Data Processing and Modeling Overview

Removed and Modified Features

- Initial shape – 101,767 observations, 50 features

- Weight – 97% missing, removed from analysis
- Payer Code – 40% missing, removed

- Admission type/source and disposition (release) type:
  - Numerically-coded columns; curated and condensed into categories
  - Removed patients who died in the hospital

- Patient diagnosis codes:
  - Series of numeric codes XXX.xx. Converted these to categorical diagnosis types (respiratory, neurological, oncology, injury, etc)

- 24 medications – removed 7 with essentially no values.
- Converted [‘No’, ‘Steady’, ‘Up’, ‘Down’] to [‘No’, ‘Yes’]
  - Is the patient taking this medication or not?

Feature Creation

- Diabetic features:
  - Diabetic patient codes were 250.xx
  - Xx reveals additional info about the patient’s diabetes
    - 250.1 – Diabetes with ketoacidosis
  - Categorized and dummified this information when available

- Diabetic medicine change:
  - Since we converted medicine codes to a simple [Yes, No], created an extra feature, “diabchange”, indicating if the patient changed their dosage of *any* diabetic medication

- Primary diagnosis:
  - Specified the primary diagnosis for which patient was admitted to the hospital

Removal of Train Set Observations

- We have a middle group (returned >30 days to the hospital)
  - These are “No” values for our model, but can offer some information to the model
  - Base AUC with everything: ~0.6662

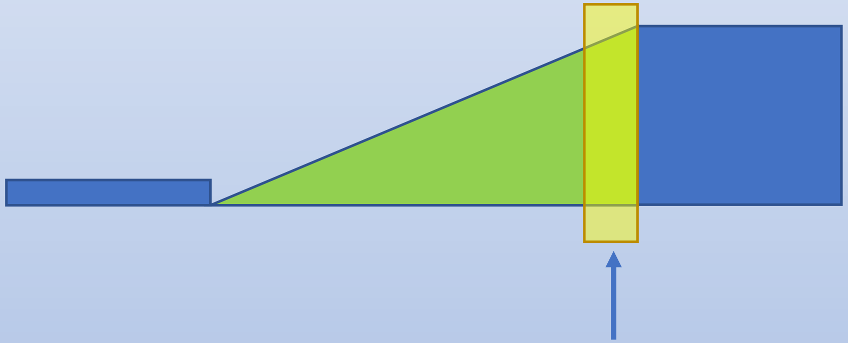
- Removed them from the train set (TrainHL)
  - AUC: 0.6587

- Performed linear regression on TrainHL, added back only middle observations with linear predict<X
  - X = 1.0 – 0.6672
  - **X = 0.75 – 0.6675 -- about 2% of observations removed**
  - X = 0.5 – 0.6661
  - X = 0.25 – 0.6622

