Advanced Metrics and Communicating Results

Ivan Corneillet

Data Scientist

Learning Objectives

After this lesson, you should be able to:

- Evaluate a model using advanced metrics such as confusion matrix and ROC/AUC curves
- 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 and convincing report and how they relate to specific audiences/stakeholders

Announcements and Exit Tickets

Q & A

Guest Speaker

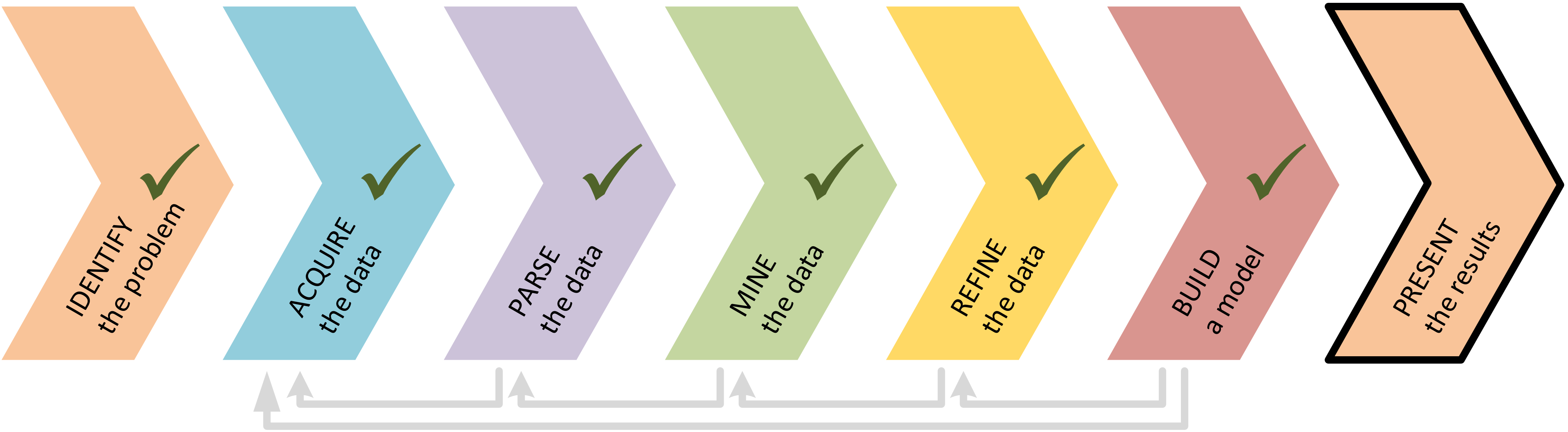
Devesh Khandelwal, General Assembly Data Science Alumnus

Today

Today, we are wrapping Unit 2 – Foundation of Modeling

Research Design and Data Analysis	Research Design	Data Visualization in *pandas*	Statistics	Exploratory Data Analysis in *pandas*
Foundations of Modeling	Linear Regression *(sessions 6, 7, and 10)*	Classification Models (k-NN, Logistic Regression) *(sessions 8, 9, and 10)*	Evaluating Model Fit *(sessions 5, 6, and 7)*	Presenting Insights from Data Models *(session 11)*
Data Science in the Real World	Decision Trees and Random Forests	Time Series Data	Natural Language Processing	Databases

… as well as the first full pass of the Data Science Workflow



Here’s what’s happening today:

- Announcements and Exit Tickets
- Guest Speaker
- ❻ Build a Model | Advanced metrics
  - Confusion Matrix
  - True Positive and False Positive Rates, ROC, and AUC
  - Plotting the ROC/AUC
  - Codealong for ROC/AUC
- ❼ Present the Results | Communicating Results
  - Showing our Work
  - Codealong to pretty up graphs
- Review
- Tickets

Pre-Work

Pre-Work

Before this lesson, you should already be able to:

- Explain the concepts of cross-validation, logistic regression, and overfitting
- Know how to build and evaluate some classification models in *sklearn* using cross-validation

❻ Build a Model

Advanced Metrics

Advanced Metrics

- Accuracy is only one of several metrics used when solving for a classification problem
  - E.g., if we know a prediction is 75% accurate, accuracy doesn't provide any insight into why the 25% was wrong. Was it wrong *equally* across all class labels? Did it just guess one class label for all predictions and 25% of the data was just the other label?
- It's important to look at other metrics to fully understand the problem

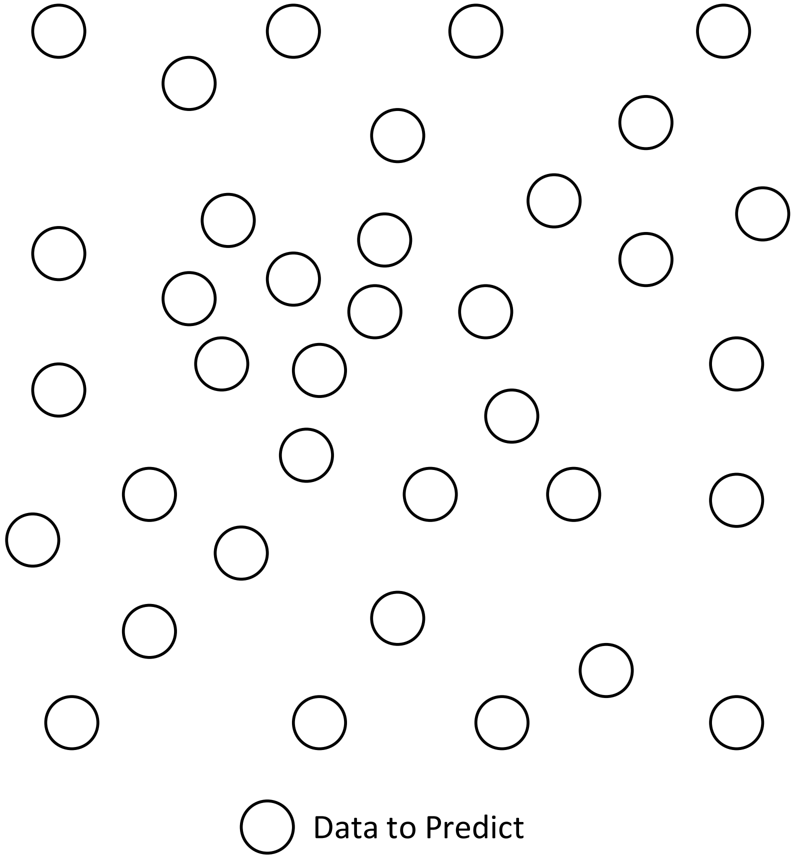
- Accuracy
  - How many observations that we predicted were correct? This is a value we'd want to increase (like R^2)

- Misclassification rate
  - Directly opposite of accuracy
  - Of all the observations we predicted, how many were incorrect? This is a value we'd want to decrease (like the mean squared error)

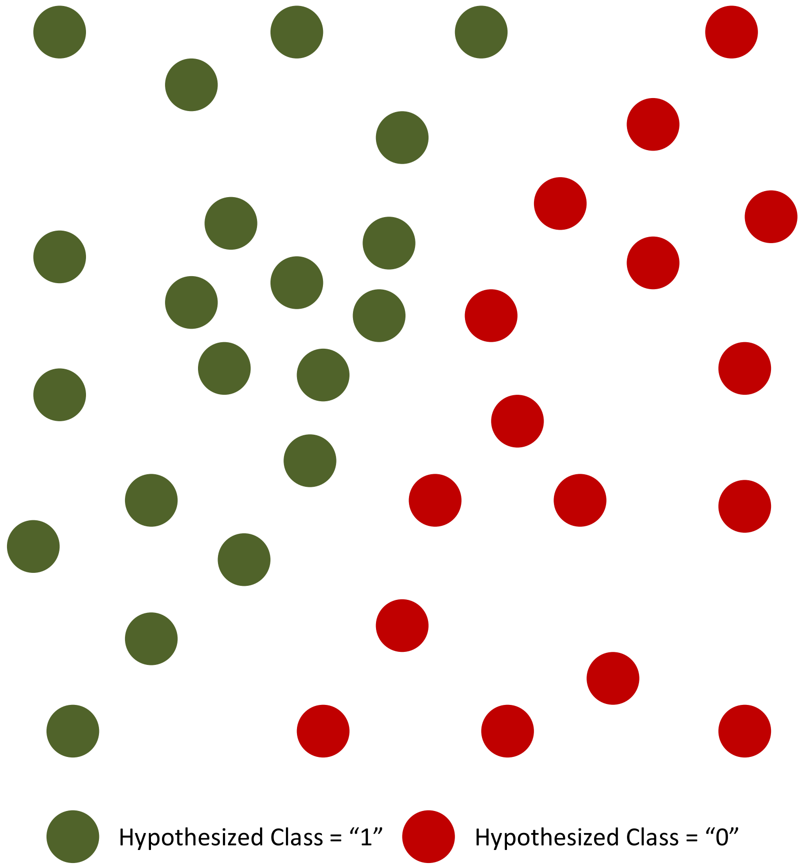
❻ Build a Model

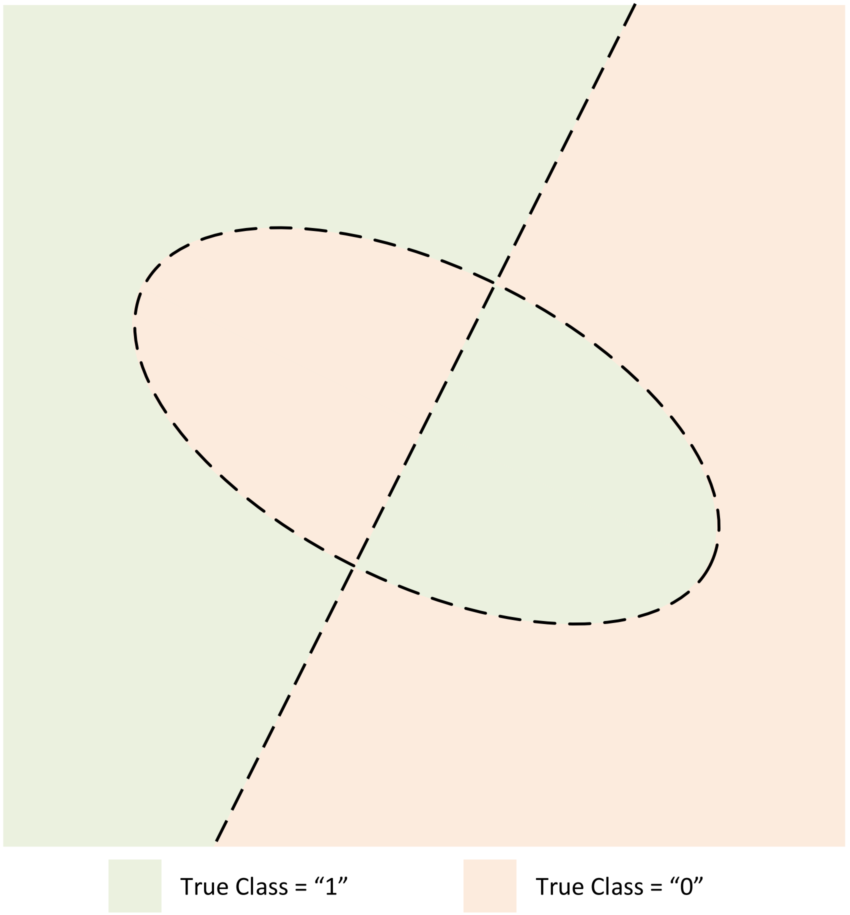
Confusion Matrix

Stepping back | Let's say we want to classify this data:

