Predicting Inpatient Hospital Readmissions Within 30 Days

Zhanna Zhanabekova
June 14, 2018
New York City, NY

Objective

- ... using 1999-2008 data on inpatient hospitalizations of patients with diabetes

- Predict what patients will be re-admitted to the hospital within 30 days of the current discharge

- Methods used: Decision Tree algorithm

Data

- Diabetic inpatient encounters
  - 1999-2008
  - 130 U.S. hospitals and integrated delivery networks
  - About 50 variables:
    - Patient number, race, gender, age category
    - Admission and discharge categories
    - Time in hospital
    - Medical specialty of admitting physician
    - Diagnoses
    - Number of lab test performed, number of medications administered, HbA1c test results,
    - Number of outpatient, inpatient, and emergency visits in the year before the hospitalization

Target Variable

- "Readmitted": Days to inpatient readmission
  - "<30" if the patient was readmitted in less than 30 days
  - ">30" if the patient was readmitted in more than 30 days
    - 30 days?
  - "No" if no record of readmission

- Target
  - = 1 if Readmitted = "<30"
  - = 0 Otherwise

Main Result (so far)

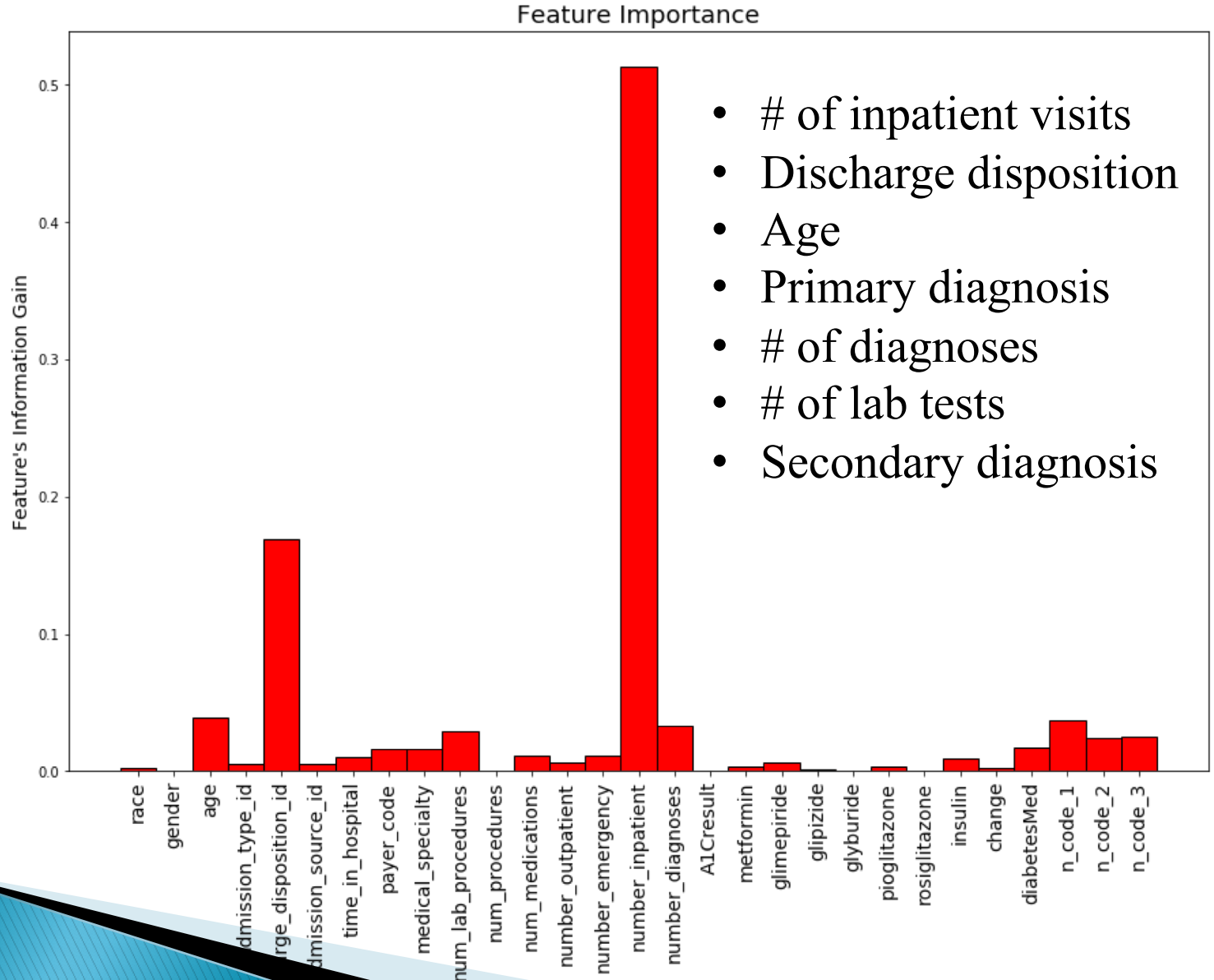
- Regularized decision tree trained on upsampled training data
  - Upsample the minority (positive) class to the size of the majority class
  - Choose hyperparameters – tree depth and number of samples in a leaf node – by using a precision-recall metric (average precision score, similar to ROC AUC)

