GENERAL ASSEMBLY

COMMUNICATING RESULTS

Trevor Lindsay

LEARNING OBJECTIVES

- Explain the trade-offs between the precision and recall of a model while articulating the cost of false positives vs. false negatives.
- Describe the difference between visualization for presentations vs. exploratory data analysis
- Identify the components of a concise, convincing report and how they relate to specific audiences/stakeholders

COURSE

PRE-WORK

PRE-WORK REVIEW

- Understand results from a confusion matrix and measure true positive rate and false positive rate
- Create and interpret results from a binary classification problem

OPENING

COMMUNICATING RESULTS

WE BUILT A MODEL…NOW WHAT?

- We’ve built our model, but there is still a gap between your Notebook with plots/figures and a slideshow needed to present your results.

- Classes so far have focused on two core concepts:
  - developing consistent practices
  - interpreting metrics to evaluate and improve model performance

- But what does that mean to your audience?

WE BUILT A MODEL…NOW WHAT?

- Imagine how a non-technical audience might respond to the following statements:
  - The predictive model I built has an accuracy of 80%.
  - Linear regression was optimized by ensuring no multicollinearity
  - Gender was more important than age in the predictive model because it has a larger coefficient.
  - Here’s the confusion matrix that shows how well the model did.

WE BUILT A MODEL…NOW WHAT?

- Who is your audience? Are they technical? What are their concerns?

- Remember: in a business setting, you may be the only person who can interpret what you've built.

- Some people may be familiar with basic visualization, but you will likely have to do a lot of "hand holding".

- You need to be able to efficiently explain your results in a way that makes sense to **all** stakeholders (technical or not).

WE BUILT A MODEL…NOW WHAT?

- Today, we’ll focus on communicating results for “simpler” problems, but this applies to any type of model you may work with.

- First, let’s review classification metrics, review our knowledge, and talk about how we might communicate what we know.

REVIEW

BACK TO THE CONFUSION MATRIX

BACK TO THE CONFUSION MATRIX

- Confusion matrices allow for the interpretation of correct and incorrect predictions for each class label.

- It is the first step for the majority of classification metrics and goes deeper than just accuracy.

BACK TO THE CONFUSION MATRIX

- Let’s recall our confusion matrix

Hypothesized class	True class: p	True class: n
Y	True Positives	False Positives
N	False Negatives	True Negatives
Column totals:	P	N

