# Contract Designation Prediction based on Monitoring Student Progress Report (MSPR)

Timothy Abramov

# Monitoring Student Progress Report (MSPR)

- Administered by NCF since Spring 2018
- Instructors fill out a report, whether they have any concerns in regards to Student's Academic Performance
- 6 unique binary concerns (1 or 0): *Attendance*, *Low Participation*, *Late/Missing Assignments*, *Other Assignments Concerns*, *Low Test Scores*, and *Danger of Unsat*

# Objective

Predict student's **Contract Designation** based solely on **MSPR Submissions** to facilitate effective prioritization of students in need of academic help

# Data

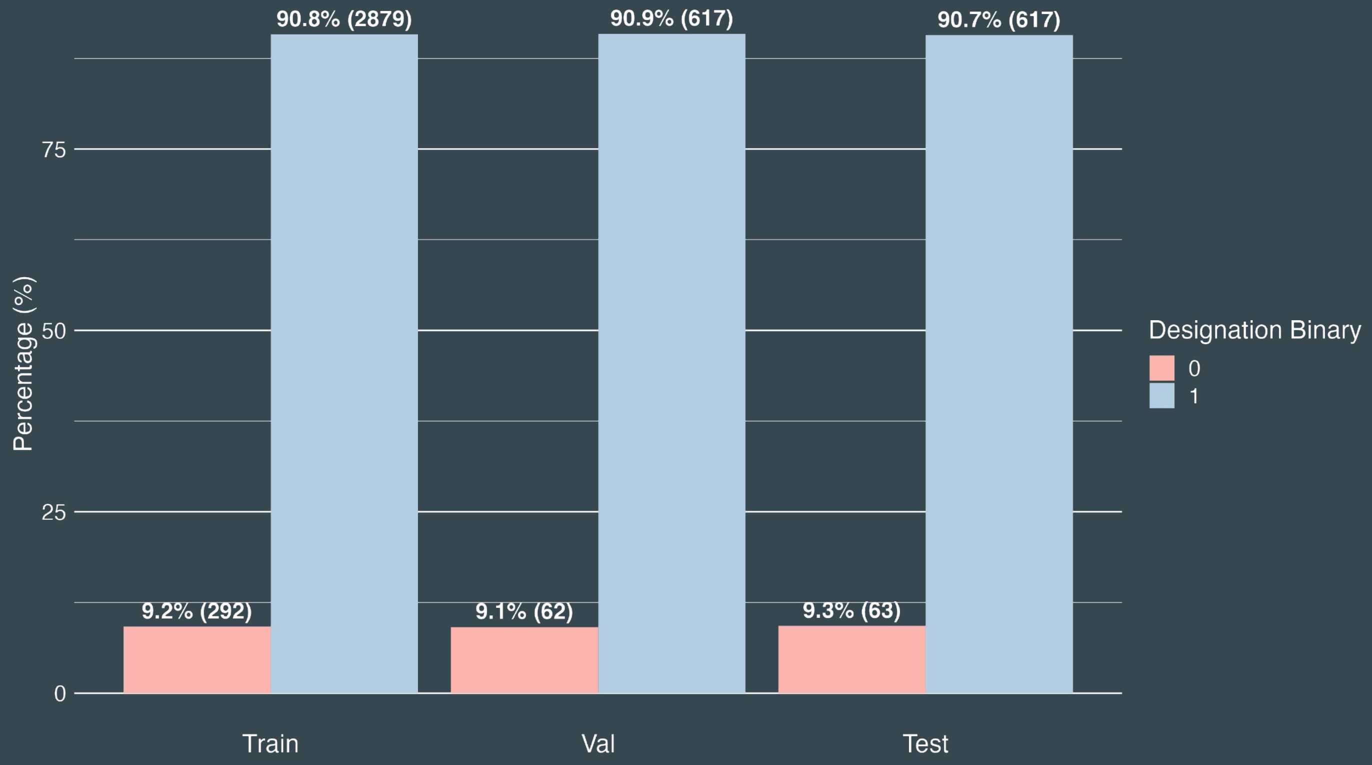