No return
=0

Return >30
Linear regression 0-1

Return <30
=1



Returns >30
Very similar to
returns <30
removed from
train set

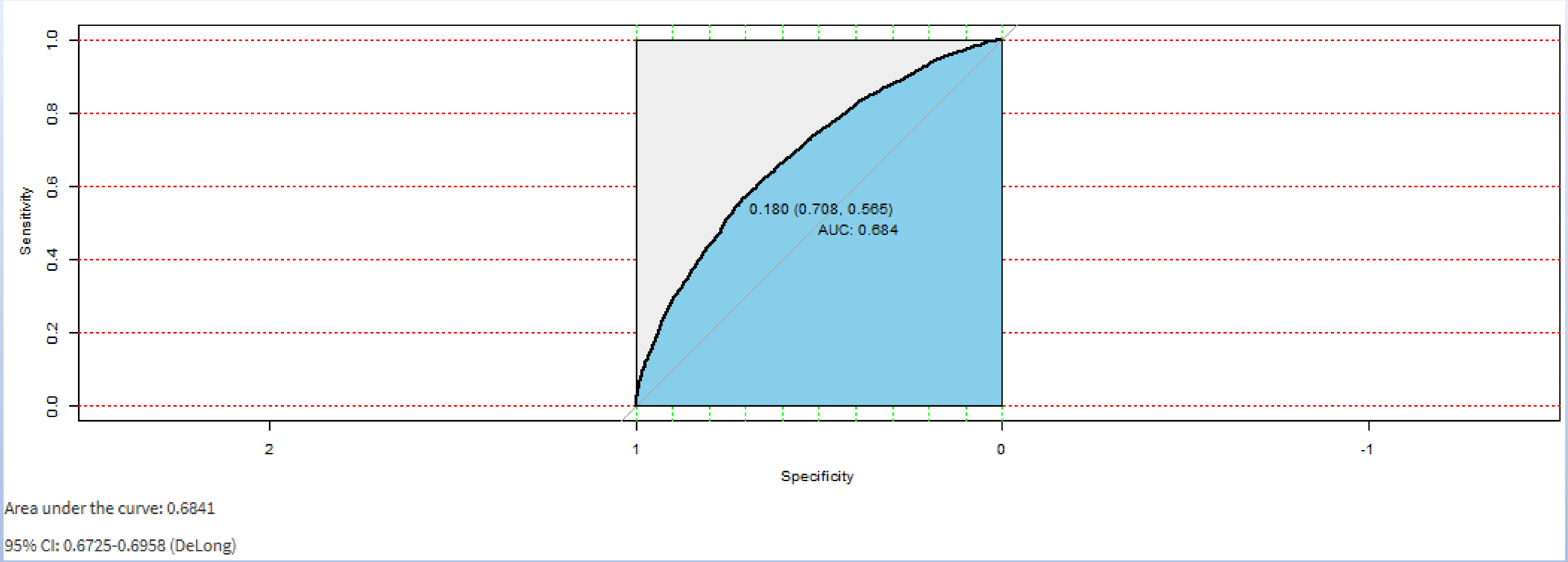
Testing and Feature Selection

- 11% of observations were a “Yes” (patient readmitted < 30 days)
- 3 models used for hyperparameter tuning:
- 5 K-fold random hyperparameter search – AUC scoring
  - Logistic regression – C=0.1, class weights = 0.2/0.8
    - **AUC = 0.668**
  - Random forest – n_trees = 1000, min_samples_split=5, min_samples_leaf=1, max_features = ‘sqrt’, max_depth=60, class_weight=0.2/0.8
    - **AUC = 0.671**
  - XGBoost – n_trees = 500, min_child_wt=10, max_depth=5, gamma=5
    - **AUC = 0.680**

Model Stacking

- AUC optimization – Mapping of true positive vs false positive rate over the range of possible probability threshholds (0.0 – 1.0)
    - AUC/F1 scores are better measurements of performance for rare classifiers than accuracy
        - Max accuracy is to essentially guess “No” for almost everything
- Stacking – Grid search (i, j, k) 0:100
    - Represent percentages of LR, RF, XGB input
- Max AUC score obtained with 56% XGB, 23% RF, 21% LR
    - **AUC = 0.684**
- Used this stacked model to generate probability of return for each patient observation

AUC – Visualization of true/false positive rates at different probability threshholds



Area under the curve: 0.6841

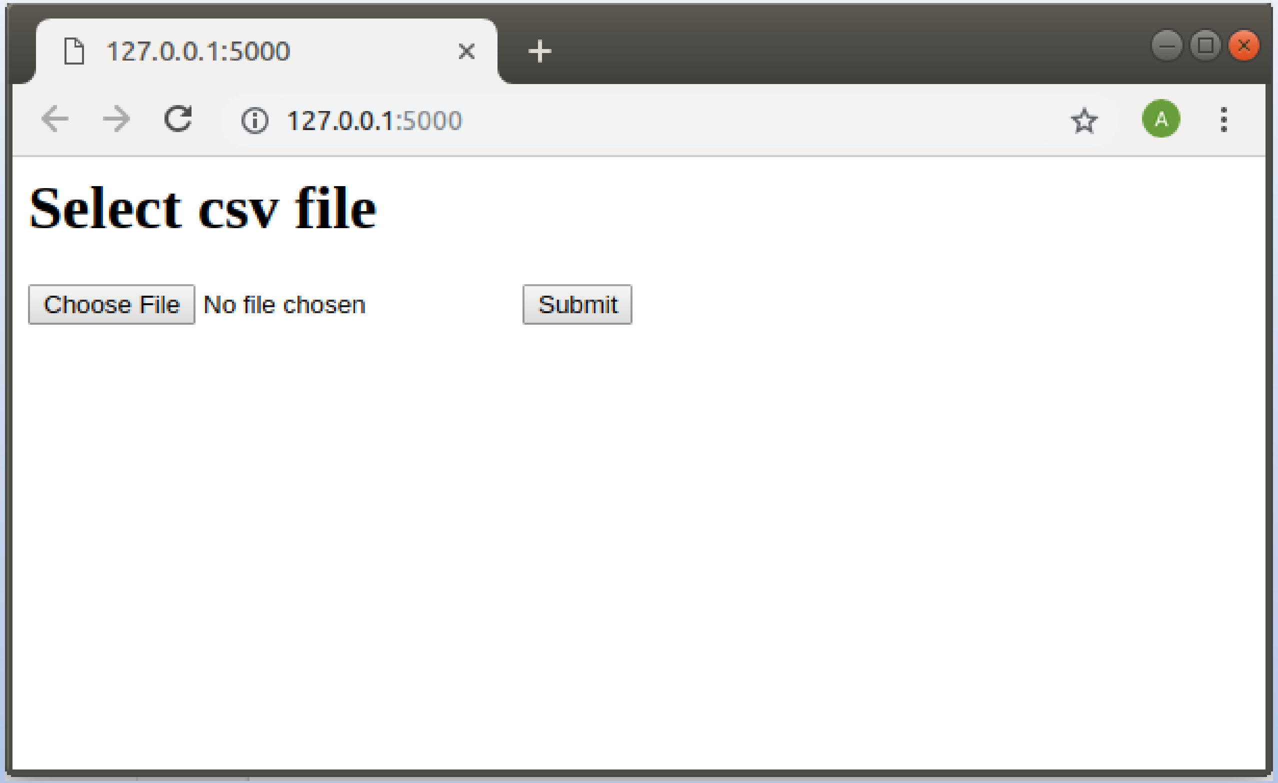
95% CI: 0.6725-0.6958 (DeLong)

