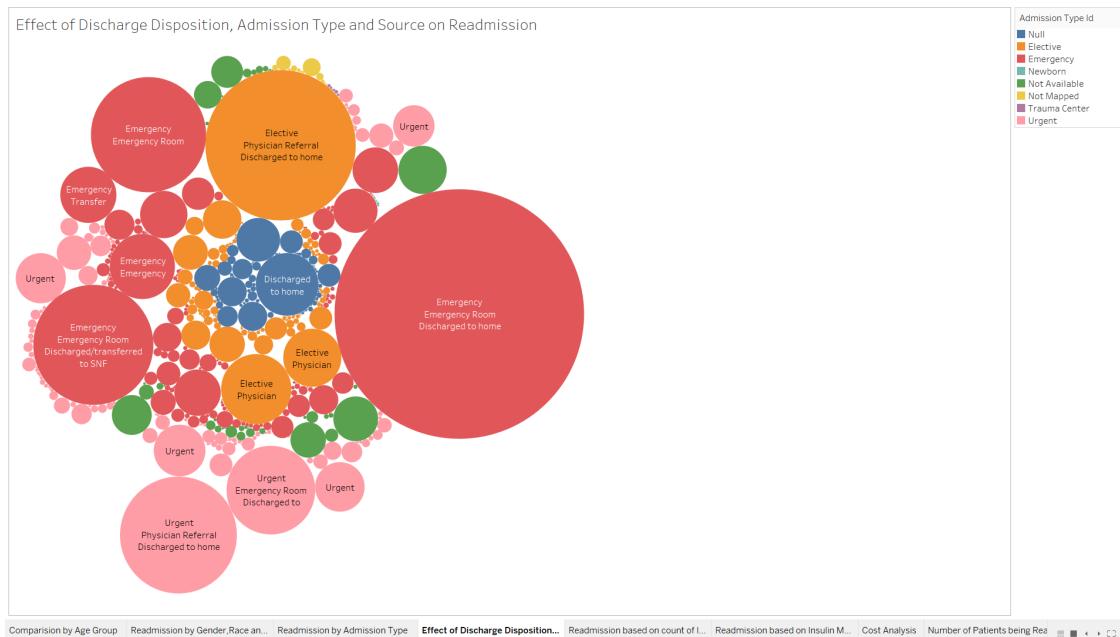
Thus, measures should be taken by the hospitals to facilitate improved treatment to reduce the chances of their readmission.

Chart – 3 (Readmission by Admission Type):



From the above chart it is clear that patients admitted through emergency are more probable to get readmitted in the near future. Emergency admission count up to 25% of the patient admissions. Thus, extra care should be taken for such patients to avoid readmissions.

Chart – 4 (Effect of Discharge Disposition, Admission Type and Source on Readmission) :