Main Result (so far)

- **Sensitivity (TPR): 0.610**
  - FNR: **0.390**
- Specificity (TNR): 0.669
  - FPR: **0.331**

- Precision: 0.204
- F1 score: 0.306
- ROC AUC: 0.677
- Average precision score: 0.256

Model #5 (Regularized PR AUC, Up)



- # of inpatient visits
- Discharge disposition
- Age
- Primary diagnosis
- # of diagnoses
- # of lab tests
- Secondary diagnosis

Data Cleaning Steps

- Drop observations that resulted in death or transfer to hospice care
- Keep only 1 encounter per patient
- Drop categorical variables where one value dominates (>= 95% of cases)
- Reduce cardinality of diag_1, diag_2, diag_3
  (note: this introduced lots of missing values)
- Medication variables: recode as binary where 0=No and 1=Steady/Up/Down

What's In the Data So Far

- About 23% of all patients have more than one encounter (inpatient visits)
- About 13% of all patients were readmitted within 30 days
- Numeric variables – no missing values
- Categorical variables – keep missing values (treat as a separate category)

Data Limitations

- Diabetic encounters
  - May be missing observations on some diabetic patients who were hospitalized with non-diabetic diagnosis codes
    - e.g., a patient has only 1 encounter in our dataset but is coded as being readmitted within 30 days. This second encounter is missing
- Length of stay is between 1 and 14 days. May be excluding really sick patients
- No info on encounter date, hospital

Deduping the Dataset

- Sometimes a patient shows up multiple times
- Want to keep only 1 encounter per patient (want observations to be ~independent)
- Encounter date is not available so come up with an ad-hoc solution

- **Step 1.**
  - Keep all patients with only 1 encounter as is
  - For patients with only 1 readmission within 30 days, select that one encounter with target=1
  - Identify all patients with multiple encounters and multiple readmissions within 30 days, and keep only those observations with target=1
  - Identify all patients with more than 1 encounter but zero readmissions within 30 days
- **Step 2:**
  - Keep only those observations with the smallest number of missing values across all columns
- **Step 3:**
  - Sort data on patient_nbr and select the first observation for each patient

Modeling Steps (1)

- Encode categorical variables but no other pre-processing
- Create training (60%), validation (20%) and test (20%) sets

Modeling Steps (2)

- **Baseline DT: Unregularized, original** train data
  - Problems: overfit + unbalanced target variable
- **Tree #2: Unregularized, downsampled** train data
  - Problems: overfit + loss of information
- **Tree #3: Regularized (ROC AUC), downsampled** train data
- **Tree #4: Regularized (average precision score), downsampled**
  - Problems: lower sensitivity
- **<u>Tree #5: Regularized (average precision score), upsampled</u>**
  - Highest sensitivity (0.61) but still room for improvement

Summary of Results

Metric	#1	#2	#3	#4	#5
	Unreg Raw	*Unreg Down*	*Reg Down*	*Reg Down pr*	*Reg Up pr*
Accur Train	1.000	1.000	0.662	0.660	0.659
Accur Valid	0.791	0.563	0.678	0.682	0.662
TP	**380**	**1,004**	**982**	**985**	**1,043**
FN	1,330	706	728	725	667
TN	10,697	6,876	8,508	8,556	8,221
FP	1,591	5,412	3,780	3,732	4,067
Sensitivity	**0.222**	**0.587**	**0.574**	**0.576**	**0.610**
FNR	0.778	0.413	0.426	0.424	0.390
Specificity	0.871	0.560	0.692	0.696	0.669
FPR	**0.129**	**0.440**	**0.308**	**0.304**	**0.331**
Precision	0.193	0.156	0.206	0.209	0.204
F1 Score	0.206	0.247	0.303	0.307	0.306
ROC AUC	0.546	0.573	0.685	0.684	0.677
PR AUC	0.138	0.142	0.242	0.252	0.256