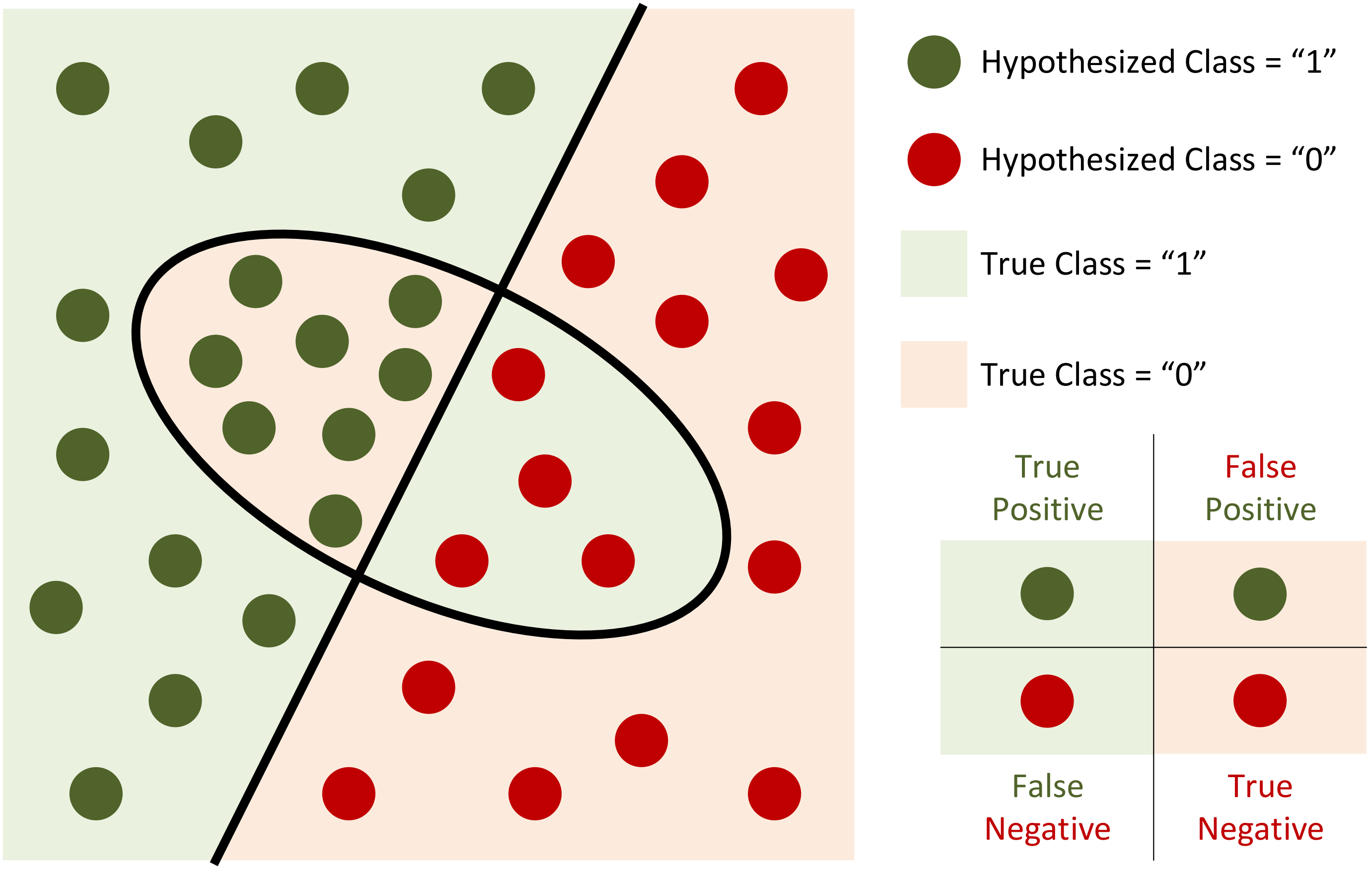


Hypothesized and true classes don’t necessarily match





We can rearrange these 4 possibilities into a 2x2 table



Confusion Matrix (a.k.a., Contingency Table or Error Matrix)

Hypothesized Class	True Class: 1	True Class: 0
1	**True Positives** (TP)	**False Positives** (FP) *(type I error)*
0	**False Negatives** (FN) *(type II error)*	**True Negatives** (TN)
Total Columns	$P = TP + FN$	$N = FP + TN$

- A confusion matrix is a specific table layout that allows visualization of the performance of a supervised learning algorithm
- Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class
- The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e., commonly mislabeling one as another)

❻ Build a Model

Codealong – Part A

Confusion Matrix

❻ Build a Model

Activity | Interpreting the confusion matrix

Activity | Interpreting the confusion matrix

EXERCISE

DIRECTIONS (10 minutes)

1. Use the variables defined in the confusion matrix (TP, FN, FP, TN, P, and N) to calculate the answers to the following questions:
    a. Overall, how often is the classifier correct?
    b. When the classifier predicts yes, how often is it correct?
    c. How often does the yes condition actually occur in our sample?
    d. When it's actually yes, how often does the classifier predict yes?
    e. When it's actually no, how often does the classifier predict yes?
    f. When it's actually no, how often does it predict no?
    g. Overall, how often is the classifier wrong?

Activity | Interpreting the confusion matrix (cont.)

EXERCISE

DIRECTIONS (cont.)

2. Given a medical exam that tests for cancer ($1 = Cancer$, $0 = Cancer\ free$), use the variables defined in the confusion matrix (TP, FN, FP, TN, P, and N) to calculate the answers to the following questions:
   a. How often is it correct when it identify patients with cancer?
   b. How often does it correctly identify patients without cancer?
   c. How often does it trigger a "false alarm" by saying a patient has cancer when they actually don't?
   d. How often does it correctly identify patients with cancer?
3. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions

Activity | Interpreting the confusion matrix (cont.)

Activity | Interpreting the confusion matrix (cont.)

Hypothesized Class \ **True Class**	**1**	**0**
1 ●	● **True Positives** (TP)	● **False Positives** (FP) *(type I error)*
0 ■	■ **False Negatives** (FN) *(type II error)*	■ **True Negatives** (TN)
Total Columns	$P = TP + FN$	$N = FP + TN$

Question: Overall, how often is the classifier correct? Answer: $\frac{TP+TN}{P+N}$ *(accuracy)*	*When the classifier predicts yes, how often is it correct?* Answer: $\frac{TP}{TP+FP}$ *(precision)*
How often does the yes condition actually occur in our sample? Answer: $\frac{P}{P+N}$ *(prevalence)*	*When it's actually yes, how often does the classifier predict yes?* Answer: $\frac{TP}{P}$ *(TPR, sensitivity, recall)*
When it's actually no, how often does the classifier predict yes? Answer: $\frac{FP}{N}$ *(FPR, fall-out)*	*When it's actually no, how often does it predict no?* Answer: $\frac{TN}{N}$ *(specificity)*
Overall, how often is the classifier wrong? Answer: $\frac{FP+FN}{P+N}$ *(misclassification rate)*	

Activity | Interpreting the confusion matrix (cont.)

		True Class	
		Has Cancer	Doesn't have cancer
Hypothesized Class	Predict Cancer ●	● True Positives (TP)	● False Positives (FP) *(type I error)*
	Predict No Cancer ■	■ False Negatives (FN) *(type II error)*	■ True Negatives (TN)
	Total Columns	$P = TP + FN$	$N = FP + TN$

How often is it correct when it identify patients with cancer? Answer: $\frac{TP}{TP+FP}$ *(precision)*	*How often does it correctly identify patients without cancer?* Answer: $\frac{TN}{N}$ *(specificity)*
How often does it trigger a "false alarm" by saying a patient has cancer when they actually don't? Answer: $\frac{FP}{N}$ *(FPR, fall-out)*	*How often does it correctly identify patients with cancer?* Answer: $\frac{TP}{P}$ *(TPR, sensitivity, recall)*
How many patients have cancer? Answer: $\frac{P}{P+N}$ *(prevalence)*	

❻ Build a Model

Activity | Interpreting the confusion matrix – Take 2

Activity | Interpreting the confusion matrix – Take 2

EXERCISE

DIRECTIONS (5 minutes)

1. We trained a binary classifier and got the following hypothesized probabilities ($\hat{p}$) for the samples in the table.
    a. What are the hypothesized classes ($\hat{c}$)?
    b. Are the samples true/false positive/negative?
2. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions

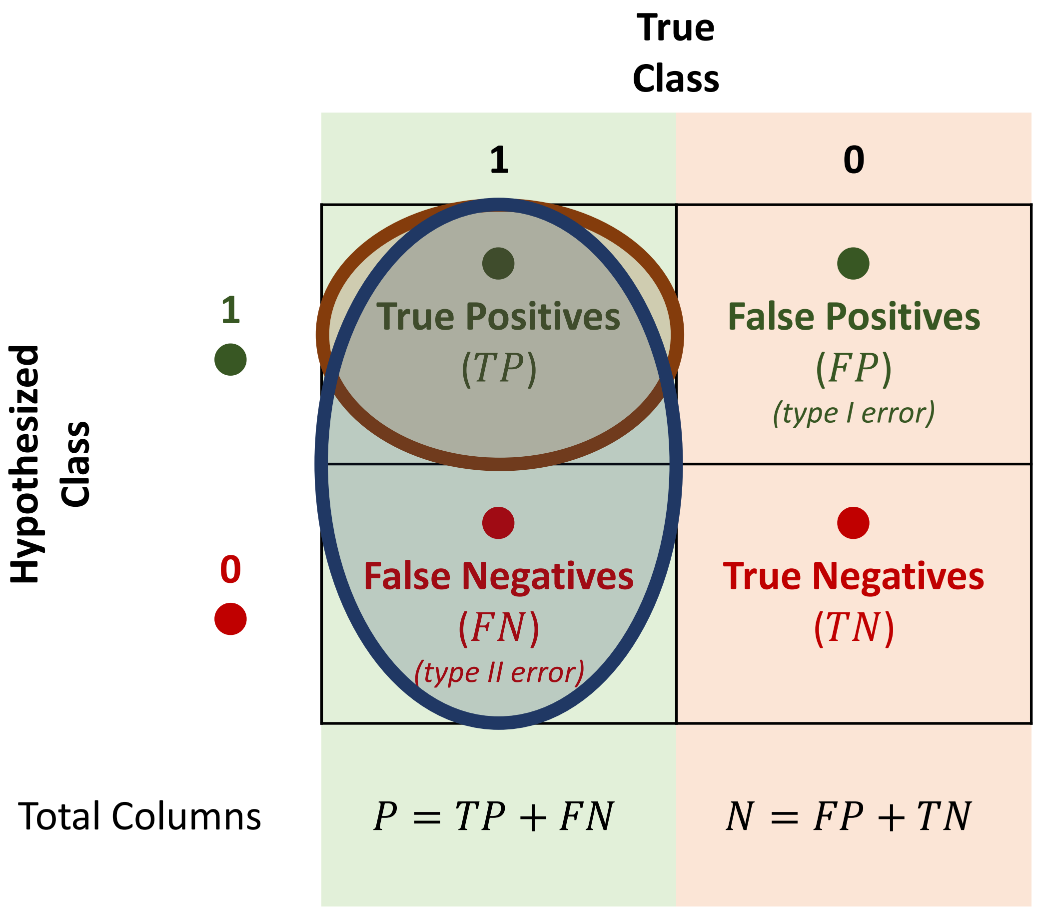
Activity | Interpreting the confusion matrix – Take 2 (cont.)

#	$\hat{p} = P(c = 1)$	$\hat{c}$	c	*True/False Positive/Negative*
1	.44	**0**	**1**	**FN**
2	.29	**0**	**0**	**TN**
3	.98	**1**	**1**	**TP**
4	.69	**1**	**0**	**FP**
5	.07	**0**	**1**	**FN**

❻ Build a Model

True and False Positive Rates, ROC, and AUC

True Positive Rate, $TPR = \frac{TP}{P}$



- When it's actually yes, how often does the classifier predict yes?
- A.k.a., "Sensitivity"

- E.g., given a medical exam that tests for cancer, how often does it correctly identify patients with cancer?
- Likewise, this can be inverted: how often does a test *correctly* identify patients without cancer

False Positive Rate, $FPR = \frac{FP}{N}$

		True Class	
		1	0
Hypothesized Class	1	True Positives (TP)	False Positives (FP) *(type I error)*
	0	False Negatives (FN) *(type II error)*	True Negatives (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

- When it's actually no, how often does the classifier predict yes?
- A.k.a., "Fall-out"

- E.g., given a medical exam that tests for cancer, how often does it trigger a "false alarm" by saying a patient has cancer when they actually don't?
- Likewise, this can be also inverted: how often does a test *incorrectly* identify patients as being cancer-free when they might actually have cancer!

True positive and false positive rates

- We can split up the accuracy of each label by using true positive and false positive rates. Using them, we can get a much clearer picture of where predictions begin to fall apart
- A good classifier would have a true positive rate approaching 1, and a false positive rate approaching 0. In a binary problem (say, predicting if someone smokes or not), it would accurately predict all of the smokers as smokers, and not accidentally predict any of the non-smokers as smokers

DS

❻ Build a Model

Activity | Introduction to the ROC space

Activity | Introduction to the ROC space

EXERCISE

DIRECTIONS (5 minutes)

1. Calculate TPR and FPR for the four confusion matrices in the handout and place them in the ROC space (TPR as a function of FPR)
2. How would you classify these four cases as a function of their performance (e.g., better or worse)
3. What does the ROC space tells you?
4. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions

Activity | Introduction to the ROC space (cont.)

EXERCISE

A

$$TPR = \frac{63}{63+37} = .63$$
$$FPR = \frac{28}{28+72} = .28$$

B

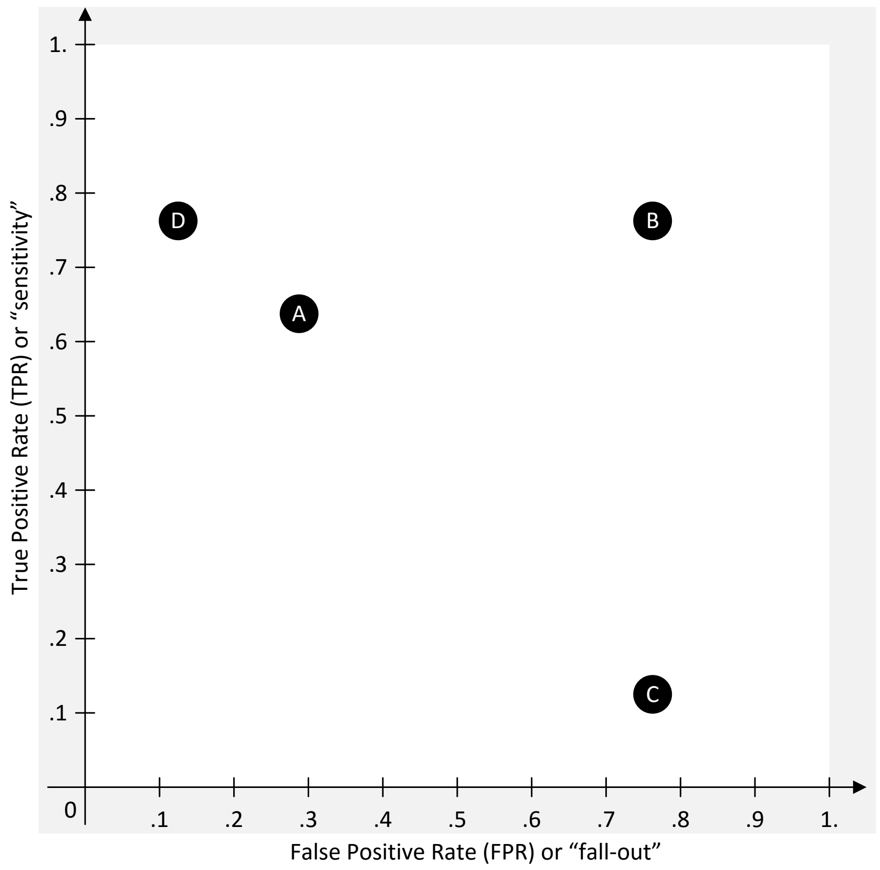
$$TPR = \frac{77}{77+23} = .77$$
$$FPR = \frac{77}{77+23} = .77$$

C

$$TPR = \frac{24}{24+76} = .24$$
$$FPR = \frac{88}{88+12} = .88$$

D

$$TPR = \frac{76}{76+24} = .76$$
$$FPR = \frac{12}{12+88} = .12$$

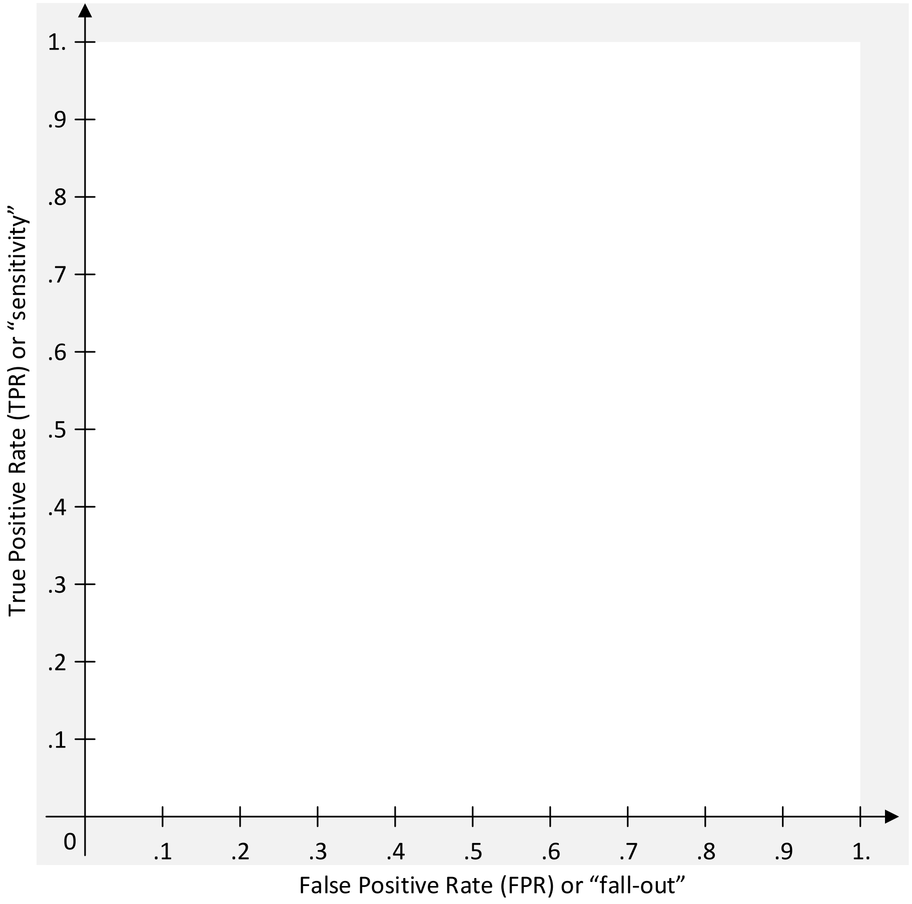


❻ Build a Model

ROC (receiver operating characteristic or relative operating characteristic) curve

ROC (receiver operating characteristic) curve (a.k.a., relative operating characteristic curve)