- Decision about who to target for a possible readmission reduction plan will depend on various factors, including:
  Resource availability, balance of true/false positives, possible downstream benefits

Flask Pipeline

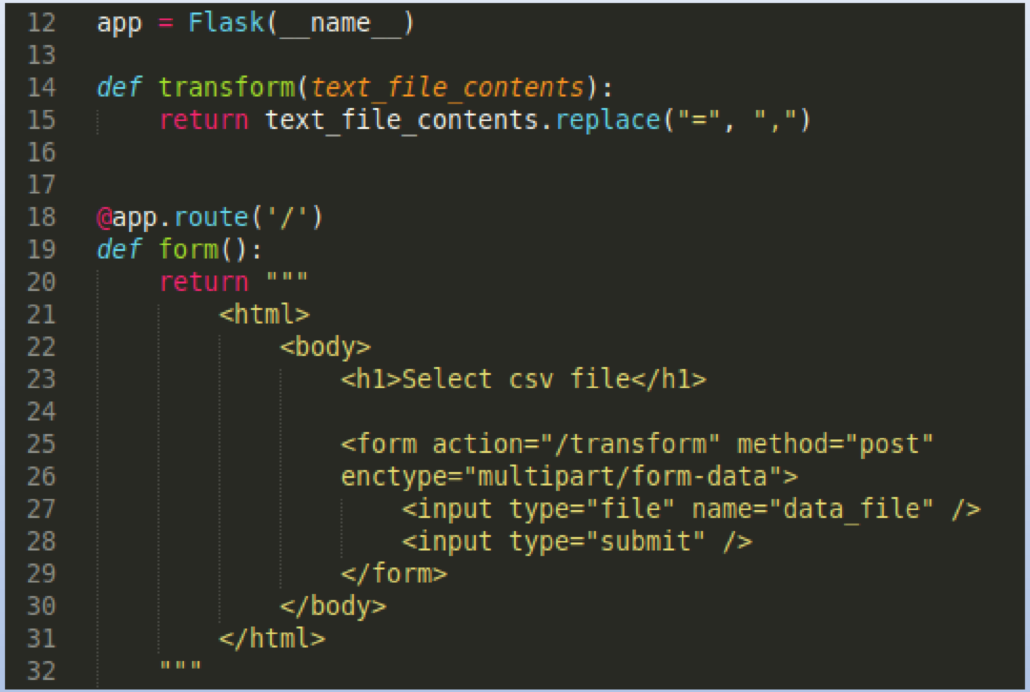
Pipeline

- Input: a csv file with patient information
- Output: the same csv file with the predictions for each patient and probability of patient being readmitted according to stacked model