Model Evaluation

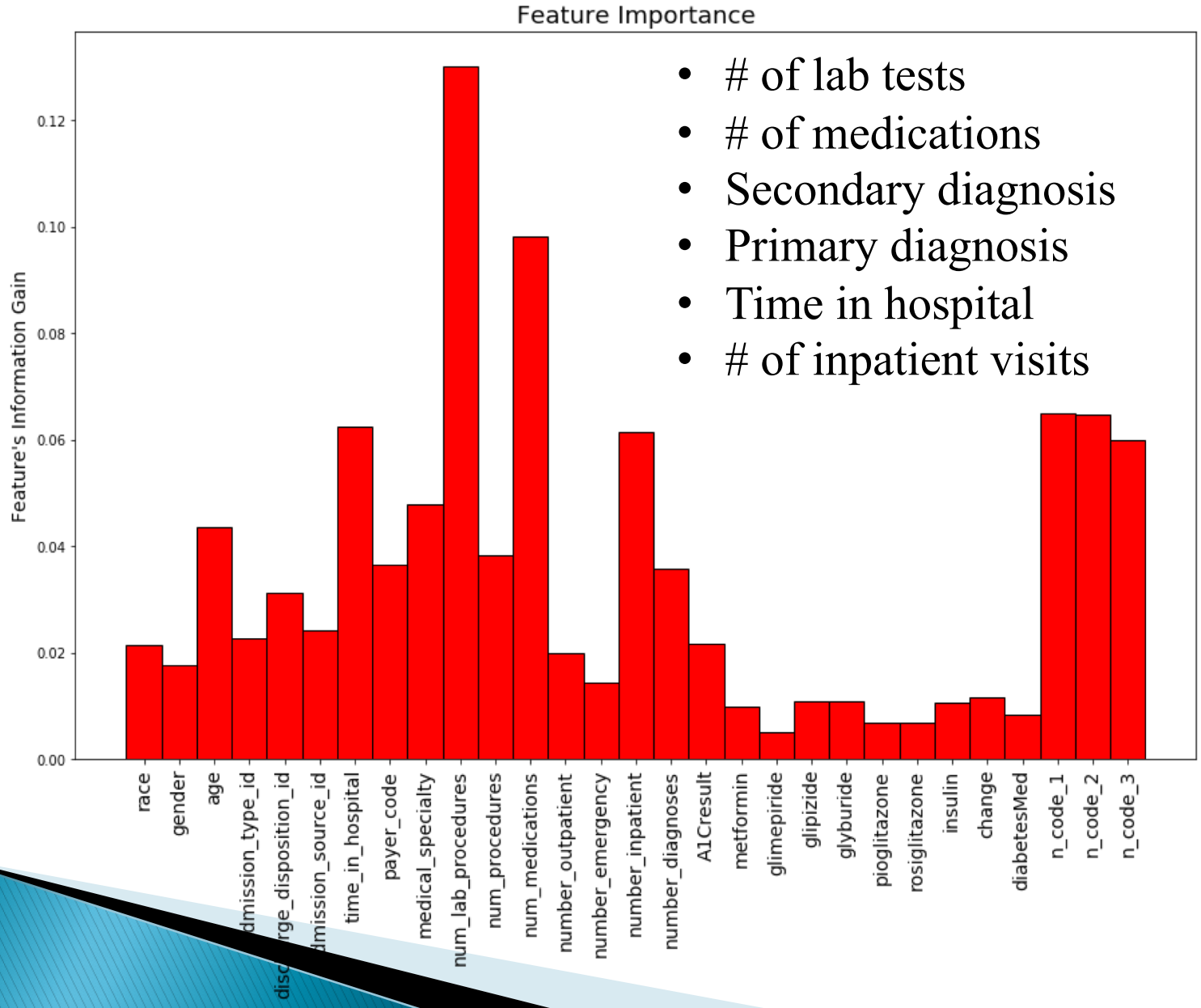
- Confusion matrix
- Sensitivity or True Positive Rate
- Specificity or True Negative Rate
- Precision
- F1, ROC AUC, PR AUC

- Readmissions (target=1) are very costly

- Want to catch as many high-risk patients as possible (max true positives), and want to minimize the false negative rate (don't want to miss any positive cases)

