Predicting Hospital Readmission in Diabetic Patients

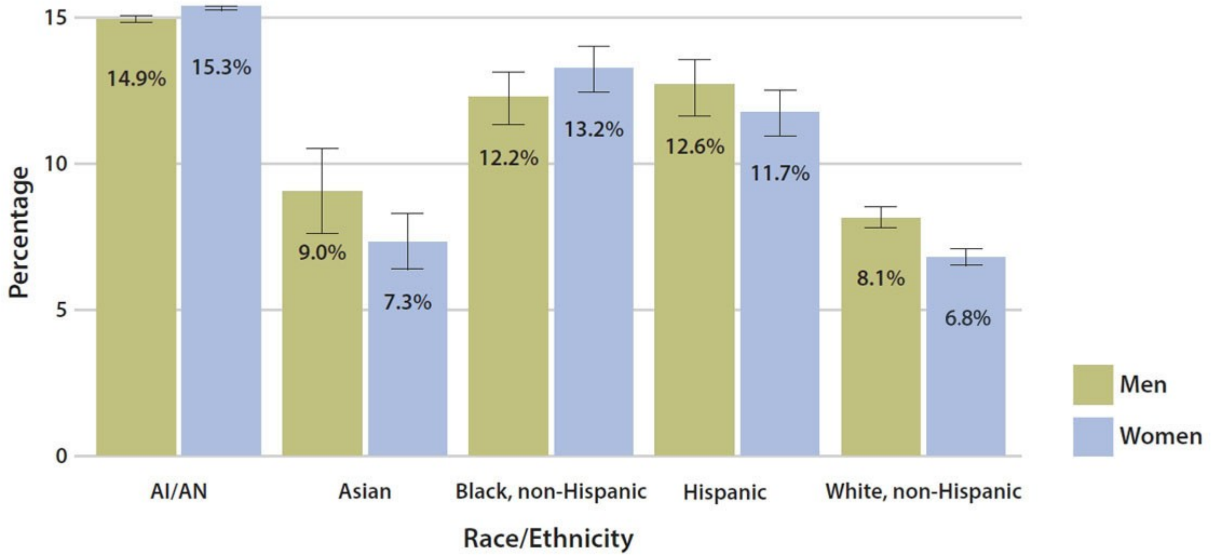
How can machine learning improve individualized medical care?

Derek Tolbert

Background

- As of 2015, 30.3 million Americans had diabetes
- 84.1 million Americans have prediabetes, a strong marker of future diabetes

Age-adjusted Diabetes Prevalence by Race/Ethnicity



Source: New CDC report: More than 100 million Americans have diabetes or prediabetes

Background- The Cost

- In 2012, the total estimated cost of diagnosed diabetes was $245 billion
- Average medical expenditures per person w/ diabetes ~ $13,700/year
- 2.3 times higher costs than non-diabetic Americans

Source: CDC: Deaths and Cost of Diabetes

Research question

- *Can machine learning be used to optimize patient care to reduce the costs of diabetes?*

  Project Goals:

  – Identify if ML algorithms can predict hospital readmission in diabetic patients

  – Identify features that are important in identifying these patients

  – Develop a model which accurately predicts readmission and identify it’s strengths/weaknesses