Select csv file

Flask code

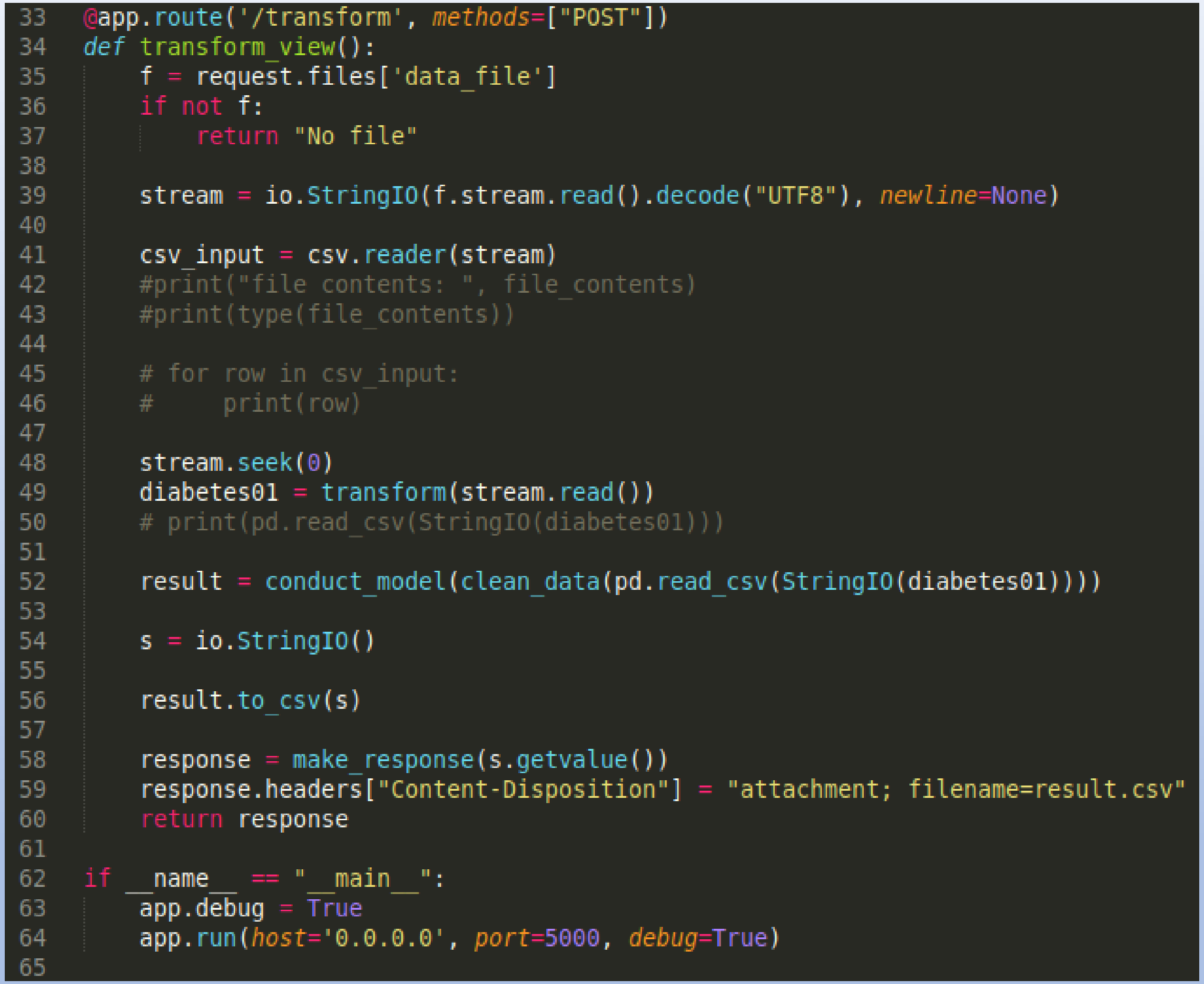


```python
app = Flask(__name__)

def transform(text_file_contents):
    return text_file_contents.replace("=", ",")


@app.route('/')
def form():
    return """
        <html>
            <body>
                <h1>Select csv file</h1>

                <form action="/transform" method="post"
                enctype="multipart/form-data">
                    <input type="file" name="data_file" />
                    <input type="submit" />
                </form>
            </body>
        </html>
    """
```



```python
@app.route('/transform', methods=["POST"])
def transform_view():
    f = request.files['data_file']
    if not f:
        return "No file"

    stream = io.StringIO(f.stream.read().decode("UTF8"), newline=None)

    csv_input = csv.reader(stream)
    #print("file contents: ", file_contents)
    #print(type(file_contents))

    # for row in csv_input:
    #     print(row)

    stream.seek(0)
    diabetes01 = transform(stream.read())
    # print(pd.read_csv(StringIO(diabetes01)))

    result = conduct_model(clean_data(pd.read_csv(StringIO(diabetes01))))

    s = io.StringIO()

    result.to_csv(s)

    response = make_response(s.getvalue())
    response.headers["Content-Disposition"] = "attachment; filename=result.csv"
    return response

if __name__ == "__main__":
    app.debug = True
    app.run(host='0.0.0.0', port=5000, debug=True)

```

Questions?