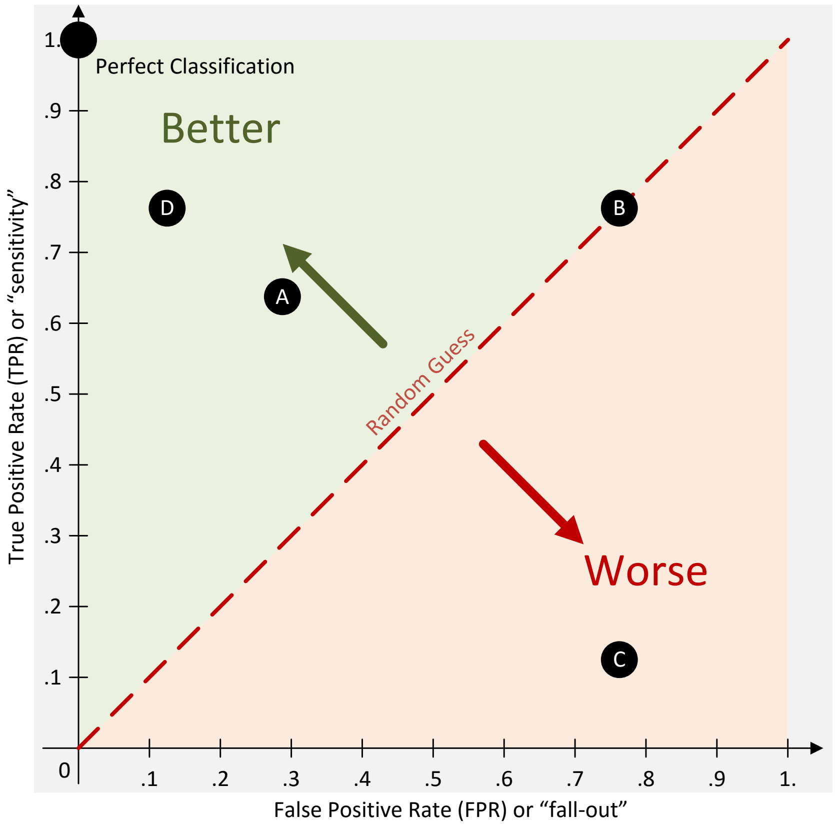
- An ROC curve plots the true positive rate (TPR) (or “sensitivity”) against the false positive rate (FPR) (or “fall-out”) at various threshold settings to illustrate the performance of a binary classifier system. The ROC curve is thus the sensitivity as a function of fall-out



The ROC space demonstrates several things:

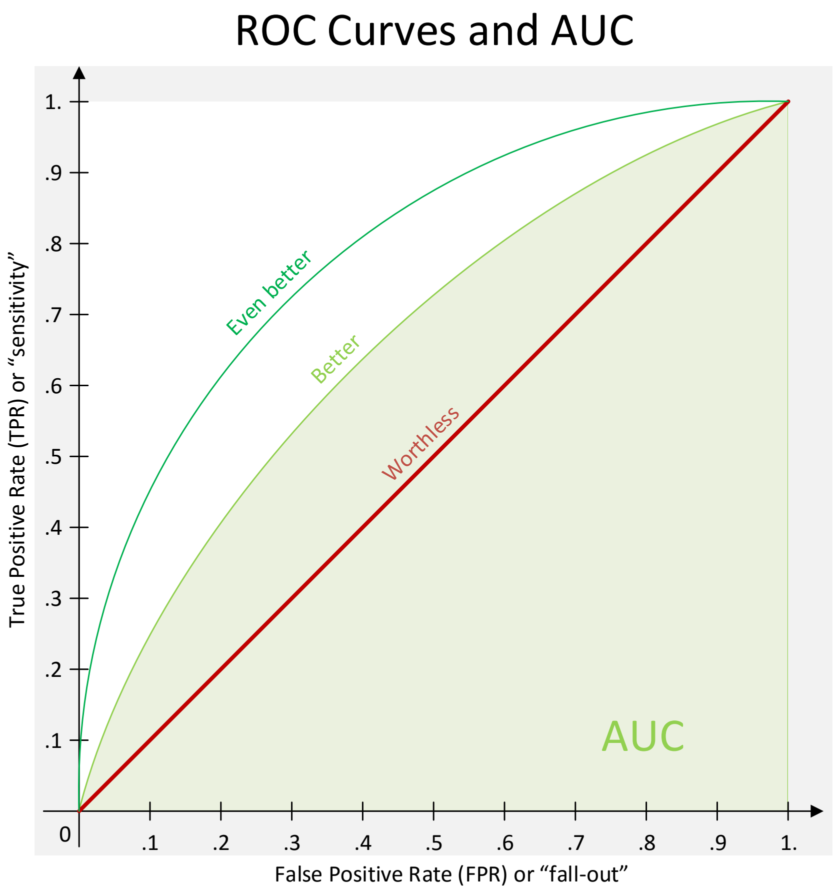
- It shows the tradeoff between sensitivity and fall-out (any increase in sensitivity will be accompanied by an increase in fallout)
  - The closer the **points** are in the left-hand border and then the top border of the ROC space, the more accurate the classifier is
  - The closer the **points** come to the 45-degree diagonal of the ROC space, the less accurate the classifier is

ROC Space



The ROC curves demonstrate several things:

- The area under the curve (AUC) is a measure of classifier accuracy
  - The closer the **curve** follows the left-hand border and then the top border of the ROC space, the more accurate the classifier is
  - The closer the **curve** comes to the 45-degree diagonal of the ROC space, the less accurate the classifier is



❻ Build a Model

Plotting an ROC curve

Plotting an ROC curve

- ❶ Discard $\hat{c}$ (hypothesized class) and whether it is a true/false positive/negative
- ❷ Order the trained sample by their decreasing hypothesized probabilities $\hat{p}$ (from more confident to have a ‘1’ down to less confident to have a ‘1’)
- ❸ Discard the original ranking from the dataset as well as $\hat{p}$
- ❹ Start at $(0, 0)$
- ❺ For each training sample in the sorted order
  - If $c = 1$, move up by $1/_{P}$
  - If $c = 0$, move up by $1/_{N}$
- ❻ If not already at $(1, 1)$, go all the way to the right, then up all the way to $(1, 1)$

Let's plot the ROC for the following trained binary classifier

EXAMPLE

#	$\hat{p}$	$\hat{c}$	c	*True/False Positive/Negative*
1	.44	0	1	FN
2	.29	0	0	TN
3	.98	1	1	TP
4	.69	1	0	FP
5	.07	0	1	FN

❶ Discard $\hat{c}$ (hypothesized class) and whether it is a true/false positive/negative

EXAMPLE

#	$\hat{p}$	c
1	.44	1
2	.29	0
3	.98	1
4	.69	0
5	.07	1

❷ Order the trained sample by their decreasing hypothesized probabilities $\hat{p}$ (from more confident to have a '1' down to less confident to have a '1')

EXAMPLE

#' (ranking by decreasing probabilities)	# (ranking from dataset)	$\hat{p}$	c
1	3	.98	1
2	4	.69	0
3	1	.44	1
4	2	.29	0
5	5	.07	1

❸ Discard the original ranking from the dataset as well as $\hat{p}$

EXAMPLE

#' (ranking by decreasing probabilities)	c
1	1
2	0
3	1
4	0
5	1

Let’s plot the ROC/AUC for the following trained binary classifier (cont.)

EXAMPLE

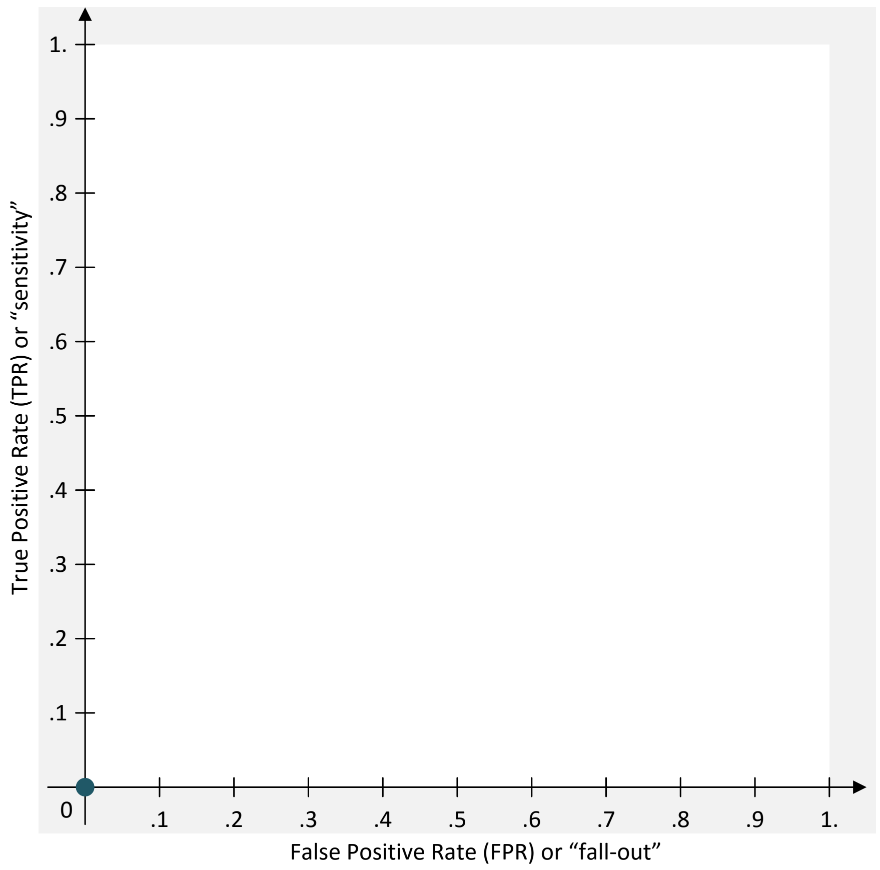
#’ (ranking by decreasing probabilities)	c
1	1
2	0
3	1
4	0
5	1

- $P = 3 \rightarrow {}^{1}/_{P} = {}^{1}/_{3}$
- $N = 2 \rightarrow {}^{1}/_{N} = {}^{1}/_{2}$

❹ Start at (0, 0)