Process Outline

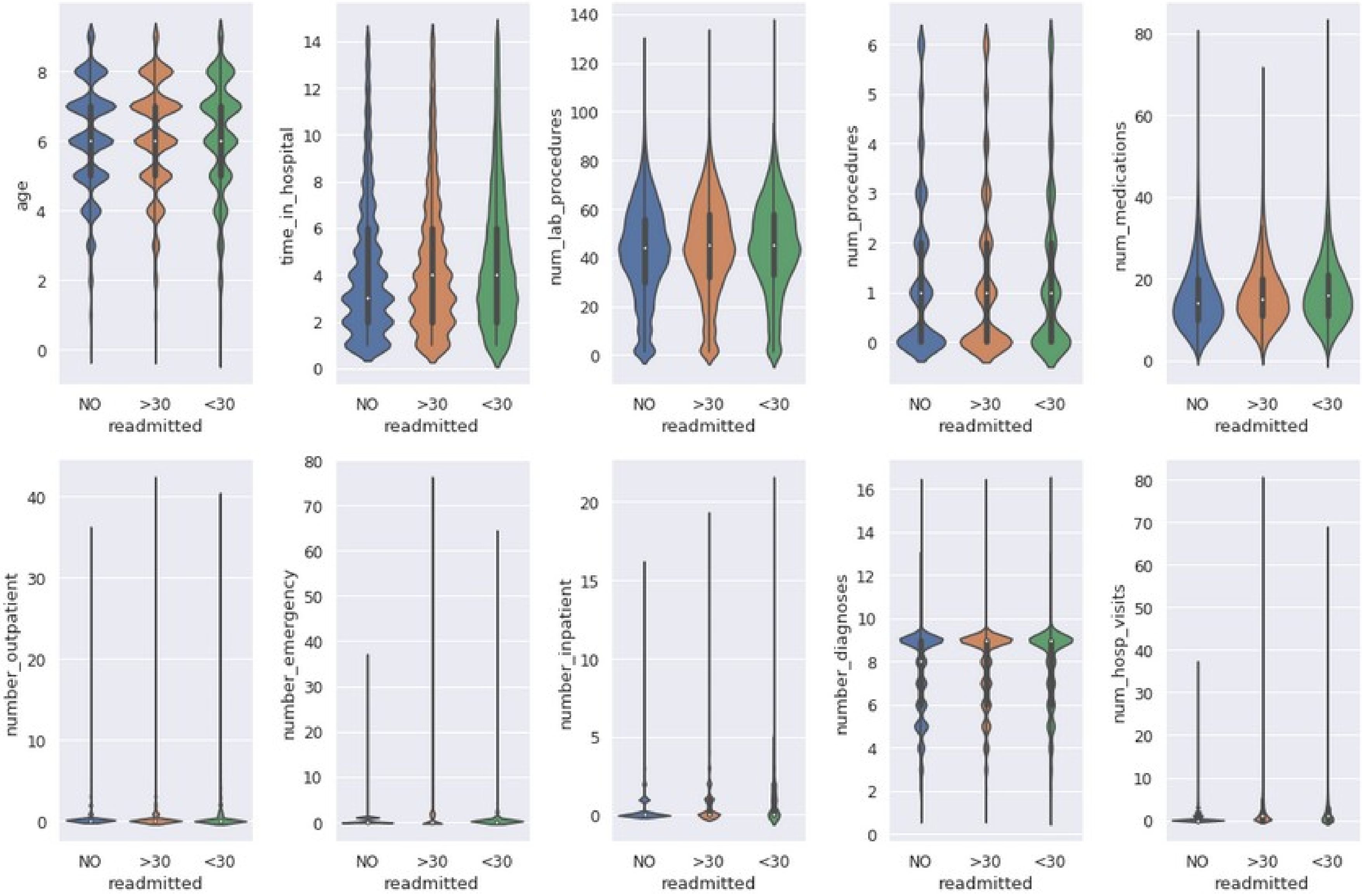
1. Data cleaning and exploration
2. Feature engineering
3. Initial modeling and feature selection
4. Model exploration and selection
5. Tuning hyperparameters and maximizing performance
6. Final model evaluation

Dataset

- n = 101766 visits
- 50 raw features (8 numerical, 42 categorical)
- Taken from 130 US hospitals
- 10 years of data (1999-2008)
- Inclusion criteria
  - Visits included must be a diabetic encounter
  - Lab tests must have been performed
  - Medications must have been administered

Source- UCI Machine Learning Repository

Changes in numerical features across readmission groups



Data cleaning

- Drop variables with > 50% missing data
- Weight, payer_code, and medical_specialty were dropped
- race continues to have 2% missing data
- diag_3 continues to have 1% missing data

Feature Engineering

Key problems:

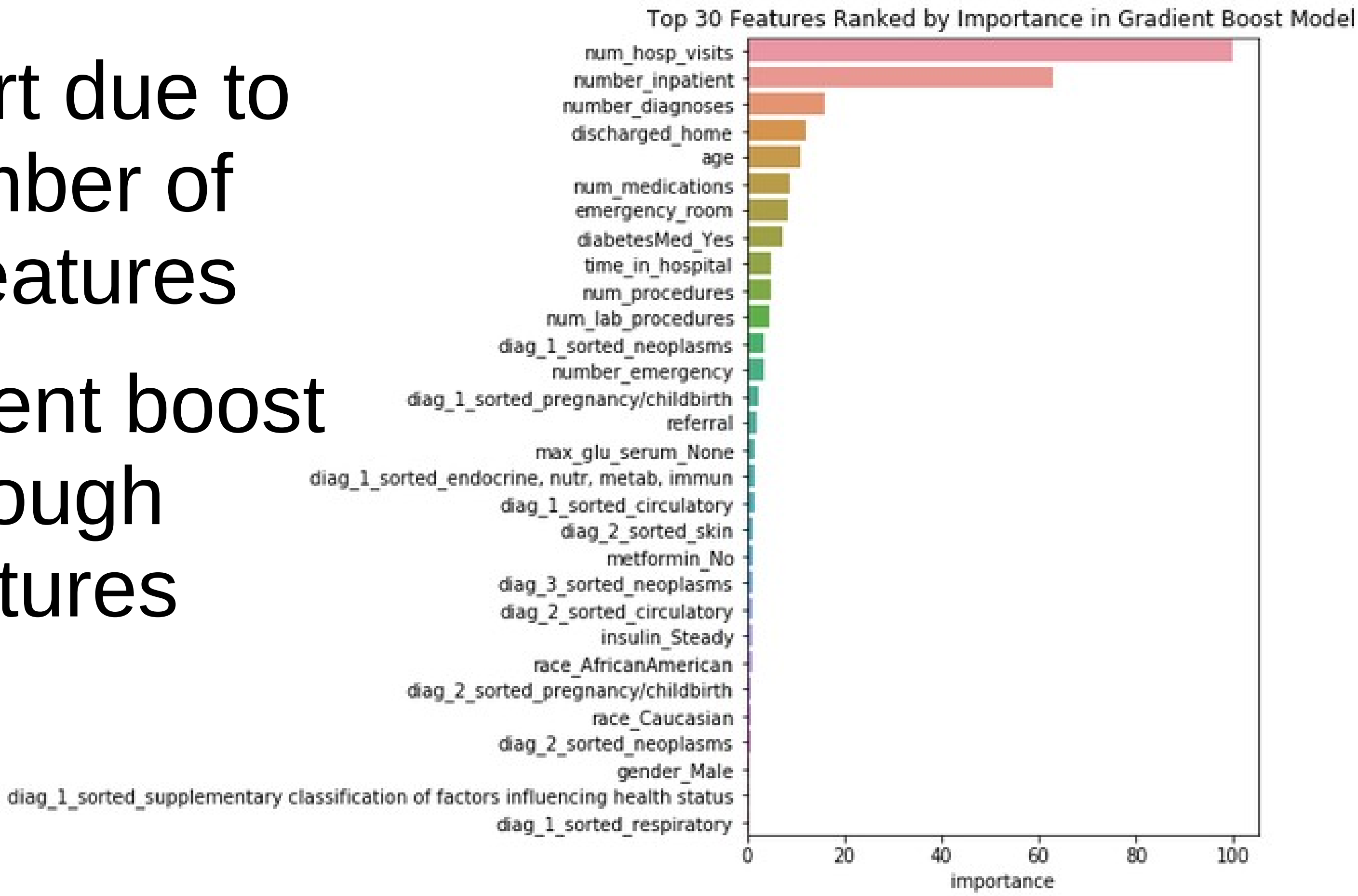
- – Large numbers of groups in the categorical features
  - i.e. 954 distinct values for diag_3
- – Some features may provide more information to our model in a different format

Feature Engineering

- Ages- from nominal to ordinal (80-90 to 8, 90-100 to 9, etc.)
- Diagnoses to broader groups using IC9 codes
- Total hospital visits
- Discharged home
- Admitted from emergency room
- Admitted from referral
- Binary readmission
- Binary readmission < 30 days

Feature Selection

- Difficult to sort due to the large number of categorical features
- Used a gradient boost model for a rough sorting of features



Feature Selection

Model	Feature set	Accuracy
Gradient Boost 1	127 features	Train set: 58.5% Test set: 58.5%
Gradient Boost 2	20 features	Train set: 57.3% Test set: 56.8%

We were able to preserve most of the accuracy of the dataset with only the top 20 features in the model