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

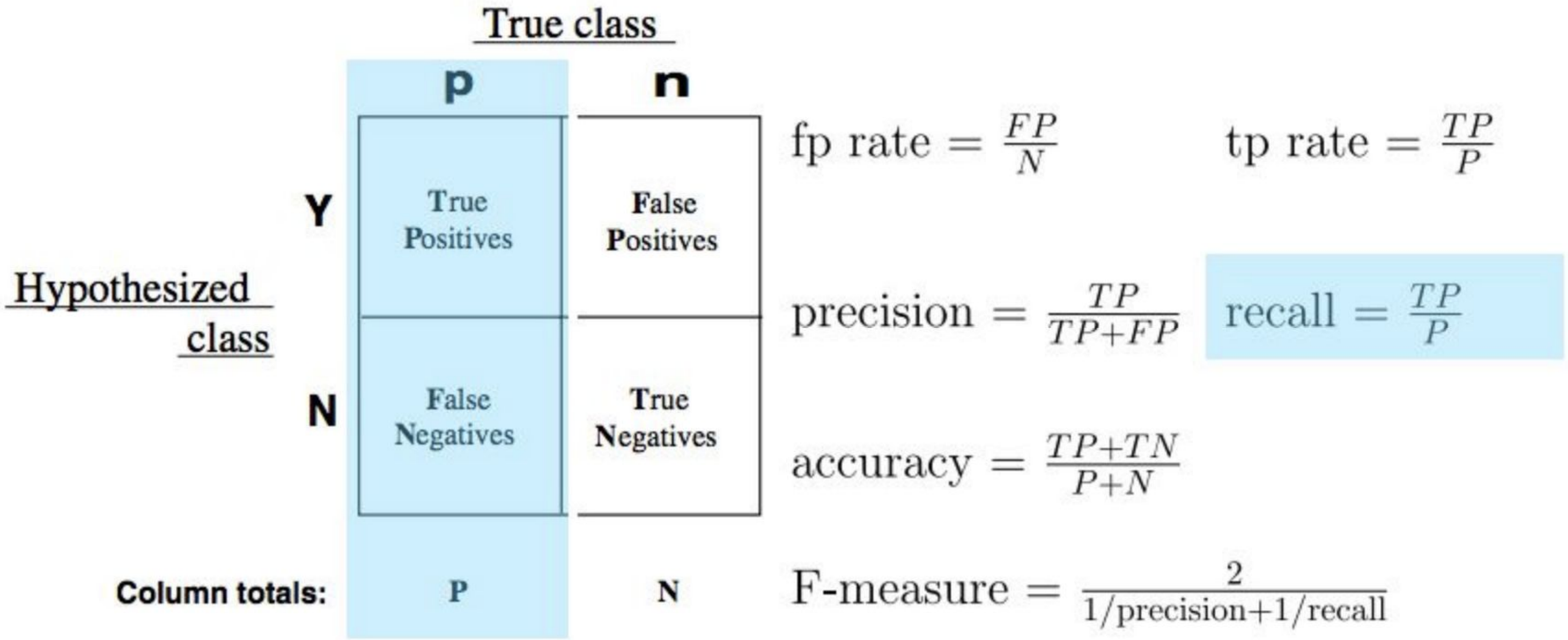
$$\text{accuracy} = \frac{TP+TN}{P+N}$$

$$\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

THE MATH FOR RECALL

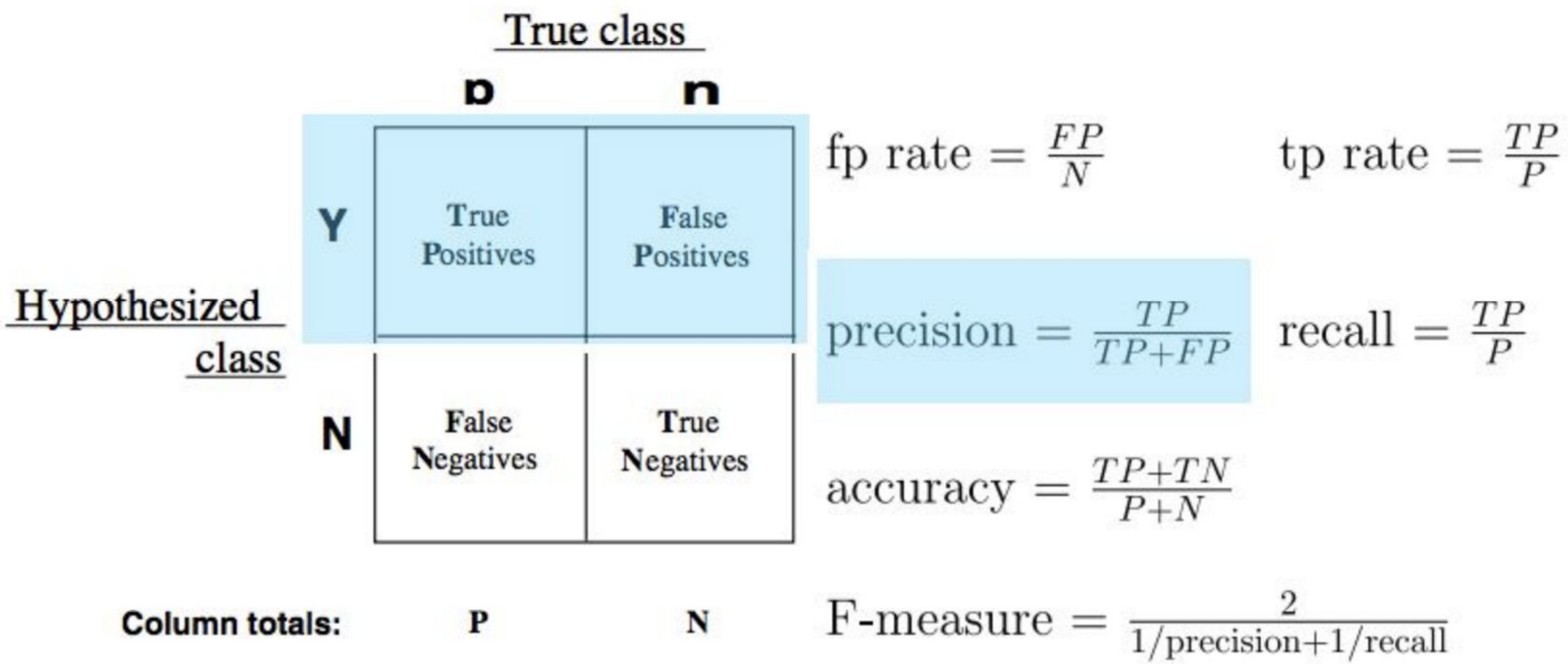
- Recall is the count of predicted true positives over the total count of that class label.

- This is the same as True Positive Rate or sensitivity.