EXAMPLE

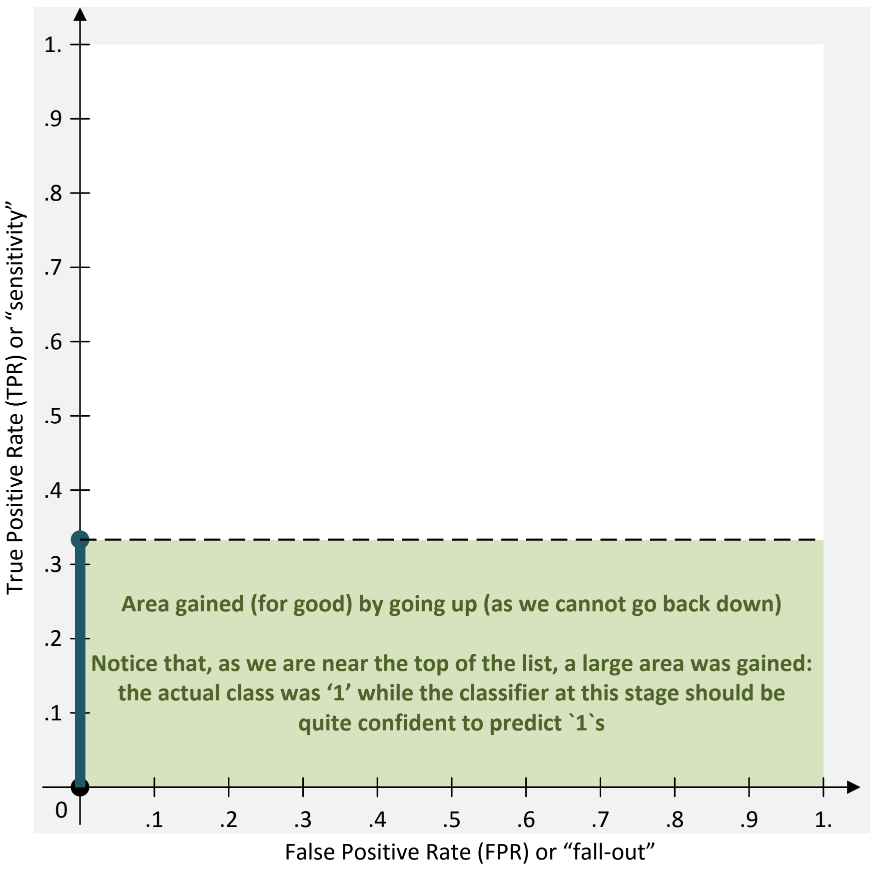
#' (ranking by decreasing probabilities)	c
1	1
2	0
3	1
4	0
5	1



❺ Because $c = 1$, move up by ${}^{1}/_{P} = {}^{1}/_{3}$

EXAMPLE

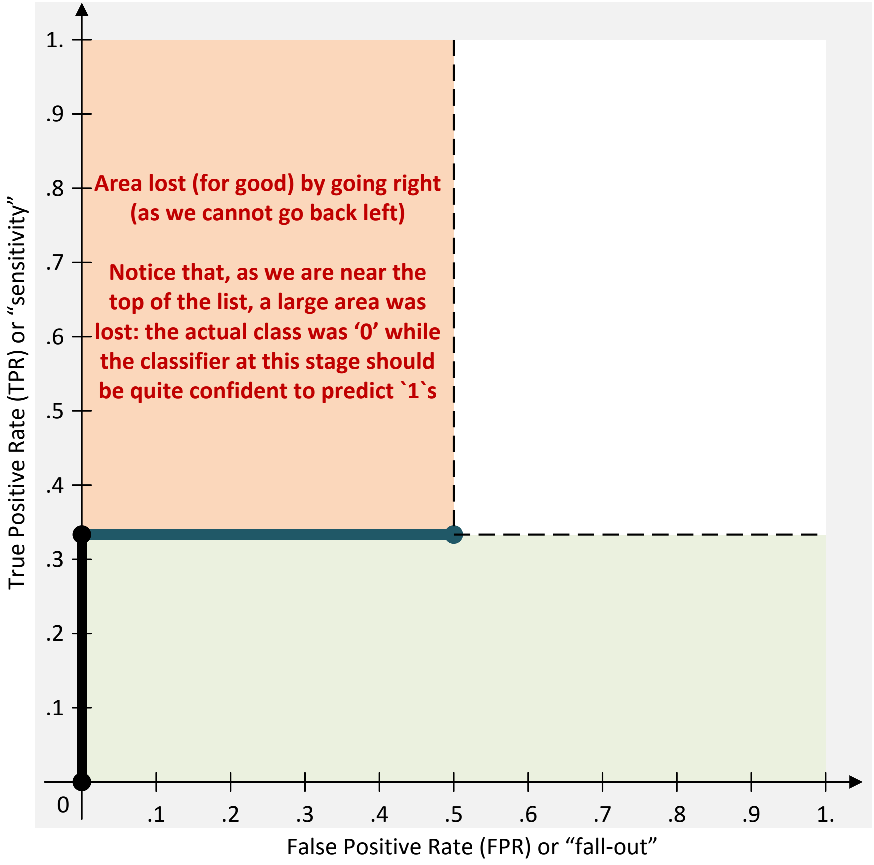
#' (ranking by decreasing probabilities)	c
1	**1**
2	0
3	1
4	0
5	1



❺ Because $c = 0$, move right by ${}^{1}/_{N} = {}^{1}/_{2}$

EXAMPLE

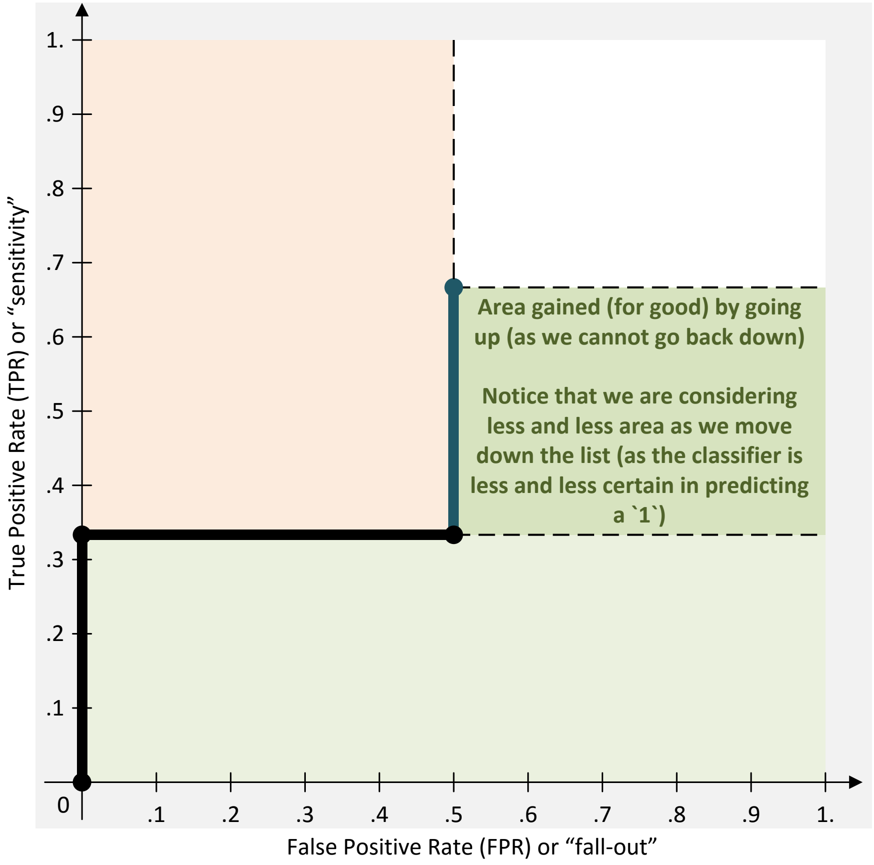
#' (ranking by decreasing probabilities)	c
1	*1*
2	**0**
3	1
4	0
5	1



❺ Because $c = 1$, move up by $^1/_P = {}^1/_3$

EXAMPLE

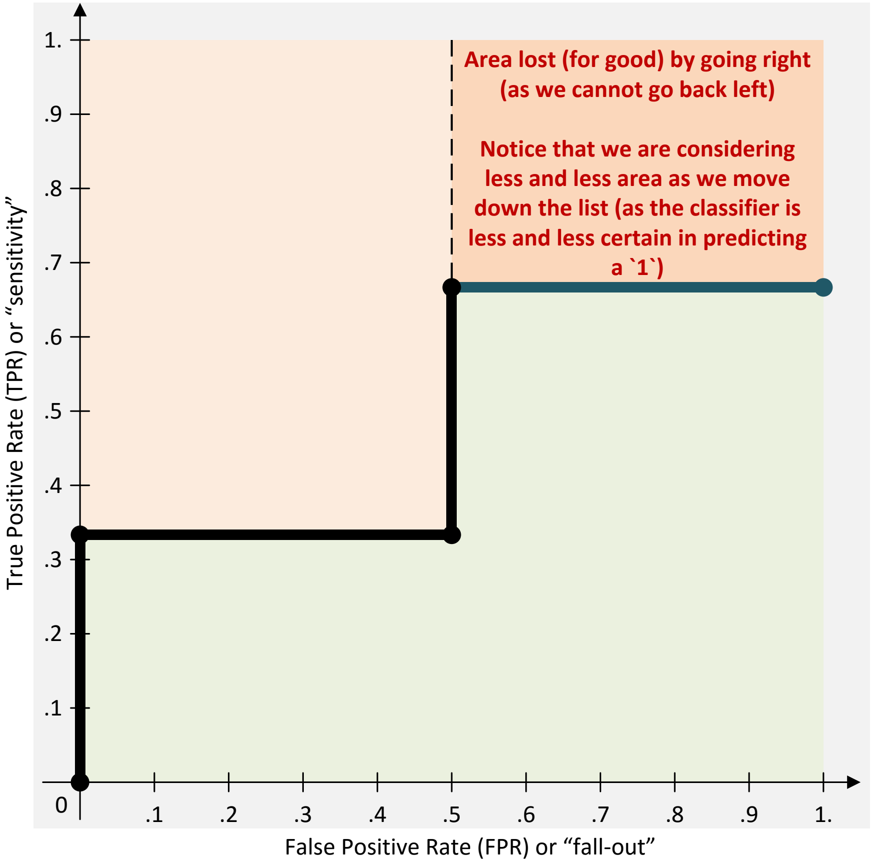
#' (ranking by decreasing probabilities)	c
1	*1*
2	*0*
3	**1**
4	0
5	1