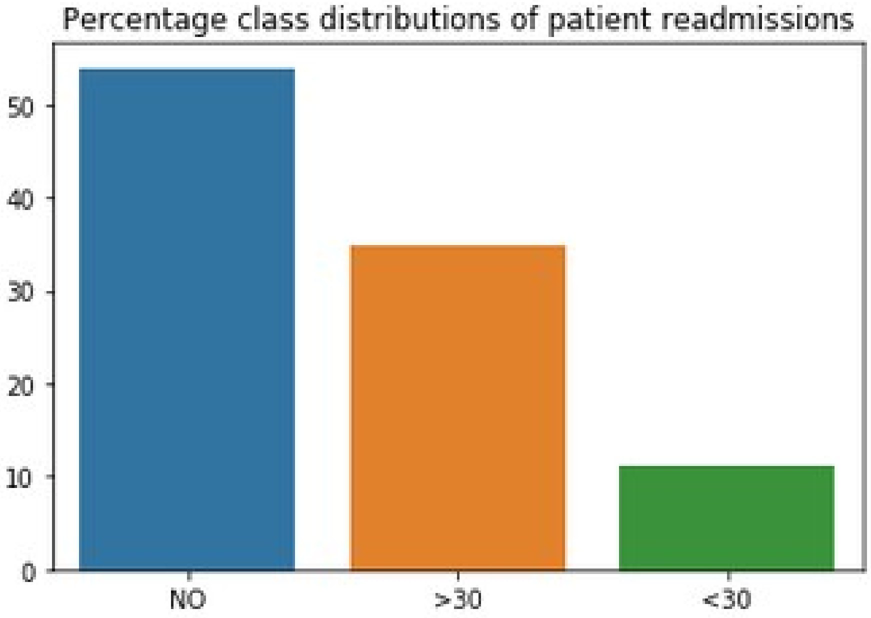
Model exploration

Purpose

- Identify models which will give the best performance in our dataset

Models to test

- Gradient boost
- KNN
- SVM
- Random Forest
- Decision Tree



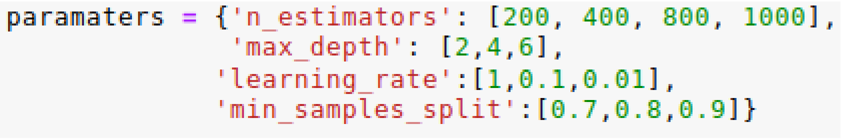
Model Exploration

Model	**Mean accuracy across 10 CV folds**	**Accuracy range across 10 CV folds**
Null accuracy (assuming all patients are not readmitted)	53.9%	0%
Gradient Boost	Train set: 57.3% Test set: 56.8%	Train set: 4.9% Test set: 5.1%
KNN	Train set: 48.3% Test set: 49.2%	Train set: 3.9% Test set: 11.1%
SVM	Train set: 54.9% Test set: 54.1%	Train set: 3.6% Test set: 5.9%
Random Forest	Train set: 54.8% Test set: 55.8%	Train set: 4.3% Test set: 7.6%
Decision Tree	Train set: 45.4% Test set: 45.4%	Train set: 4.5% Test set: 8.1%

Tuning Hyperparameters

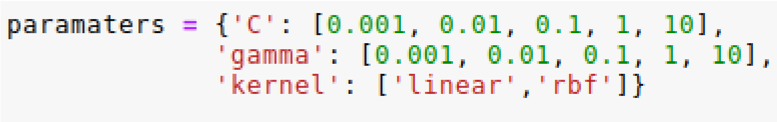
GridSearchCV

– Gradient Boost



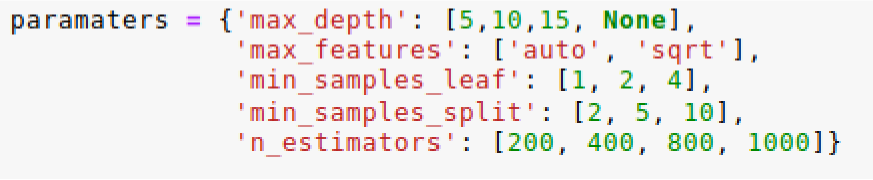
```
paramaters = {'n_estimators': [200, 400, 800, 1000],
               'max_depth': [2,4,6],
              'learning_rate':[1,0.1,0.01],
              'min_samples_split':[0.7,0.8,0.9]}
```

– SVM



```
paramaters = {'C': [0.001, 0.01, 0.1, 1, 10],
             'gamma': [0.001, 0.01, 0.1, 1, 10],
             'kernel': ['linear','rbf']}
```

– Random Forest



```
paramaters = {'max_depth': [5,10,15, None],
              'max_features': ['auto', 'sqrt'],
              'min_samples_leaf': [1, 2, 4],
              'min_samples_split': [2, 5, 10],
              'n_estimators': [200, 400, 800, 1000]}
```

Model Selection

Model	Mean accuracy across 10 CV folds	Accuracy range across 10 CV folds
Null accuracy (assuming all patients are not readmitted)	53.9%	0%
Gradient Boost	Train set: 56.9% Test set: 57.5%	Train set: 1.1% Test set: 1.9%
SVM	Train set: 56.1% Test set: 56.8%	Train set: 2.3% Test set: 5.7%
Random Forest	Train set: 57.5% Test set: 57.8%	Train set: 1.2% Test set: 1.6%

These are the results of running the models with the parameters found using GridSearchCV.