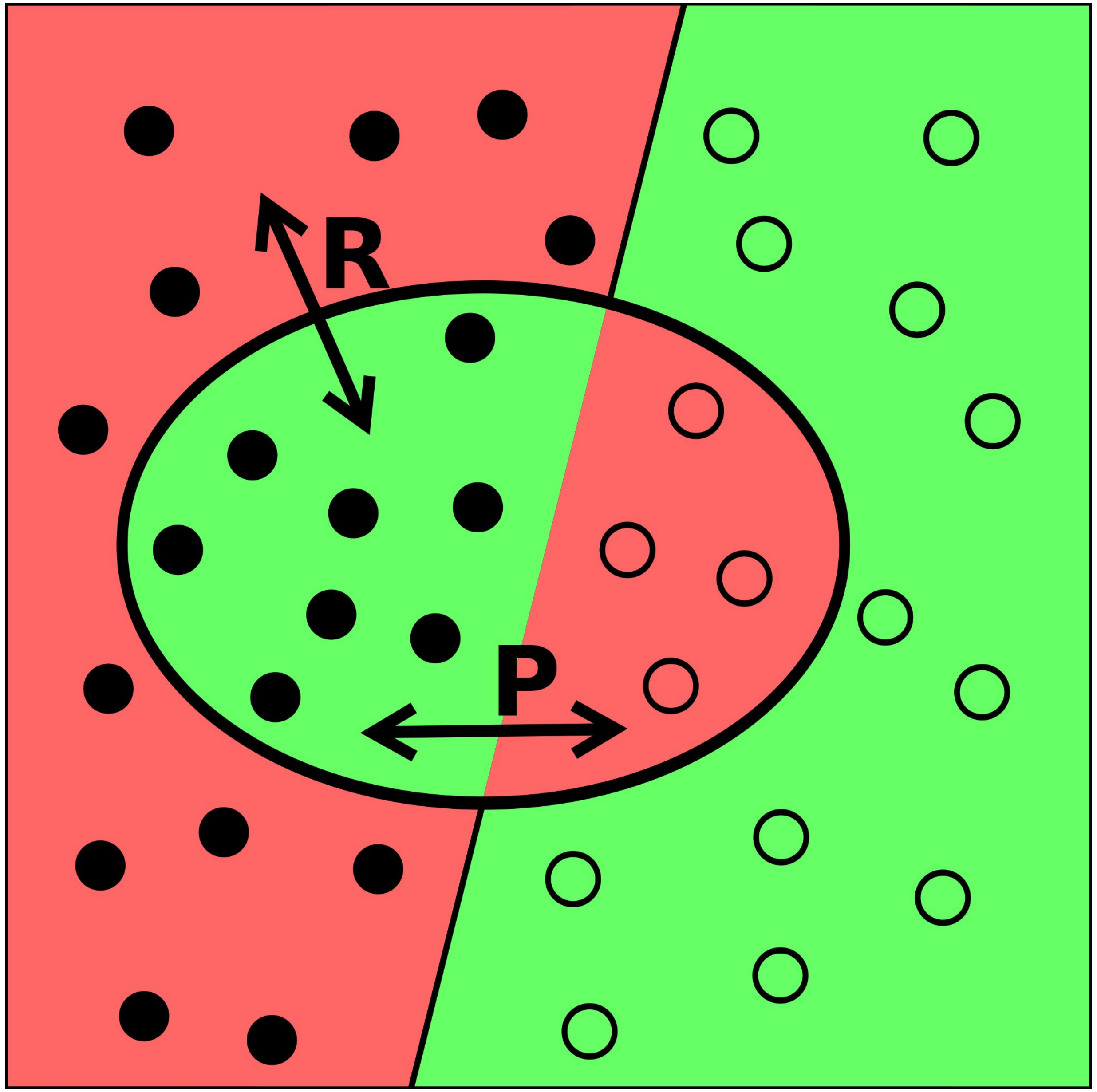


THE MATH FOR PRECISION

- Precision, or positive predicted value, is calculated as the count of predicted true positives over the count of all values predicted to be positive



VISUALIZING PRECISION AND RECALL



DEMO

UNDERSTANDING TRADEOFFS

UNDERSTANDING TRADEOFFS

- Let’s consider the following data problem: we are given a data set in order to predict or identify traits for typically late flights.
- Optimizing toward recall, we could assume that every flight will be delayed.
- The trade-off, a lower precision, is that this could create even further delays, missed flights, etc.