Contract Designation

Unique MSPR forms

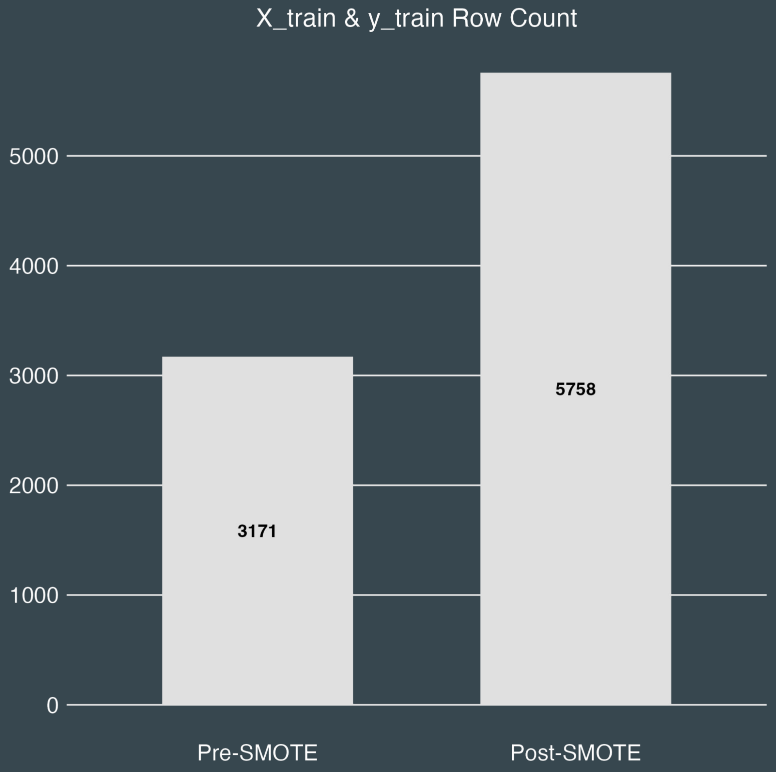
Aggregate Concerns by Student, by Term

| <row> | ATTENDANCE | LOW PARTICIPATION | LATE/MISSING ASSIGNMENTS | OTHER ASSIGNMENTS CONCERNS | LOW TEST SCORES | DANGER of UNSATING | COURSE COUNT | DESIGNATION BINARY |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 2 | 0 | 0 | 1 | 3 | **1** |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | **1** |
| 3 | 1 | 1 | 1 | 2 | 0 | 0 | 4 | **1** |
| … | … | … | … | … | … | … | … | **…** |
| 4530 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | **1** |

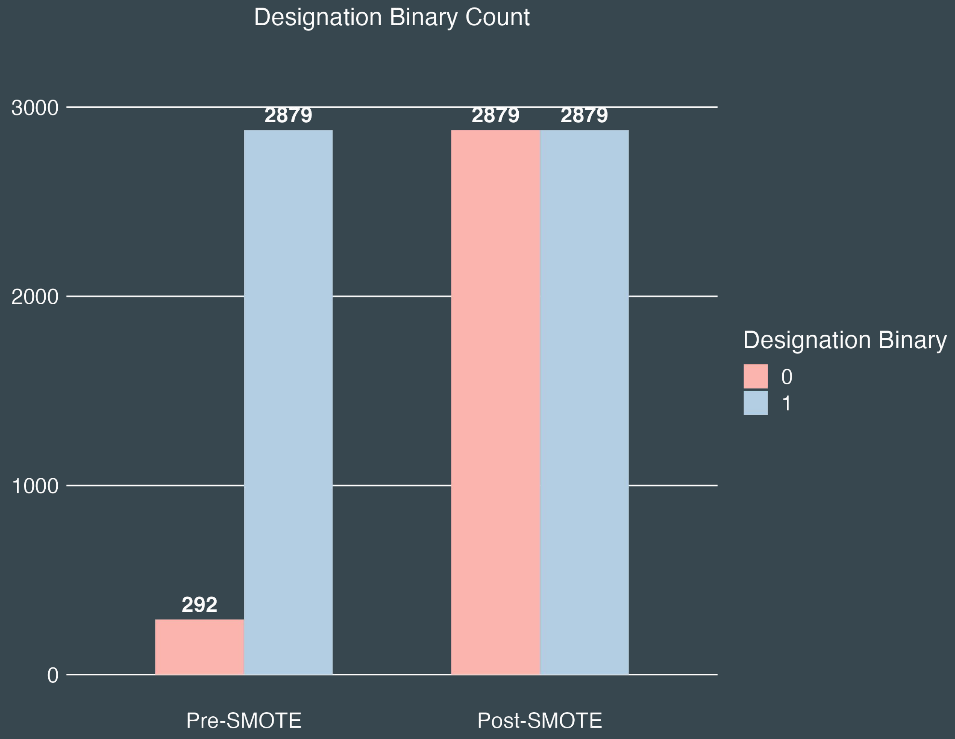
# Stratified Splitting - same label distributions between splits