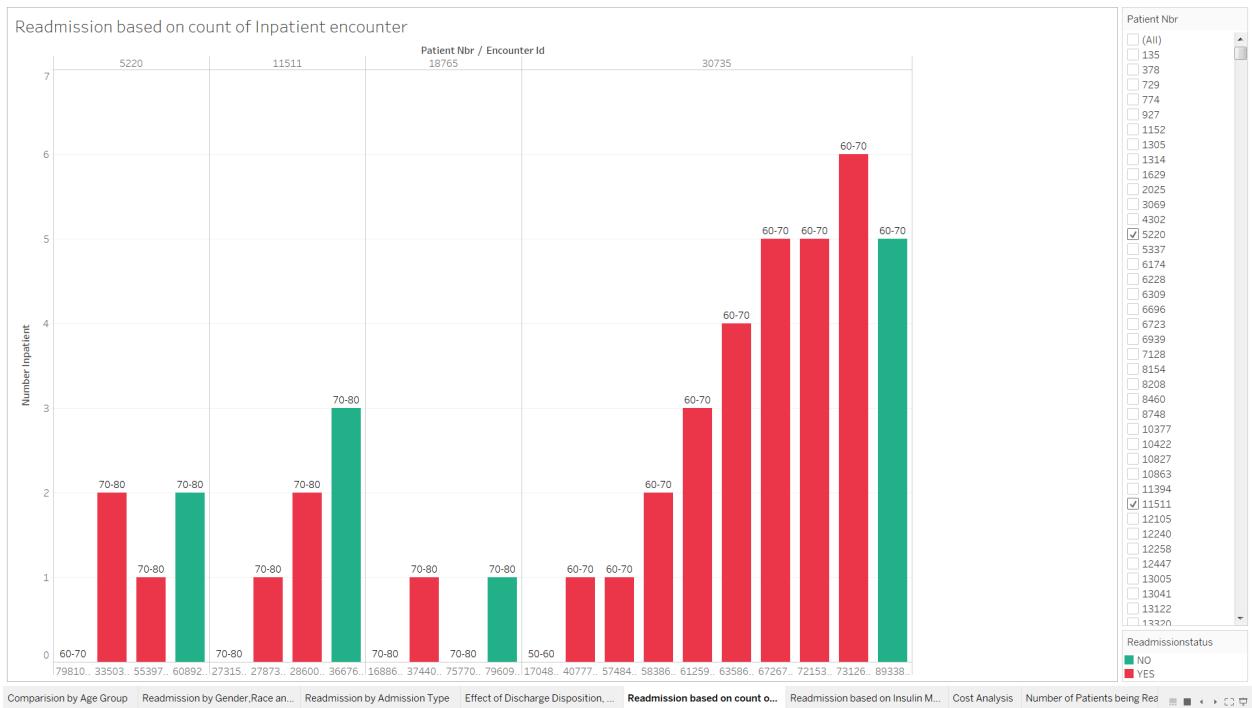


Discharge disposition, admission type, and admission source were identified as strong predictors of readmission. These factors are most important for identifying high-risk patients.

A patient brought in as an emergency and who are discharged to home are more likely to be readmitted in the near future.

Thus, emergency admissions should be closely monitored before being discharged to home.

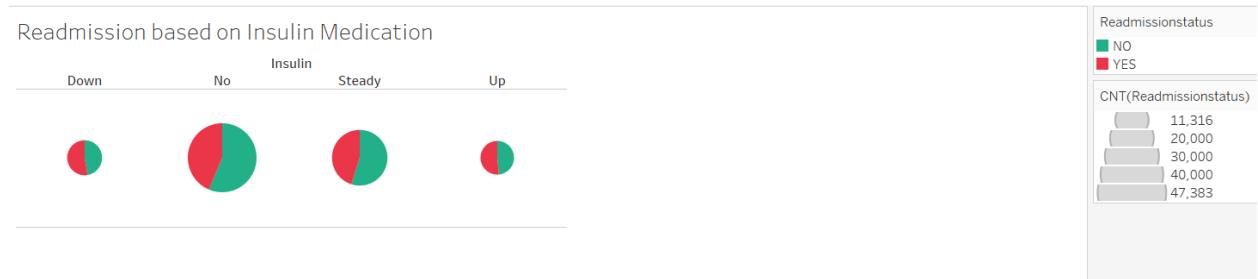
Chart – 5 (Readmission based on count of Inpatient encounter):



Number of inpatient visits is another important factor. From the above chart it can be depicted that these patients have been regularly admitted, it may be due to different types of diagnosis that have been performed on them over their visits as inpatients or may be because of inadequate care provided on their initial visits.

These insights can help healthcare providers to improve inpatient diabetic care.

Chart – 6 (Readmission based on Insulin Medication):



Insulin is always necessary for diabetes patients because the body has no internal source of insulin, particularly those who have difficulty controlling their diabetes with oral medications.

From the above chart it can be visualized that readmission rate is more when the dosage for insulin have been increased or decreased for patients. On the other hand the readmission are less when the dosage has been steady. Thus, proper care must be taken while prescribing change in dosage considering various other factors mentioned above.