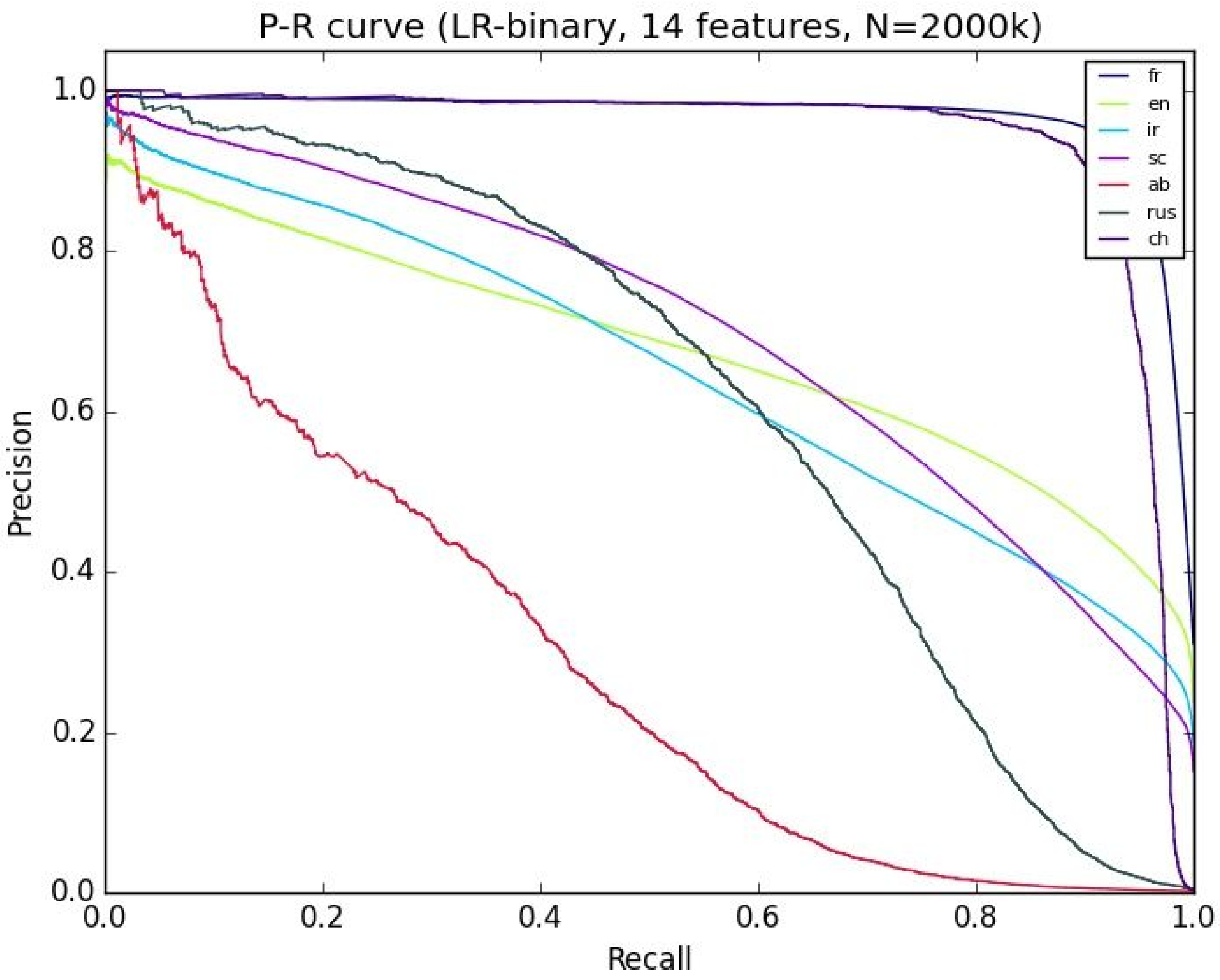
UNDERSTANDING TRADEOFFS

- Optimizing toward precision, we would specifically look to identify flights that will be late.

- The trade-off here would be lower recall. We might miss flights that would be delayed, causing a strain on the system.

PRECISION RECALL CURVE

- Another way to visualize a model and compare between models



SHOWING WORK

SHOWING WORK

- We’ve spent a lot of time exploring our data and building a reasonable model that performs well.

- However, if we look at our visuals, they are most likely:

  - Statistically heavy: Most people don’t understand histograms.

  - Overly complicated: Scatter matrices produce too much information.

  - Poorly labeled: Code doesn’t require adding labels, so you may not have added them.

SHOWING WORK

- In order to convey important information to our audience, make sure our charts are:
  - Simplified
  - Easily interpretable
  - Clearly labeled

SIMPLIFIED

- At most, you'll want to include figures that either explain a variable on its own or explain that variable's relationship with a target.
- If your model used a data transformation (like natural log), just visualize the original data.
- Try to remove any unnecessary complexity.

EASILY INTERPRETABLE

- Any stakeholder looking at a figure should be seeing the exact same thing you're seeing.
- A good test for this is to share the visual with others less familiar with the data and see if they come to the same conclusion.
- How long did it take them?

CLEARLY LABELED

- Take the time to clearly label your axis, title your plot, and double check your scales - especially if the figures should be comparable.

- If you're showing two graphs side by side, they should follow the same Y axis.

QUESTIONS TO ASK

- When building visuals for another audience, ask yourself these questions:
  - **Who:** Who is my target audience for the visual?
  - **What:** What do they already know about this project? What do they need to know?
  - **How:** How does my project affect this audience? How might they interpret (or misinterpret) the data?

DEMO

VISUALIZING MODELS OVER VARIABLES

VISUALIZING MODEL OVER VARIABLES

- One effective way to explain your model over particular variables is to plot the predicted values against the most explanatory variables.

- For example, in logistic regression, plotting the probability of a class against a variable can help explain the range of effect of the model.