❺ Because $c = 0$, move left by ${}^{1}/_{N} = {}^{1}/_{2}$

EXAMPLE

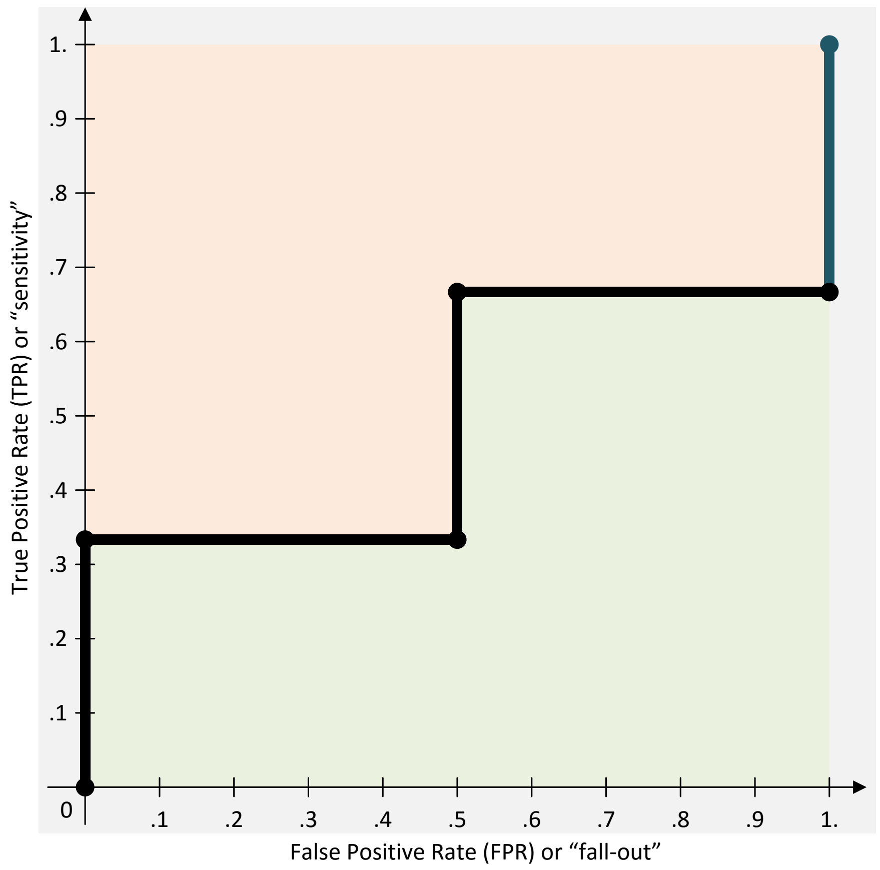
#' (ranking by decreasing probabilities)	c
1	*1*
2	*0*
3	*1*
4	**0**
5	1



❺ Because $c = 1$, move up by $^1/_P = {}^1/_3$

EXAMPLE

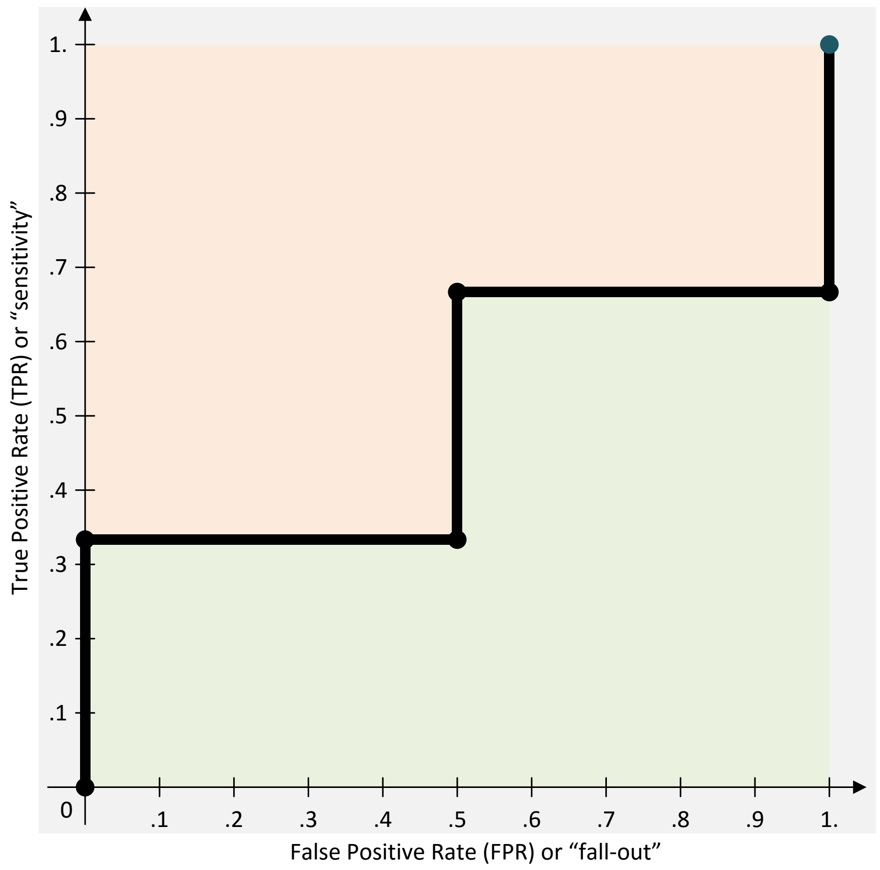
#' (ranking by decreasing probabilities)	c
1	*1*
2	*0*
3	*1*
4	*0*
5	**1**



❻ If not already at $(1, 1)$, go all the way to the right, then up all the way to $(1, 1)$

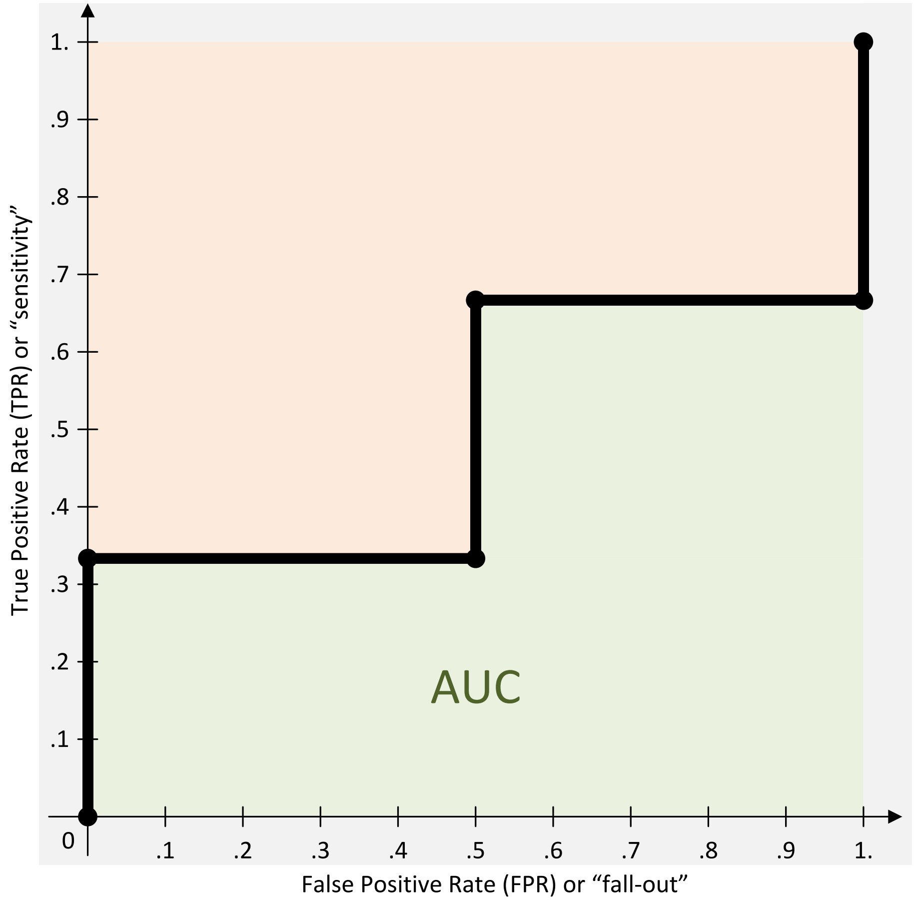
EXAMPLE

#' (ranking by decreasing probabilities)	c
1	*1*
2	*0*
3	*1*
4	*0*
5	*1*



Let's plot the ROC/AUC for the following trained binary classifier (cont.)

EXAMPLE



Plotting an ROC curve (cont.)

- Notes
  - We don't rely on a threshold (e.g., .5) for plotting ROC curves. Indeed, moving up or right is independent of $\hat{p}$ (we discarded it in step ❸) and only relies on a decreasing ranking of $\hat{p}$ and then c
  - As a matter of fact, you can use ROC curves to select the best threshold but we won't address it here

❻ Build a Model

Codealong – Part B

ROC/AUC

❼ Present the Results

Communicating Results

We built a model! Now what?

- We’ve built our model, but there is still a gap between our iPython notebook with its plots and figures and a slideshow needed to present our 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? (cont.)

- Imagine how a non-technical audience might respond to the following statements:
  - “The predictive model I built has an accuracy of 80%”
  - “Logistic regression was optimized with L2 regularization”
  - “Gender was more important than age in the predictive model because it has a ‘larger coefficient’”
  - “Here’s the AUC chart that shows how well the model did”

We built a model! Now what? (cont.)

- Who is your audience? Are they technical? What are their concerns?
  - 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? (cont.)

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

❼ Present the Results

Showing our Work

Showing our 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

To convey important information to your audience, make sure your charts are simplified, easily interpretable, and 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 against a target
- If your model used a data transformation (like natural log), just visualize the original data
- Remove 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

When building visuals for another audience, ask yourself who, what, and how

Who	What	How
Who is my target audience for the visual?	What do they already know about this project? What do they need to know?	How does my project affect this audience? How might they interpret (or misinterpret) the data?