# SMOTE - balance the label classes



X_train & y_train Row Count

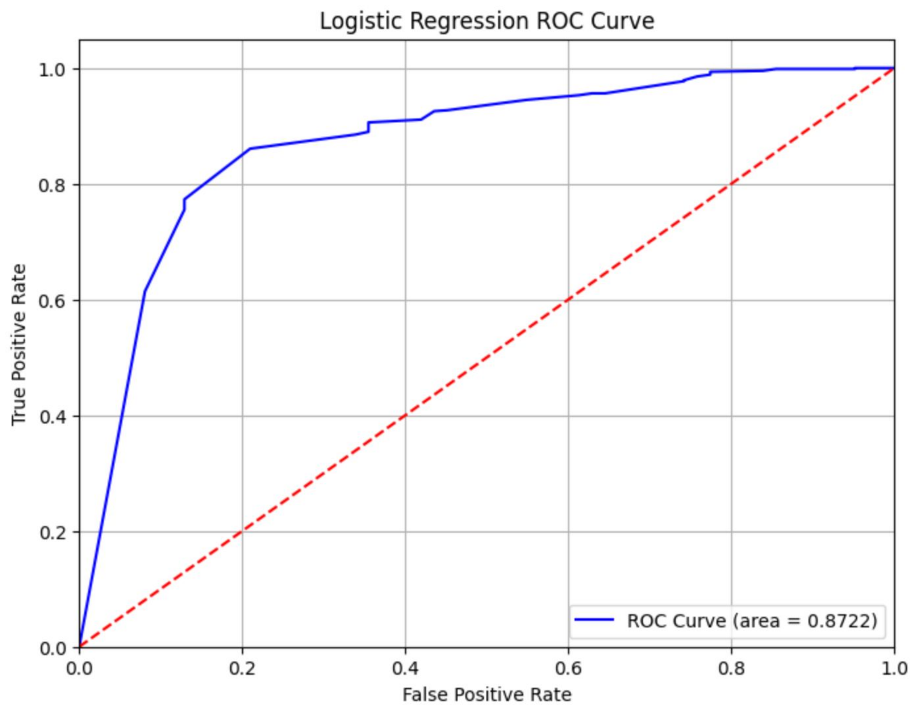


Designation Binary Count

# Baseline Model (Logistic)

| Performance Metric | Validation | Test |
|---|---|---|
| Accuracy | 0.8645 | 0.8618 |
| Precision | 0.9630 | 0.9612 |
| Recall | 0.8849 | 0.8833 |
| F1-score | 0.9223 | 0.9206 |
| ROC AUC | 0.8722 | 0.8390 |