VISUALIZING MODEL OVER VARIABLES

- We'll use the flight delay data for all following examples. Let's build our first model and plot.
- https://www.rita.dot.gov/bts/help/aviation/index.html
- Open the a Jupyter notebook and follow along

VISUALIZING MODEL OVER VARIABLES

```
# read in the file and generate a quick model (assume we've done the data exploration already)
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('seaborn')

import sklearn.linear_model as lm

df = pd.read_csv('../../data/flight_delays.csv')
```

VISUALIZING MODEL OVER VARIABLES

```
# Remove rows where target, DEP_DEL15, is null
df = df[df['DEP_DEL15'].notnull()].copy()

# Convert UNIQUE_CARRIER to a dummy variable
df = pd.get_dummies(df, prefix='CARRIER', columns=['UNIQUE_CARRIER'])

# Convert DAY_OF_WEEK to a dummy variable
df = pd.get_dummies(df, prefix='DOW', columns=['DAY_OF_WEEK'], drop_first=True)
```

VISUALIZING MODEL OVER VARIABLES

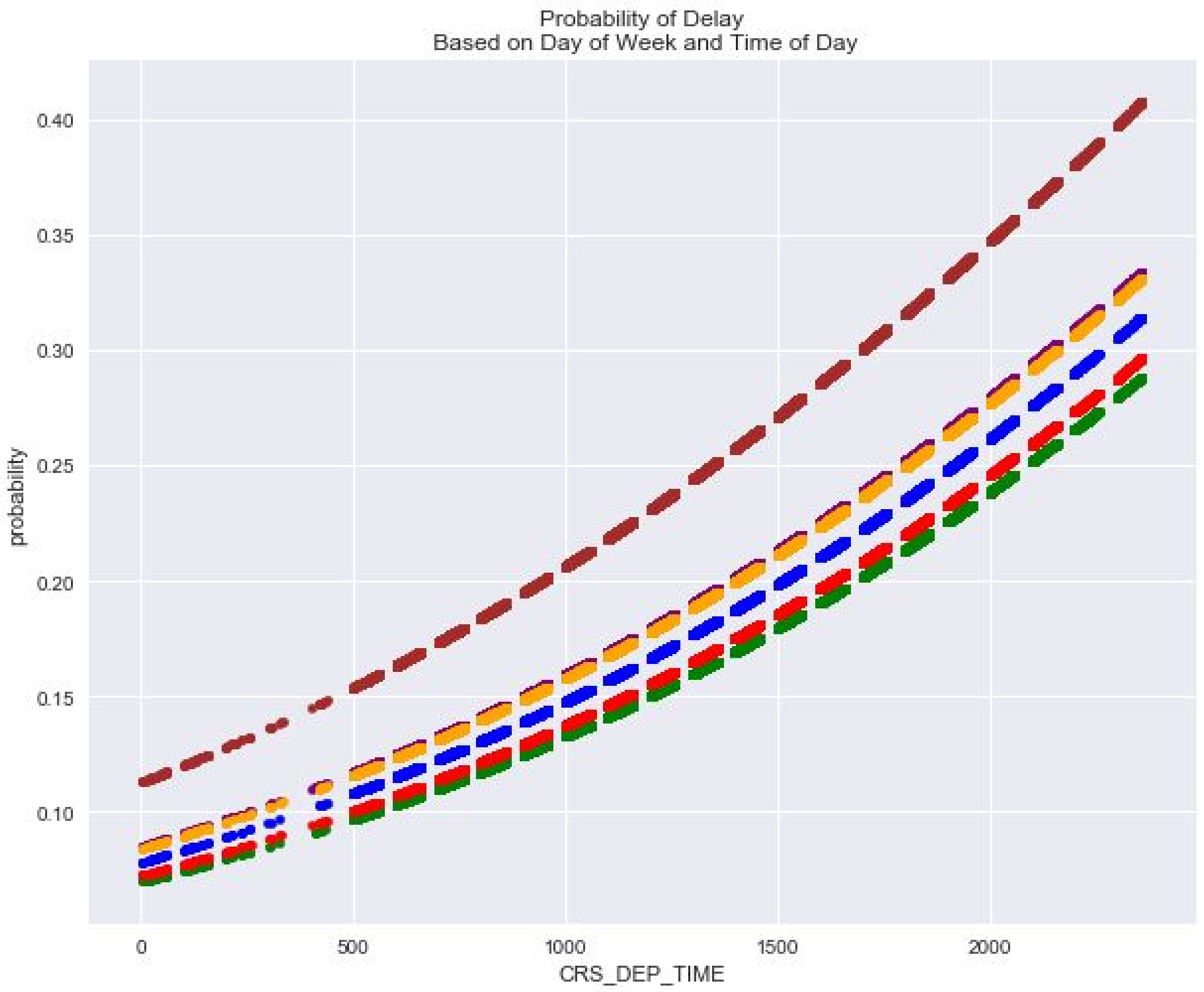
```
# Create features and target set
features = [col for col in df.columns if 'DOW_' in col] + ['CRS_DEP_TIME']
target = 'DEP_DEL15'

X = df[features]
y = df[target]

# Build a model
cls = lm.LogisticRegression()
cls.fit(X, y)

# Predict probability of delay
df['probability'] = cls.predict_proba(X)[:, 1]
```