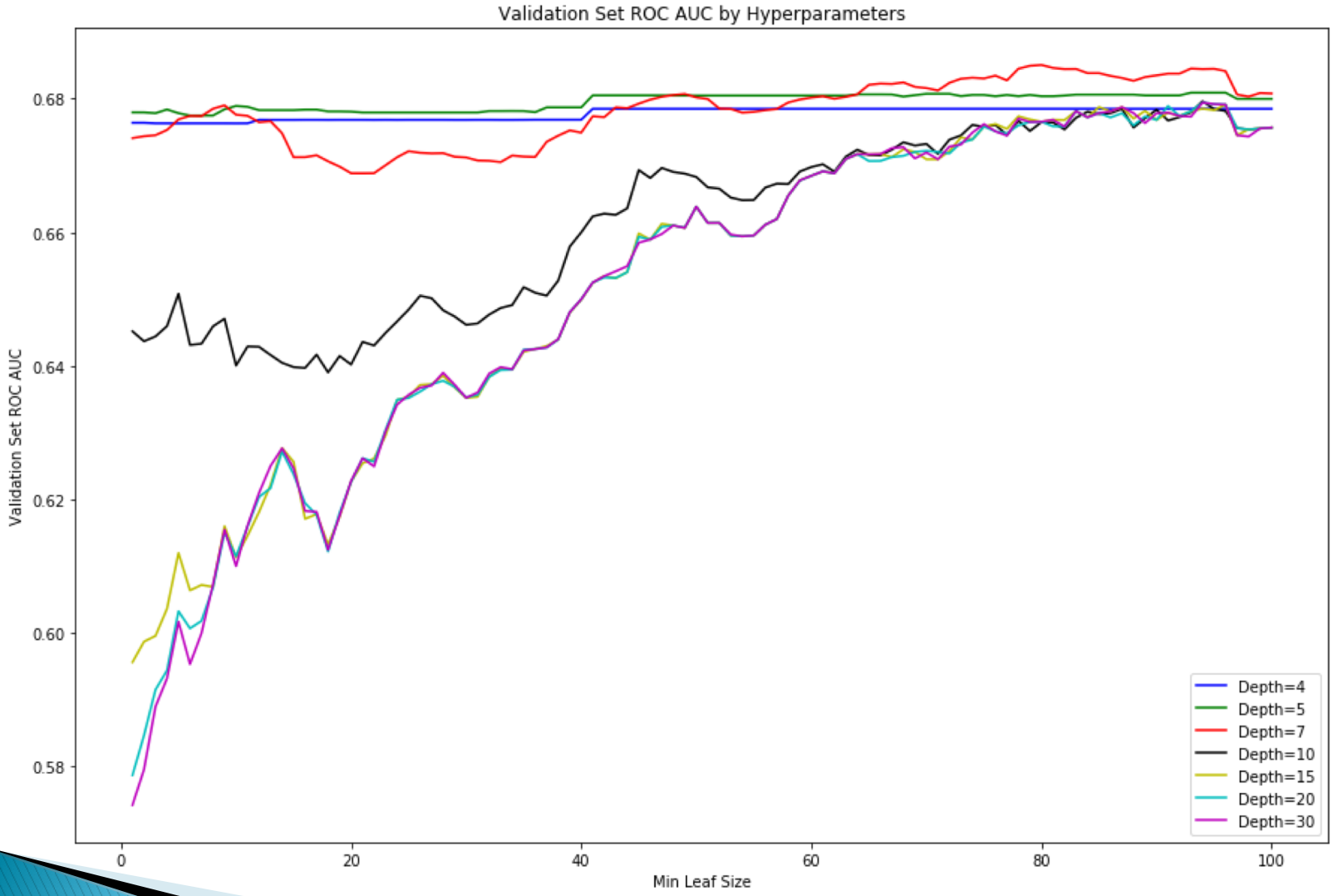
- So choose a model that gives the highest sensitivity