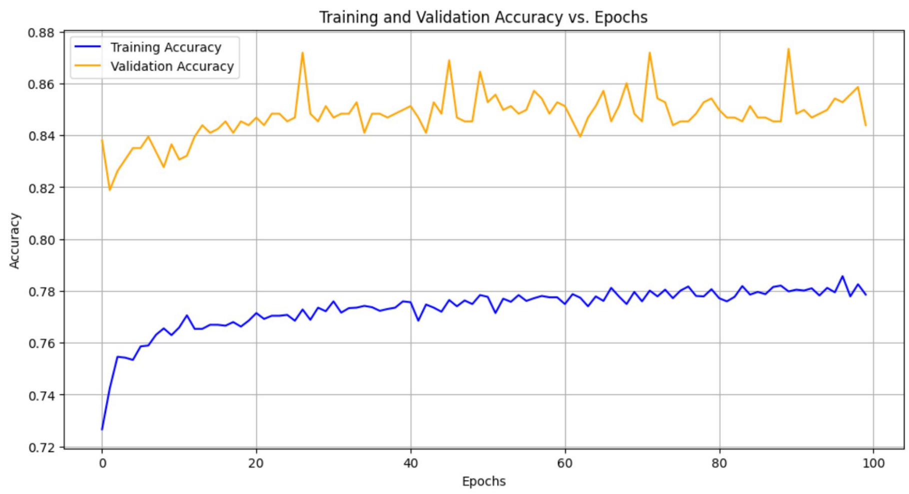
Best hyperparameters:
{C: 0.001, penalty: l1, solver: saga}

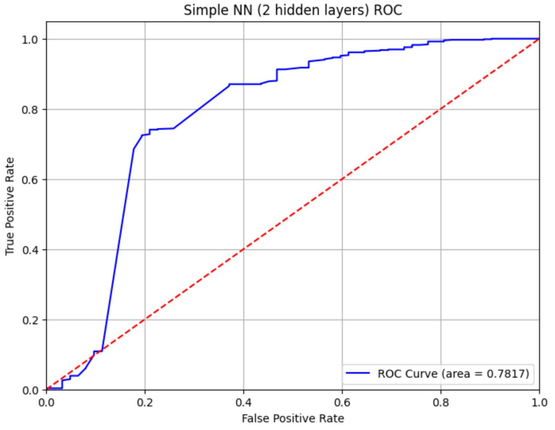


# Simple NN(2 hidden layers)

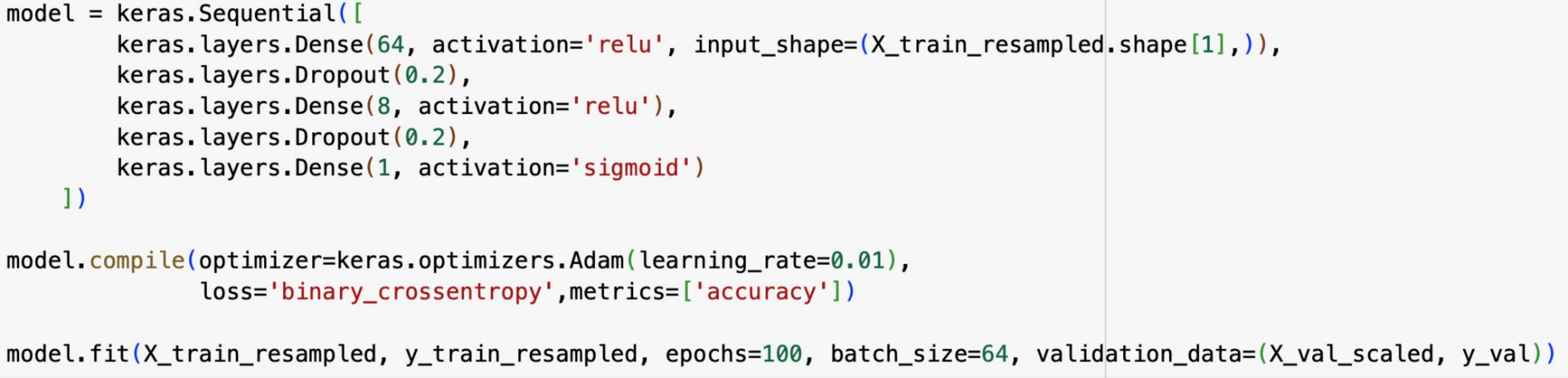
| Performance Metric | Validation | Test |
|---|---|---|
| Accuracy | 0.8439 | 0.8412 |
| Precision | 0.9522 | 0.9504 |
| Recall | 0.8720 | 0.8703 |
| F1-score | 0.9103 | 0.9086 |
| ROC AUC | 0.7817 | 0.7528 |