Back to the Drawing Board

- So far none of the models have performed very well
- To simplify the classification problem, I decided to have the model classify the binary readmission variables
- Additionally I increased the number of features from 20 to 40

Simplifying the Classification

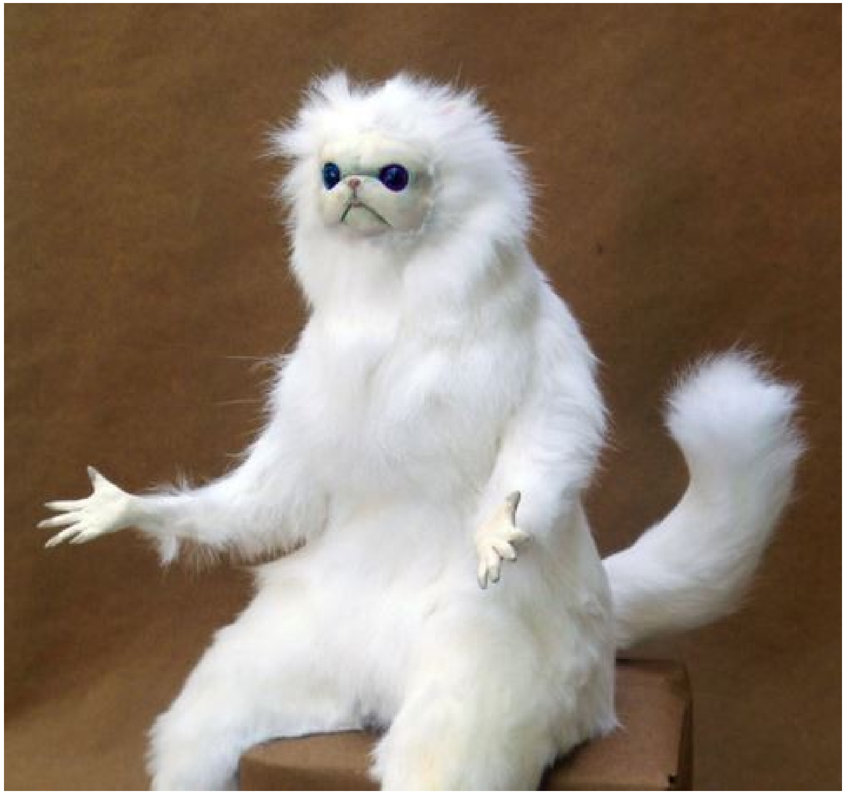
Model	Mean accuracy across 10 CV folds	Accuracy range across 10 CV folds
Null for binary_readmitted	65.1%	-
RF for binary_readmitted	Train set: 62.3% Test set: 62.9%	Train set: 1.5% Test set: 2.6%
Null for binary_readmitted30	88.8%	-
RF for binary_readmitted30	Train set: 88.7% Test set: 89.2%	Train set: 0.01% Test set: 0.04%

While our overall accuracy is best when only classifying for the patients that will be readmitted within 30 days, our model is barely performing better than chance.

Retuning the Random Forest

- GridSearchCV using both accuracy and sensitivity scores
- This improved sensitivity from 0% to 0.5%… ouch