Visualizing Models over Variables

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

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

Visualizing Performance Against Baseline

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

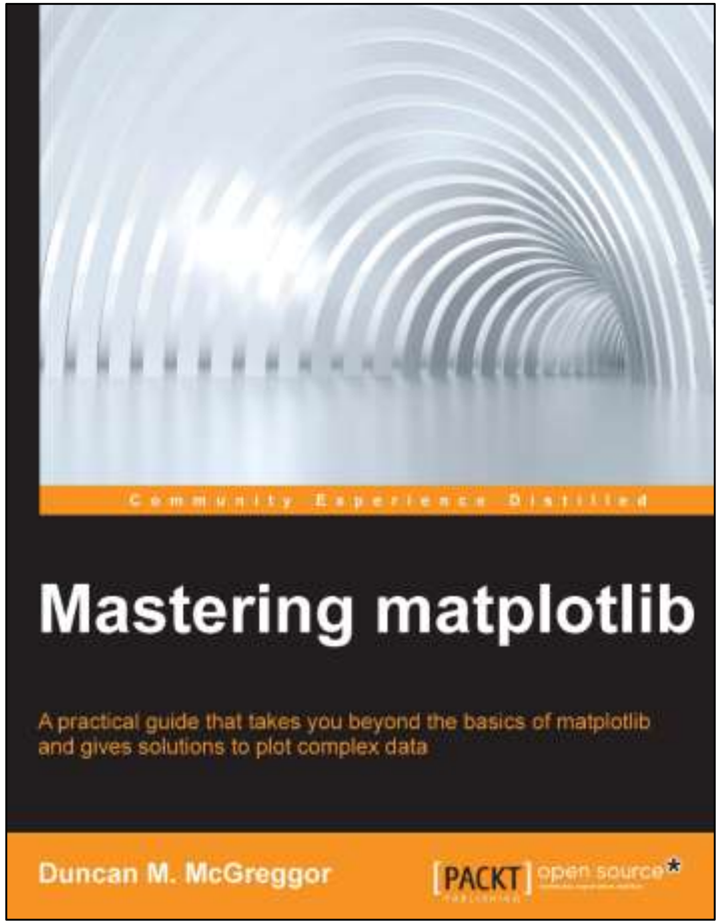
- Plots like this will also be useful when talking to your peers – other data scientists or analysts who are familiar with your project and interested in the progress you've made

❼ Present the Results

Codealong – Part C

Prettying up Graphs

A good resource to learn more about *matplotlib* (optional; not required for the course)



Review

Review

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

Review (cont.)

You should now be able to:

- Evaluate a model using advanced metrics such as confusion matrix and ROC/AUC curves
- 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

Next Class

Decision Trees and Random Forests

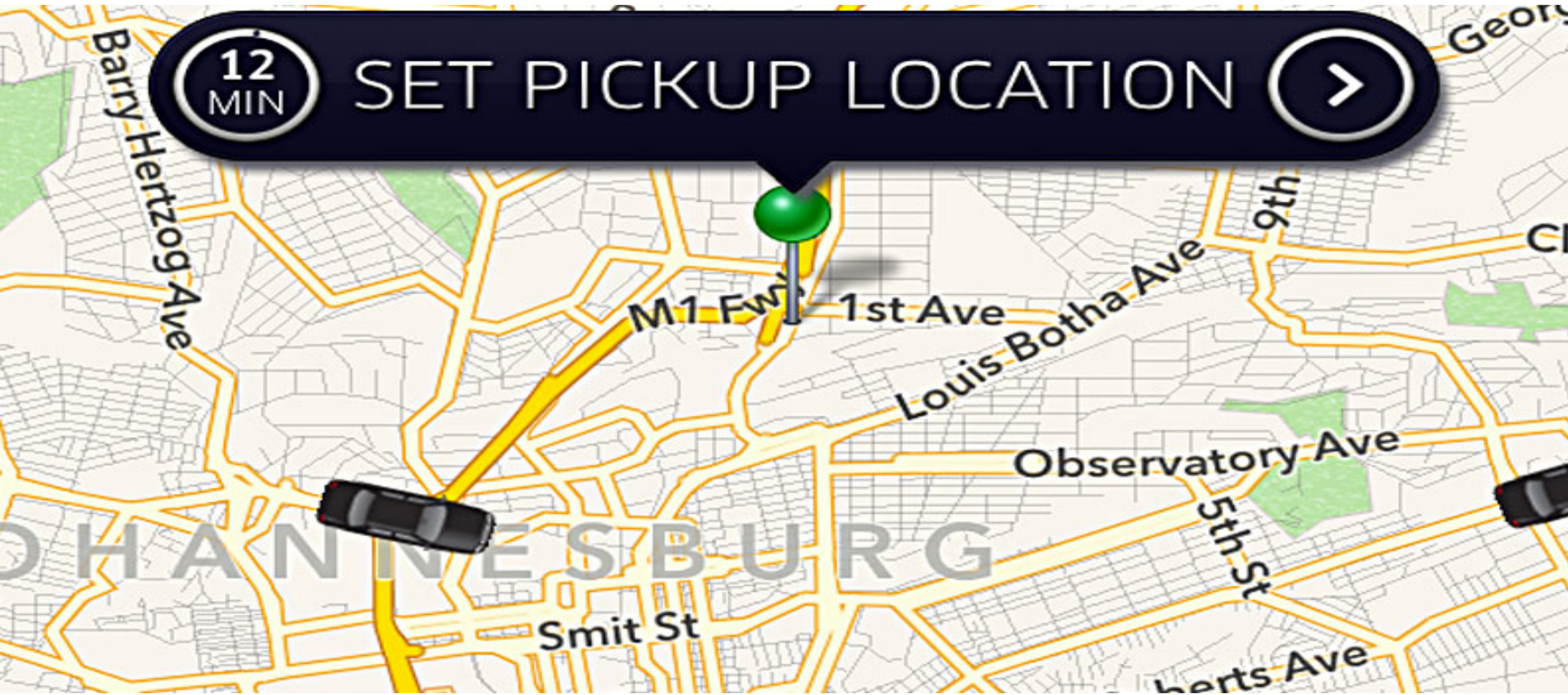
Learning Objectives

After the next lesson, you should be able to:

- Understand and build decision tree models for classification and regression
- Understand the differences between linear and non-linear models
- Understand and build random forest models for classification and regression
- Know how to extract the most important predictors in a random forest model

Exit Ticket

Don't forget to fill out your exit ticket here



Predicting Cab Booking Cancellations

by Devesh Khandelwal

Agenda

- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

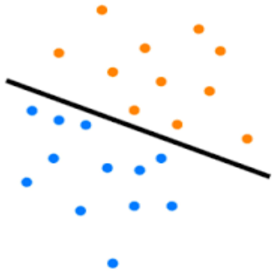
Problem Statement

Customers can cancel the booking up to the **last minute** of pick up at **no cost** to them

Cancelled booking dents the revenue of the company and adds operational overheads

Use the Data collected over time to predict the probability of booking cancellation

Problem Analysis



Classification Task – Classify the Cancellation feature into :

- ✓ '0' (Not Cancelled)

  or

- ✓ '1' (Cancelled)

Agenda

- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

Dataset

Training Data-
- ✓ 43 K records
- ✓ 18 Features

Uneven Classes
- ✓ Approx 7% of the total bookings are actually Cancelled(Training Data)

Source:- https://inclass.kaggle.com/c/predicting-cab-booking-cancellations/data

Features at a Glance

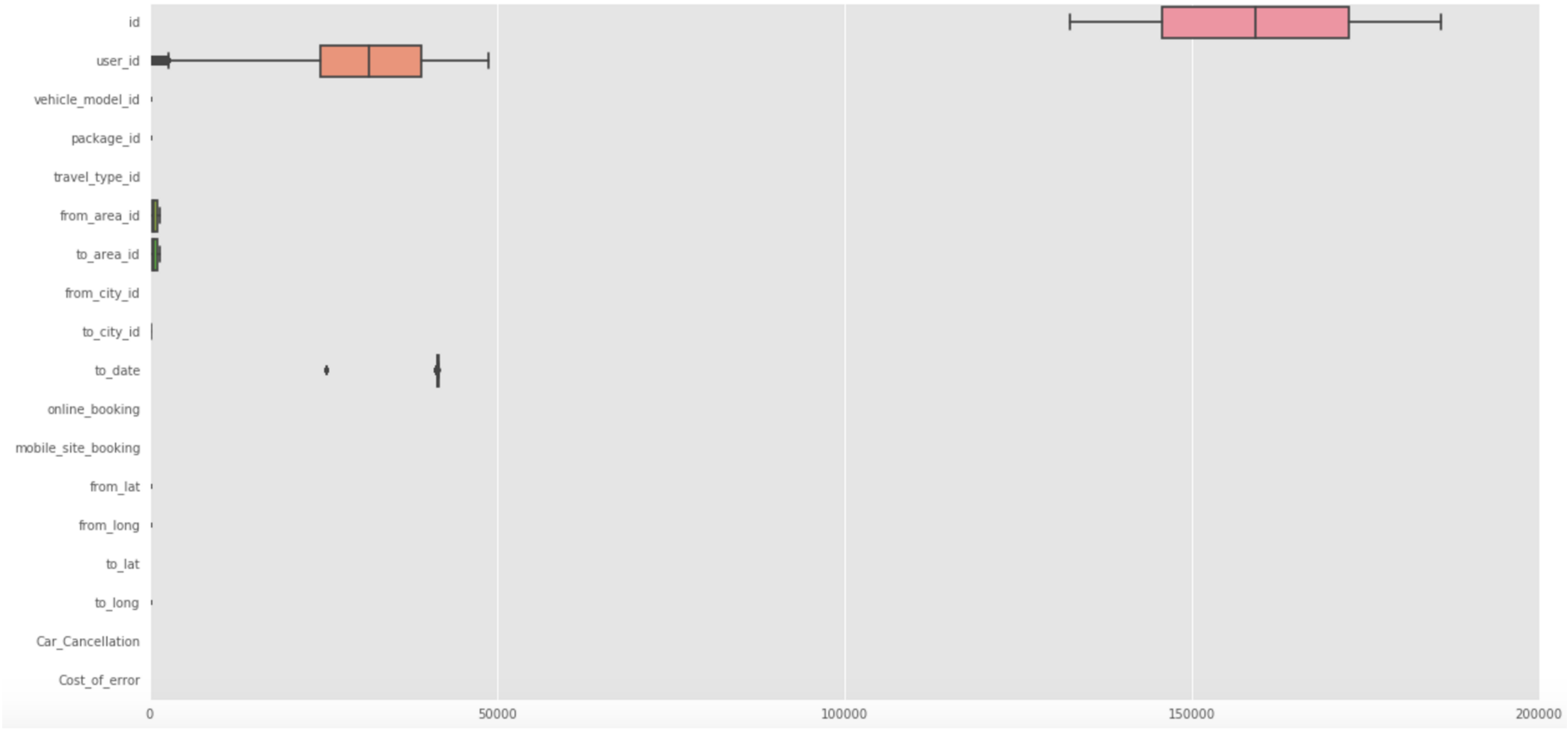
Features set includes:

✓ Vehicle attributes

✓Booking attributes including-

- ➢ Online
- ➢ GPS data
- ➢ Mobile
- ➢ Travel Type
- ➢ Source
- ➢ Destination

Features at a Glance(Contd..)