Best hyperparameters:
{learning rate: 0.01,  dropout: 0.2,
layer1: 64, layer2: 8}





# Simple NN (2 hidden layers) architecture

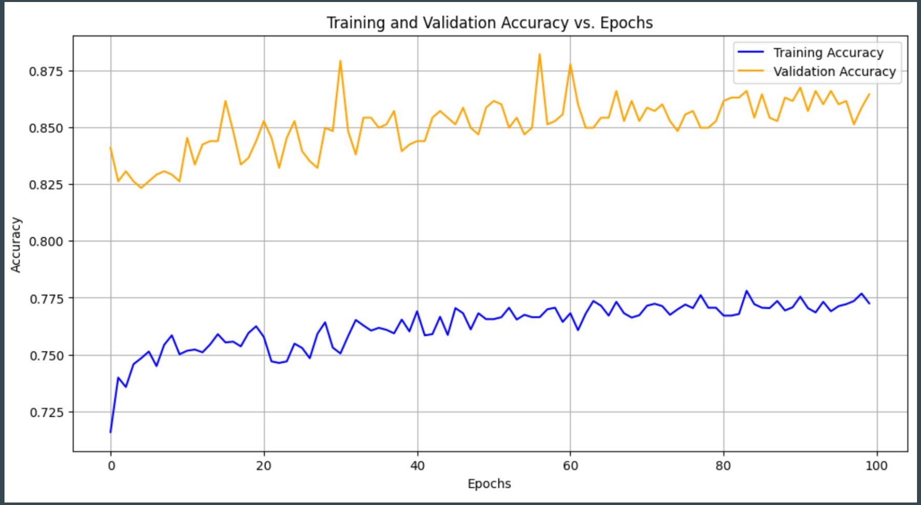


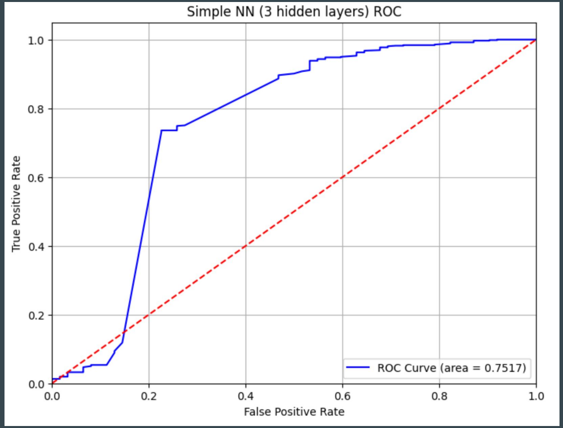
```
model = keras.Sequential([
        keras.layers.Dense(64, activation='relu', input_shape=(X_train_resampled.shape[1],)),
        keras.layers.Dropout(0.2),
        keras.layers.Dense(8, activation='relu'),
        keras.layers.Dropout(0.2),
        keras.layers.Dense(1, activation='sigmoid')
    ])

model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.01),
              loss='binary_crossentropy',metrics=['accuracy'])

model.fit(X_train_resampled, y_train_resampled, epochs=100, batch_size=64, validation_data=(X_val_scaled, y_val))
```