VISUALIZING MODEL OVER VARIABLES

```
ax = plt.gca()
colors = ['blue', 'green', 'red', 'purple', 'orange', 'brown']
for index, color in enumerate(colors):
    dow = features[index]
    df[df[dow] == 1].plot(
        x='CRS_DEP_TIME',
        y='probability',
        kind='scatter',
        color=color,
        figsize=(10,8),
        ax=ax,
    )
ax.set(title='Probability of Delay\n Based on Day of Week and Time of Day')
```

VISUALIZING MODEL OVER VARIABLES

- This visual can help showcase the range of effect on delays from both day of the week and time of day.
- Given this model, some days are more likely to have delays than others.
- The likelihood of delay increases as the day goes on.



ACTIVITY: TRY IT OUT

- Directions
  - 1. Adjust the model to make delay predictions using airlines instead of day of week, and time, then plot the effect on CRS_DEP_TIME=1
  - 2. Try plotting the inverse: pick either model and plot the effect on CRS_DEP_TIME=0
- Deliverable
  - The new plots

DEMO

VISUALIZING PERFORMANCE AGAINST BASELINE

VISUALIZING PERFORMANCE AGAINST BASELINE

- Another approach of visualization is the effect of your model against a baseline, or -- even better -- against previous models.

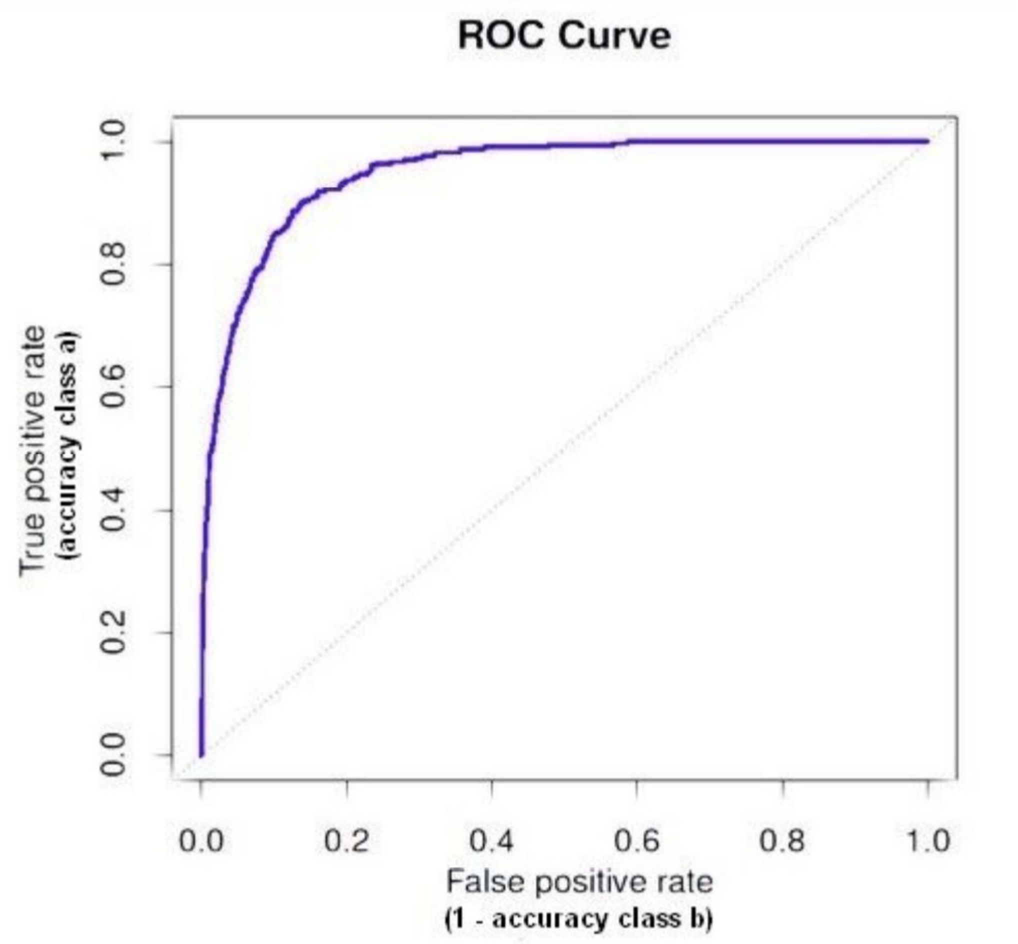
- Plots like this will also be useful when talking to your peers - other data scientists or analysts who are familiar with your project and interested in the progress you’ve made.

VISUALIZING PERFORMANCE AGAINST BASELINE

- For classification, we've practiced plotting AUC and precision-recall plots. Consider the premise of each:

  - AUC plots explain and represent "accuracy" as having the largest area under the curve. Good models will be high and to the left.