Baseline Model

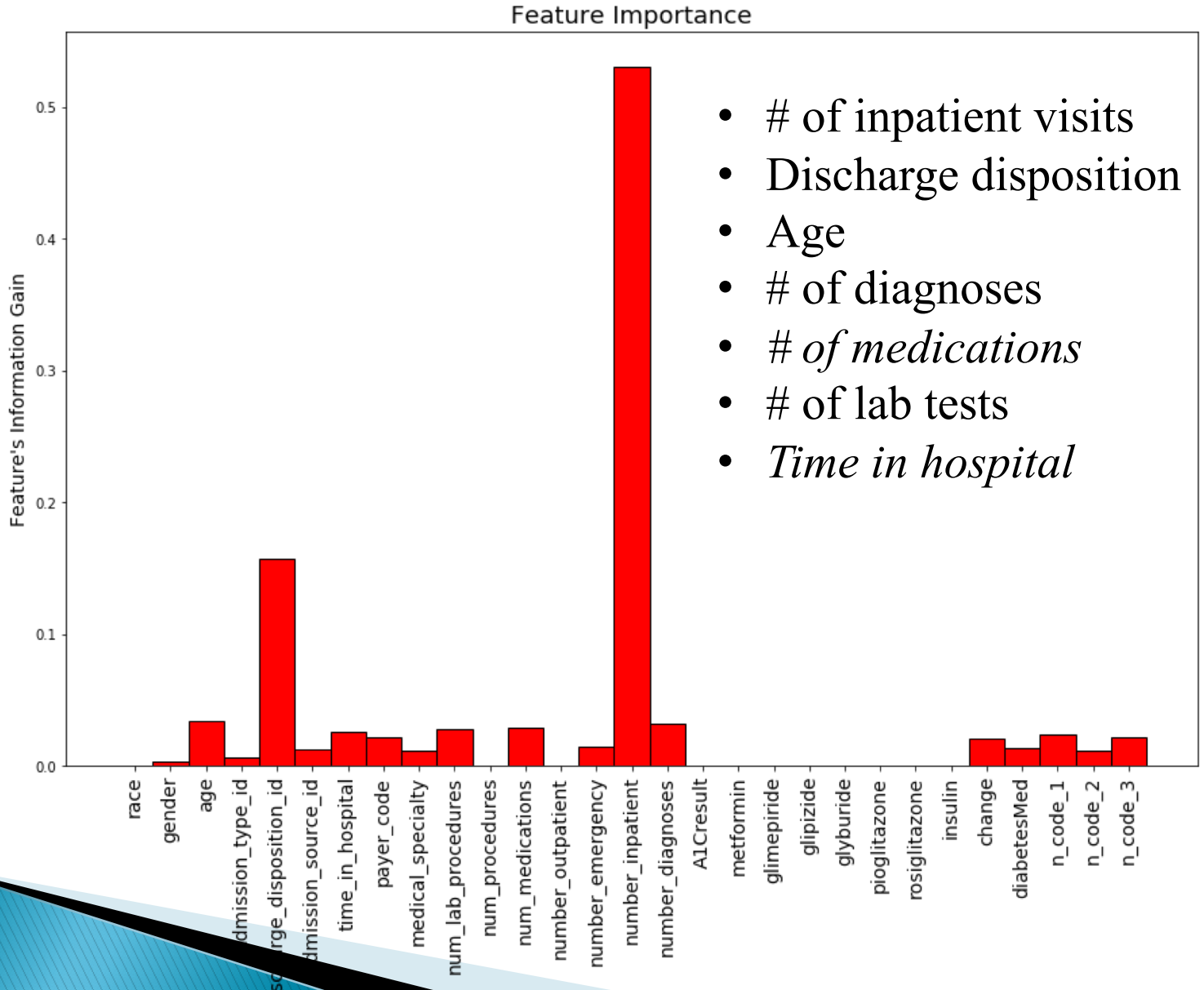


- # of lab tests
- # of medications
- Secondary diagnosis
- Primary diagnosis
- Time in hospital
- # of inpatient visits

Model #3 (Regularized ROC AUC, Down)