Chart -7 (Number of Patients being Readmitted):



From the above chart we can conclude that 47% patients have been readmitted, out of which around 11% have been readmitted within 30 days of discharge.

Improvements in care at the time of patient discharge can reduce readmission rates.

Hospitals, in collaboration with their medical communities, can take a number of actions to reduce readmissions: ensure patients are clinically ready at discharge; reconcile medications; improve communications among providers involved in transition of care; and educate patients about symptoms to monitor, whom to contact with questions, and where and when to seek follow-up care.

Chart – 8 (Cost Analysis):

The following chart gives a picture about how much cost is spent on treatment of the readmitted patients.

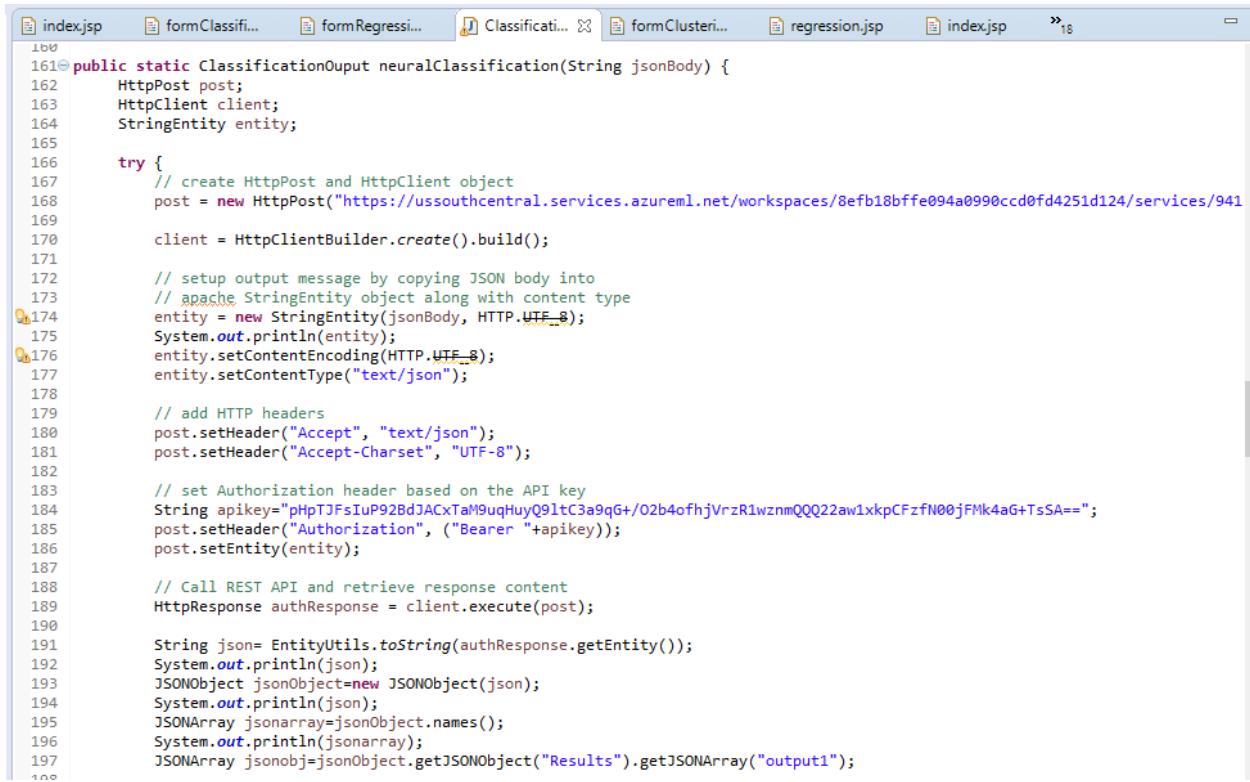
From the chart it is clear that \$458 million have been spent on the treatment for readmitted patients.

Saving such huge amount cost is essential for healthcare system.



Code Snippet to access Azure Web Service from Spring:

Using API URL and key for the each algorithm the following code requests azure api by providing values for selected features and expects the prediction output in the form of json.