  - For precision-recall plots, it will depend on the cost requirements. Either a model will have good recall at the cost of precision or vice versa.

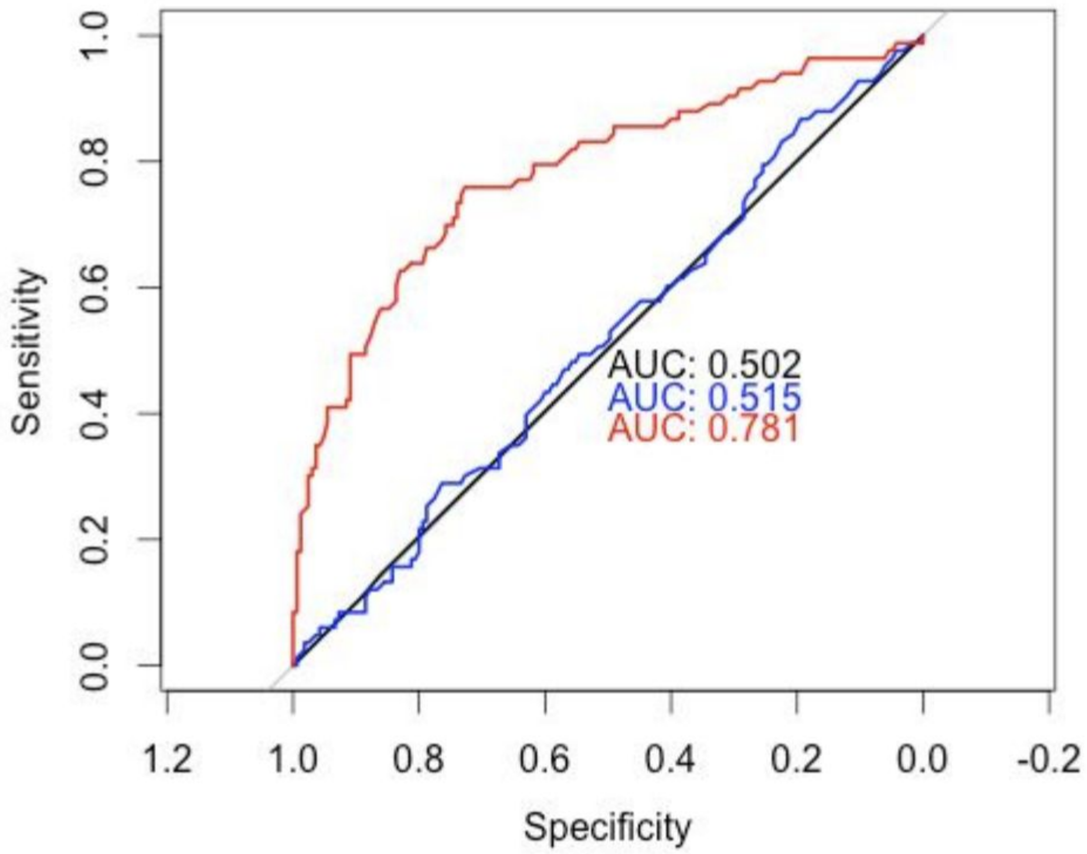


VISUALIZING PERFORMANCE AGAINST BASELINE

- When comparing multiple models:

- For AUC plots, you'll be interested in which model has the largest area under the curve.

- For precision-recall plots, based on the cost requirement, you are looking at which model has the best precision given the same recall, or the best recall given the same precision.