# Simple NN(3 hidden layers)

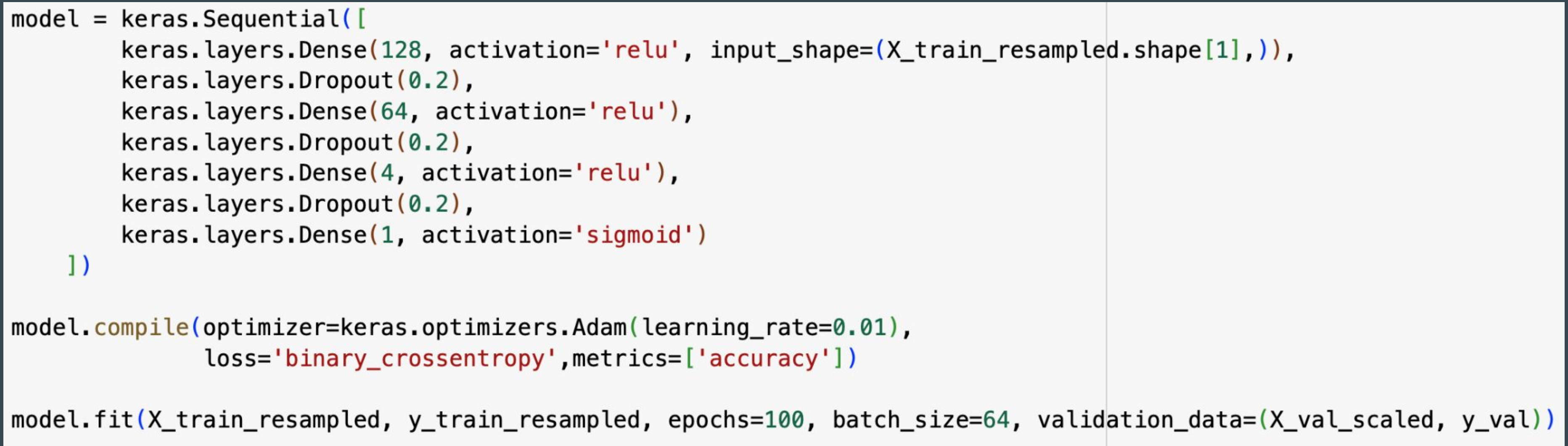
| Performance Metric | Validation | Test |
|---|---|---|
| Accuracy | 0.8645 | 0.8559 |
| Precision | 0.9472 | 0.9482 |
| Recall | 0.9011 | 0.8898 |
| F1-score | 0.9236 | 0.9181 |
| ROC AUC | 0.7517 | 0.7559 |

Best hyperparameters:
{learning rate: 0.01,  dropout: 0.2,
layer1: 128, layer2: 64, layer3 = 4}





# Simple NN (3 hidden layers) architecture



```
model = keras.Sequential([
        keras.layers.Dense(128, activation='relu', input_shape=(X_train_resampled.shape[1],)),
        keras.layers.Dropout(0.2),
        keras.layers.Dense(64, activation='relu'),
        keras.layers.Dropout(0.2),
        keras.layers.Dense(4, activation='relu'),
        keras.layers.Dropout(0.2),
        keras.layers.Dense(1, activation='sigmoid')
    ])

model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.01),
              loss='binary_crossentropy',metrics=['accuracy'])

model.fit(X_train_resampled, y_train_resampled, epochs=100, batch_size=64, validation_data=(X_val_scaled, y_val))
```

# Performance Comparison

| | Test Data Set | | |
|---|---|---|---|
| Performance Metric | Logistic Regression | Simple NN (2 layers) | Simple NN (3 layers) |
| Accuracy | **0.8618** | 0.8412 | 0.8559 |
| Precision | **0.9612** | 0.9504 | 0.9482 |
| Recall | 0.8833 | 0.8703 | **0.8898** |
| F1 | **0.9206** | 0.9086 | 0.9181 |
| ROC AUC | **0.8390** | 0.7528 | 0.7559 |

# Potential Improvements

- More features for the dataset
- More epochs for NN models + early stopping
- Use random seeds to lock down the models
- Don’t split into train-val-test, only train-val?

# Thank You