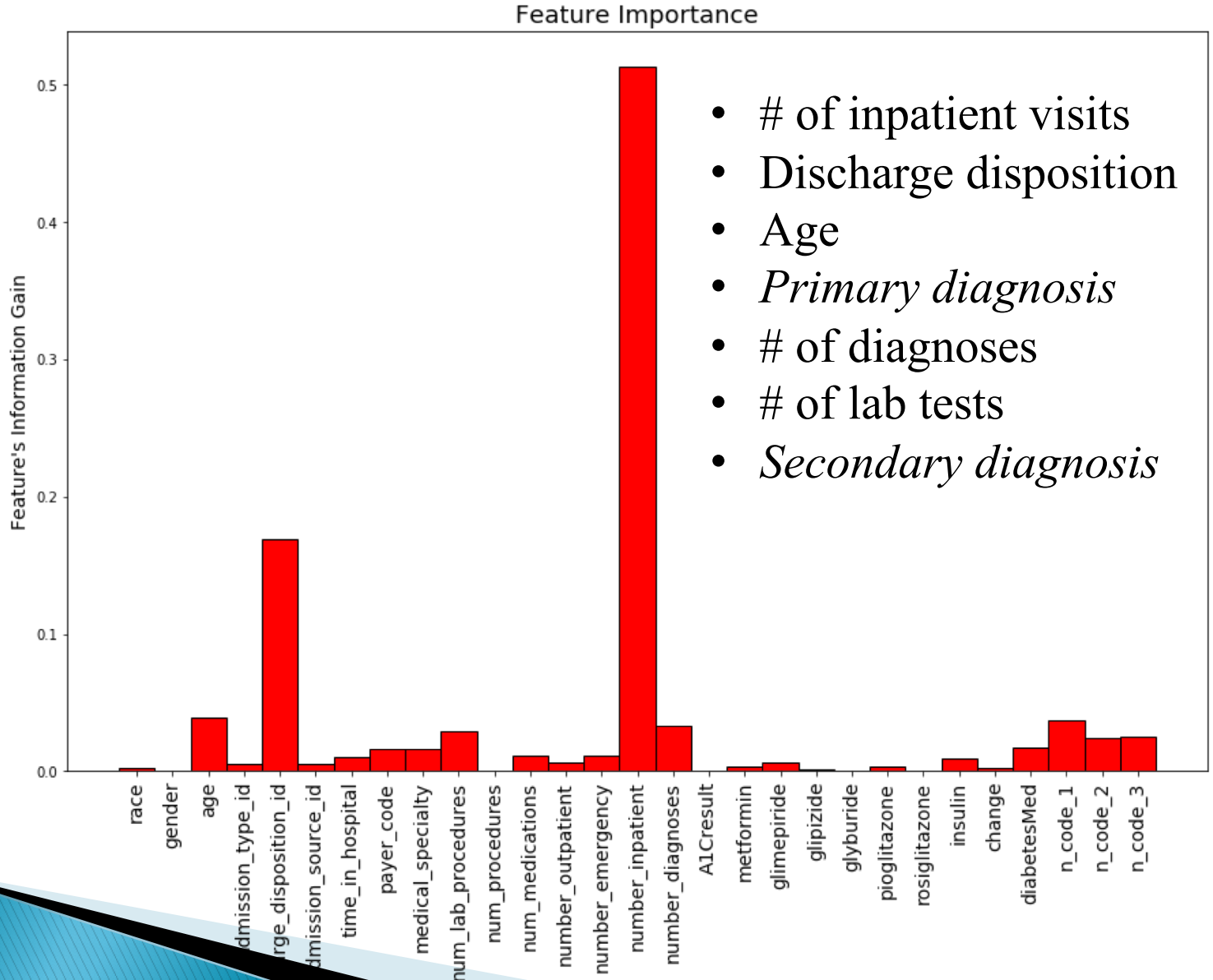


Model #3 (Regularized ROC AUC, Down)



- # of inpatient visits
- Discharge disposition
- Age
- # of diagnoses
- *# of medications*
- # of lab tests
- *Time in hospital*

Model #5 (Regularized PR AUC, Up)



- # of inpatient visits
- Discharge disposition
- Age
- *Primary diagnosis*
- # of diagnoses
- # of lab tests
- *Secondary diagnosis*

Next Steps (1)

- Understand where your last model makes mistakes
- Investigate depths around 7 (i.e., depth=6, 8, and 9)
- DTs do not "know" how to classify observations not seen in training data. Try cross-validation where we use all slices of data to train models
- Consider implementing AdaBoost where models would be trained in sequence to improve the performance of weak learners
- Could try a random forest model (though random forests are better for dealing with overfitting and reducing variance)

Next Steps (2)

- Consider keeping original categories for diagnosis and medication variables (more granular)
  - Introduced many missing values when reduced cardinality of diag_1, diag_2, and diag_3
- Consider re-estimating the model where the target is 1 if readmitted within 30 days and 0 if NOT readmitted
  - Drop the readmitted after 30 days category
- Try a different set of (still relatively simple) models like a logistic regression or SVM (need to pre-process data)

Next Steps (3)

- More involved: Go to the original data source and see if can obtain more information: e.g., encounter date, hospital, patient's zip code, procedures performed, more detailed provider characteristics, etc.
- Investigate the number of previous inpatient encounters