VISUALIZING PERFORMANCE OVER BASELINE

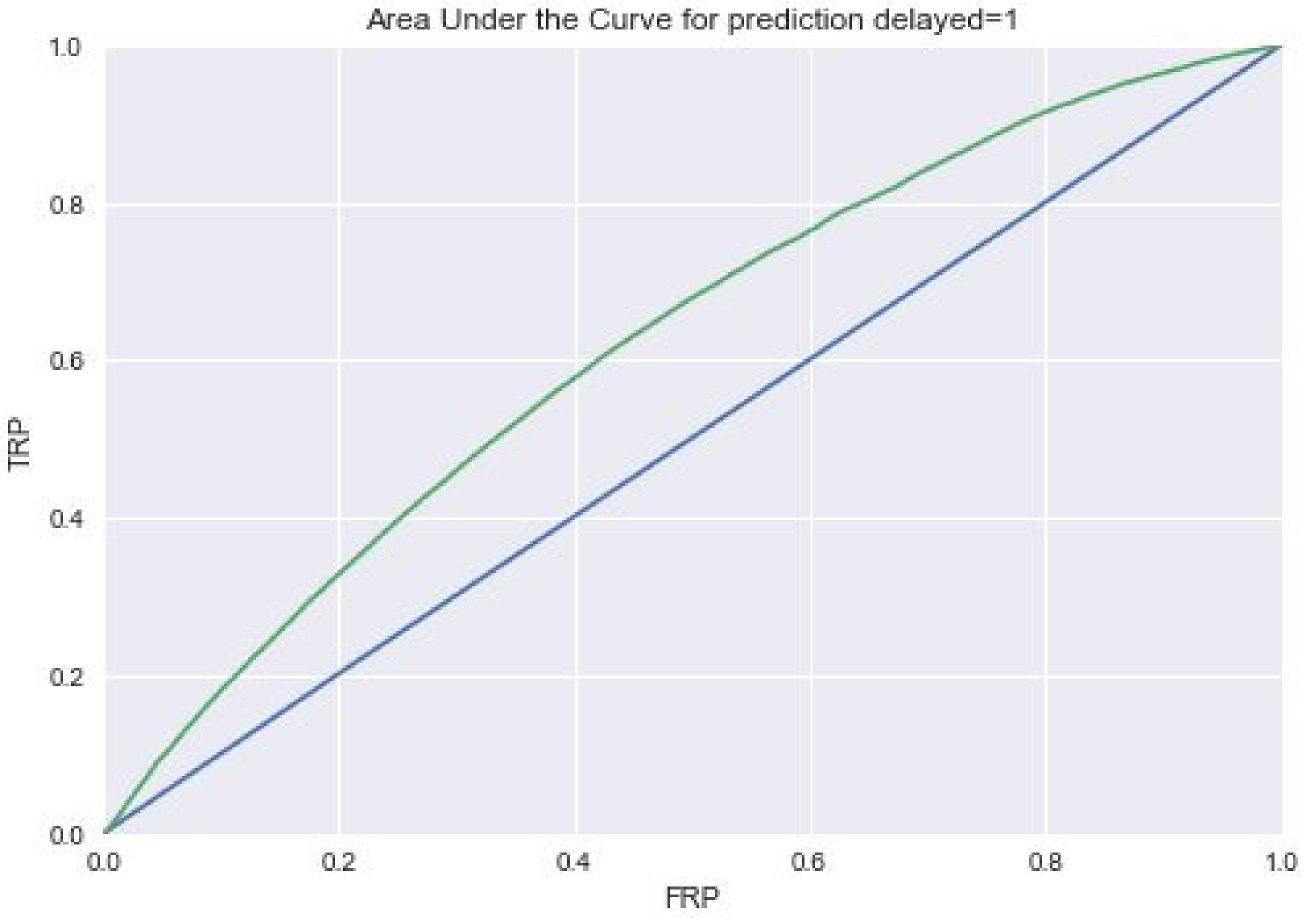
```
from sklearn.dummy import DummyClassifier
from sklearn import metrics

dummy = DummyClassifier()
dummy.fit(X, y)

df['probability_dummy'] = dummy.predict_proba(X)[:, 1]
```

VISUALIZING PERFORMANCE OVER BASELINE

```python
ax = plt.gca()

vals = metrics.roc_curve(y, df['probability_dummy'])
ax.plot(vals[0], vals[1])
vals = metrics.roc_curve(y, df['probability'])
ax.plot(vals[0], vals[1])

ax.set(
    title='Area Under the Curve for prediction delayed=1',
    ylabel='TRP',
    xlabel='FRP',
    xlim=(0, 1),
    ylim=(0, 1)
)
```

VISUALIZING PERFORMANCE AGAINST BASELINE

- This plot showcases:
  - The model using data outperforms a baseline dummy model.
  - By adding other features, there’s some give and take with probability as the model gets more complicated.



ACTIVITY -- TRY IT OUT

- In a similar approach, use the sklearn precision_recall_curve function to enable you to plot the precision-recall curve of the four models from above.

    - Keep in mind precision in the first array is returned from the function, but the plot shows it as the y-axis

- 2. Explain what is occurring when the recall is below 0.2.

- 3. Based on this performance, is there a clear winner at different thresholds?

- 4. **Bonus:** Redo both the AUC and precision-recall curves using models that have been cross validated using kfold. How do these new figures change your expectations for performance?

INDEPENDENT PRACTICE

PROJECT PRACTICE

ACTIVITY -- TRY IT OUT

- Using models built from the flight data problem earlier in class, work through the same problems. Your data and models should already be accessible.

- Your goals:

  - There are many ways to manipulate this data set. Perhaps mix both carrier and day of week in your model. Consider what is a proper "categorical" variable, and keep only what is significant. Aim to have a visual that clearly explain the relationship of variables you've used against the predicted flight delay.

  - Generate the AUC or precision-recall curve (based on which you think makes more sense), and have a statement that defines, compared to a baseline, how your model performs and any caveats. For example: "My model on average performs at x rate, but the features under-perform and explain less of the data at these thresholds."

TOPIC REVIEW

TOPIC REVIEW

- What do precision and recall mean? How are they similar and different to True Positive Rate and False Positive Rate?

- How does cost benefit analysis play a role in building models?

- What are at least two very important details to consider when creating visuals for a project's stakeholders?

- Why would an AUC plot work well for a data science audience but not for a business audience? What would be a more effective visualization for that group?