Time to try resampling the dataset



Resampling and Final Model Evaluation

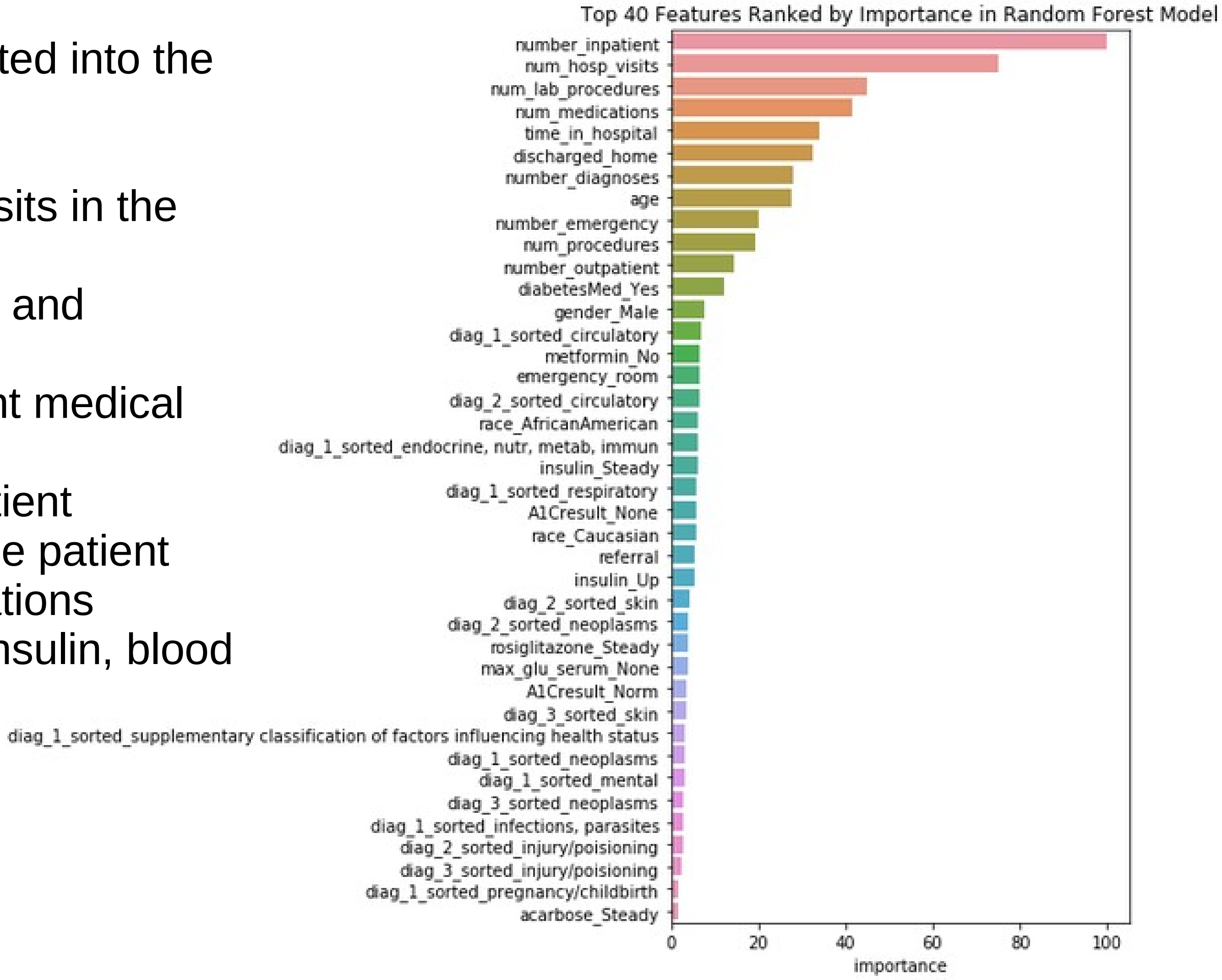
Resampling Technique	Mean accuracy across 10 CV folds	Accuracy range across 10 CV folds	Sensitivity and Precision in Test
Oversampling the minority class	Train set: 64.4% Test set: 89.3%	Train set: 2.1% Test set: 0.2%	Sensitivity: 0.56 Precision: 0.16
Undersampling the majority class	Train set: 60.7% Test set: 89.3%	Train set: 1.7% Test set: 0.1%	Sensitivity: 0.62 Precision: 0.16
Creating synthetic samples of minority class (SMOTE)	Train set: 78.9% Test set: 89.3%	Train set: 30.8% Test set: 0.1%	Sensitivity: 0.25 Precision: 0.14

The model trained on undersampled data was our best overall performer. This model was able to correctly classify 89.3% of the data in all diabetic patients; however, it was unable to classify 38% of the patients who were readmitted within 30 days.

Final Feature Evaluation

Features can be broadly aggregated into the following groups:

1. Number and type of hospital visits in the previous year
2. Number of medical procedures and medications
3. The length of time of the current medical visit
4. The discharge status of the patient
5. The age, race and gender of the patient
6. Medical diagnoses and medications
7. Diabetic testing results (A1C, insulin, blood glucose, etc.)



Summary

- *Can machine learning be used to optimize patient care to reduce the costs of diabetes?*
  - Certainly, this model shows that ML algorithms can be used to predict patients that will be readmitted to the hospital. This can alert providers to try an address potential issues before they arise.
- Future research should investigate additional features to find stronger predictors of readmittance