The screenshot shows a Java code editor with multiple tabs at the top. The active tab contains Java code for making a REST API call to an Azure ML service. The code uses Apache HttpClient to send a POST request with JSON body and receive a JSON response. It includes imports for java.util, java.io, org.apache.http.client.methods, org.json, and org.json.JSONArray.

```

160 public static ClassificationOutput neuralClassification(String jsonBody) {
161     HttpPost post;
162     HttpClient client;
163     StringEntity entity;
164
165     try {
166         // create HttpPost and HttpClient object
167         post = new HttpPost("https://ussouthcentral.services.azureml.net/workspaces/8efb18bffe094a0990cc0fd4251d124/services/941
168         client = HttpClientBuilder.create().build();
169
170         // setup output message by copying JSON body into
171         // apache StringEntity object along with content type
172         entity = new StringEntity(jsonBody, HTTP.UTF_8);
173         System.out.println(entity);
174         entity.setContentType(HTTP.UTF_8);
175         entity.setContentType("text/json");
176
177         // add HTTP headers
178         post.setHeader("Accept", "text/json");
179         post.setHeader("Accept-Charset", "UTF-8");
180
181         // set Authorization header based on the API key
182         String apikey="phpTJFsIuP92BdJAxCtM9uqHuyQ9ltC3a9qG+/02b4ofhjVrzR1wznmQQQ22aw1xkpCFzfN00jFMk4aG+TsSA==";
183         post.setHeader("Authorization", ("Bearer "+apikey));
184         post.setEntity(entity);
185
186         // Call REST API and retrieve response content
187         HttpResponse authResponse = client.execute(post);
188
189         String json= EntityUtils.toString(authResponse.getEntity());
190         System.out.println(json);
191         JSONObject jsonObject=new JSONObject(json);
192         System.out.println(jsonObject);
193         JSONArray jsonArray=jsonObject.names();
194         System.out.println(jsonArray);
195         JSONArray jsonobj=jsonObject.getJSONObject("Results").getJSONArray("output1");
196
197     } catch (Exception e) {
198         e.printStackTrace();
199     }
200 }

```

Request/Response to API:

The following image shows the JSON request and response sent to the api. We used Postman to test each web service before integrating with the web service.

The screenshot shows the Postman application interface with two API requests and their corresponding responses.

Request 1:

- Method: POST
- URL: <https://ussouthcentral.services.azureml.net/workspaces/daea485135a432781ecc9e9e938a2c/services/0>
- Headers (2):
 - Content-Type: application/json
 - Authorization: Bearer VoMz70f/u+3Om23A12iMH39Tnlamoo8Hji/
- Body (raw JSON):


```

1 {
2   "Inputs": {
3     "input1": [
4       {
5         "Date": "2013-01-02T00:00:00",
6         "Hour": "1",
7         "Month": "12",
8         "Consumption_per_squaremeter": "0.002073",
9         "TemperatureF": "15.6",
10        "Dew_PointF": "32.9",
11        "Sea_Level_PressureIn": "29.53",
12        "Wind_SpeedMPH": "9.8",
13        "WindDirDegrees": "235",
14      }
15    ],
16  },
17  "GlobalParameters": {}
18 }
19
20
21 
```

- **Conclusion**

Based on the analysis and modelling, we can infer that lot of cost is penalized on hospitals due to increased readmissions. Approximately 47,000 patients were readmitted based on this dataset. Out of which 11,000 patients were readmitted within 30 days. So total penalty based on factor of 3%(based on current year) on total readmission cost which is approximately 458 million dollars, is 13749450 dollars. Hence, we can infer almost 1 million dollars is penalty in 1 year, which can

be saved and patients will get better care, so that they don't get readmitted again.

- References:

- <https://www.hindawi.com/journals/bmri/2014/781670/>
- <https://data.medicare.gov/Hospital-Compare/Hospital-Readmissions-Reduction-Program/9n3s-kdb3>
- <https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008#>