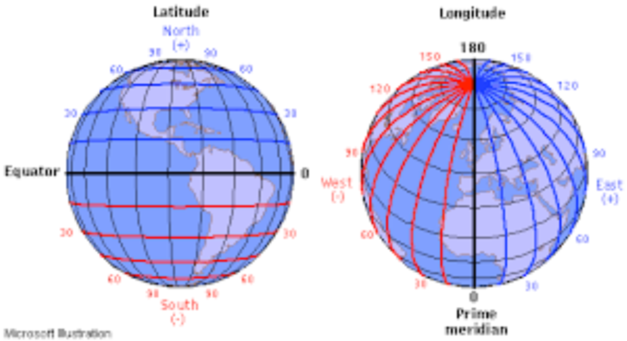


Agenda

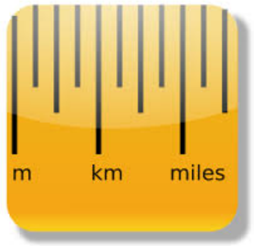
- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

Feature Engineering

(GPS Data)



Booking Coordinates (Latitude ,longitude of source & Destination)



New feature ’Distance’

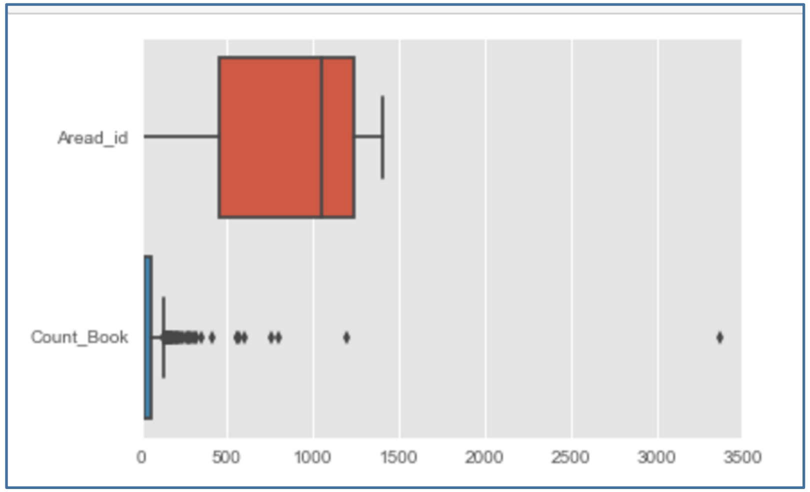
Implementation

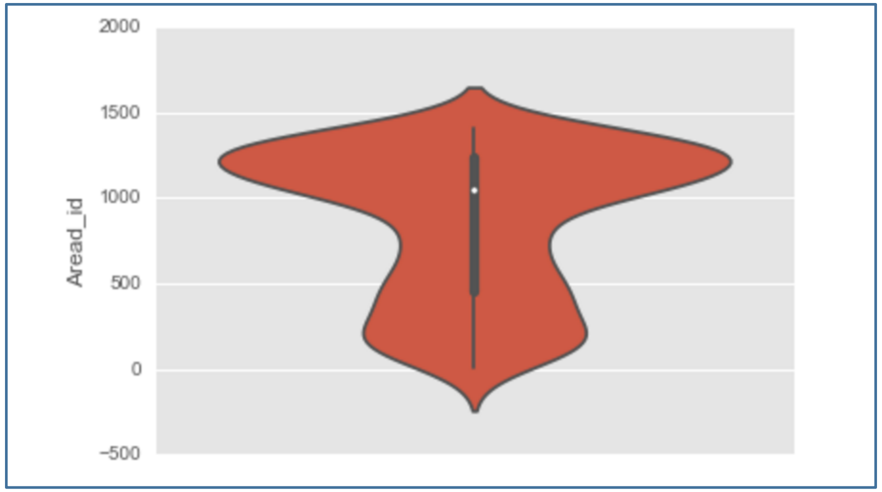
- df['distance'] = 6367 * 2 * np.arcsin(np.sqrt(np.sin(np.radians(df['to_lat']) - math.radians(37.2175900)/2)**2 + math.cos(math.radians(37.2175900)) * np.cos(np.radians(df['to_lat']) * np.sin(np.radians(df['from_long']) - math.radians(-56.7213600)/2)**2)))
- df['distance']=df.distance/1000
- df.distance = df.distance.apply(replace_null)

Feature Engineering

(Area information)

- Data set has features **from_area_id** and **to_area_id** that depicts the location of the origin and destination
- 599 unique values for feature- ‘**Area_id**’



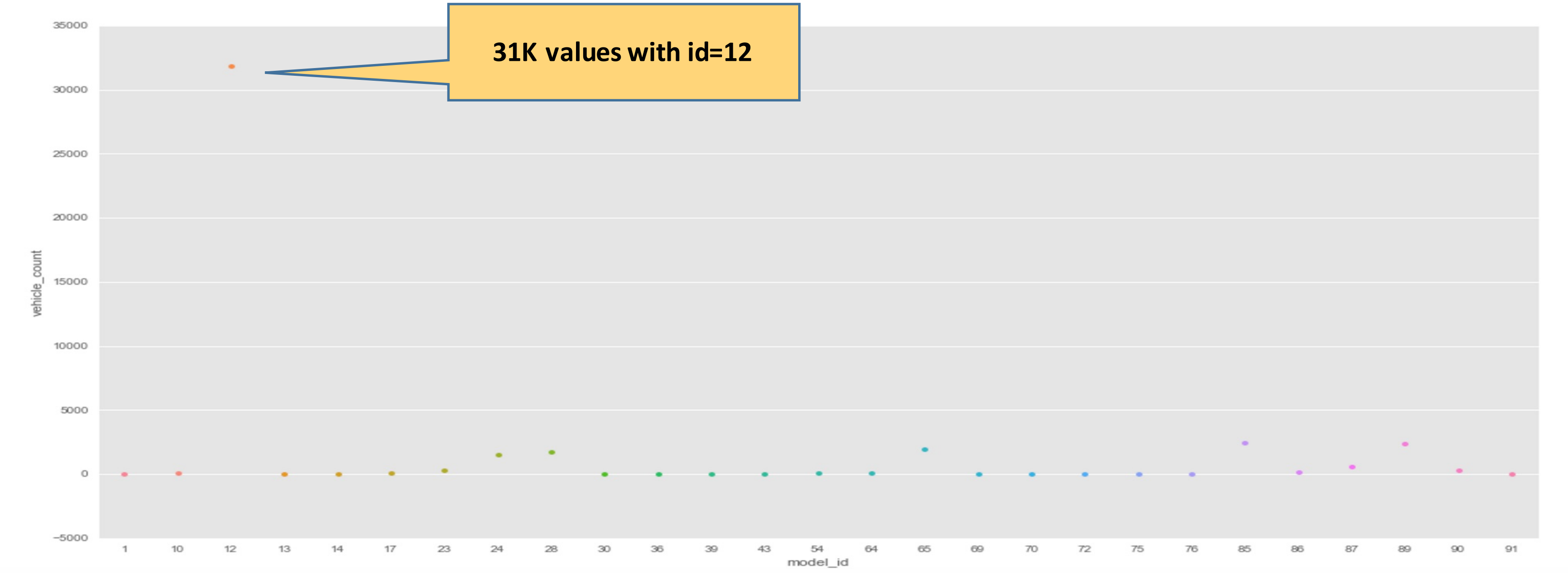


- Majority of the bookings cater to a few of the areas as is evident from the density function
- New feature ‘Popular_Pickup’=0 if area_id of the booking is not from the popular_area and 1 otherwise
- New feature ‘Popular_Drop’=0 if area_id of the booking is not from the popular_area and 1 otherwise

Feature Engineering

(Fleet Analysis)

Vehicle_Model_id- 16 unique values



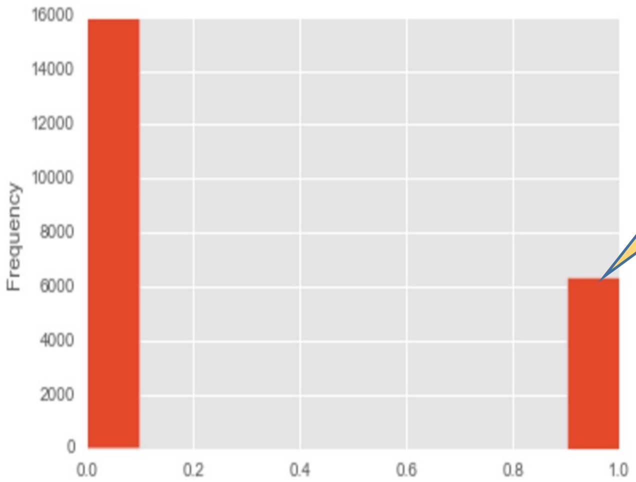
- Creating new_feature- vehicle_category

- cat_1 = vehicle_cat_df.vehicle_count.max()
- cat_2 = round(vehicle_cat_df.vehicle_count.quantile(.75))
- cat_3 = round(vehicle_cat_df.vehicle_count.quantile(.5))
- cat_4 = round(vehicle_cat_df.vehicle_count.quantile(.25))

Feature Engineering

(User segmentation)

User_id – Id of the user requesting the service



Distribution of User_id

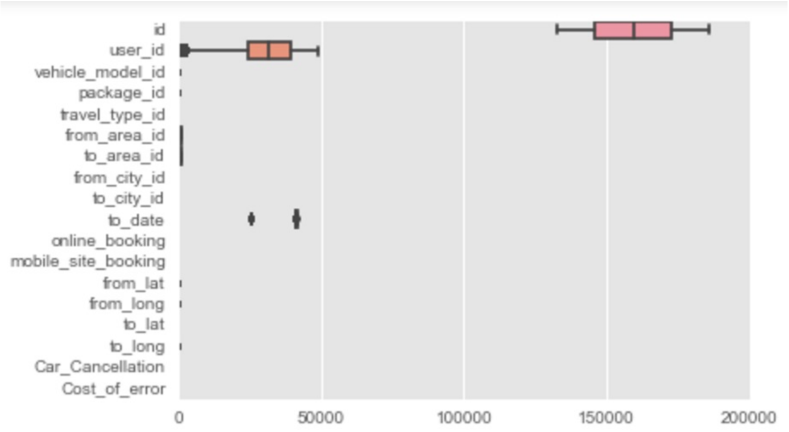
- **22K unique value**
- **6K returning users**

Transformed to

New Feature – is_frequent
- ✓ Is_frequent = 1 (returning user)
- ✓ Is_frequent = 0 (one time user)

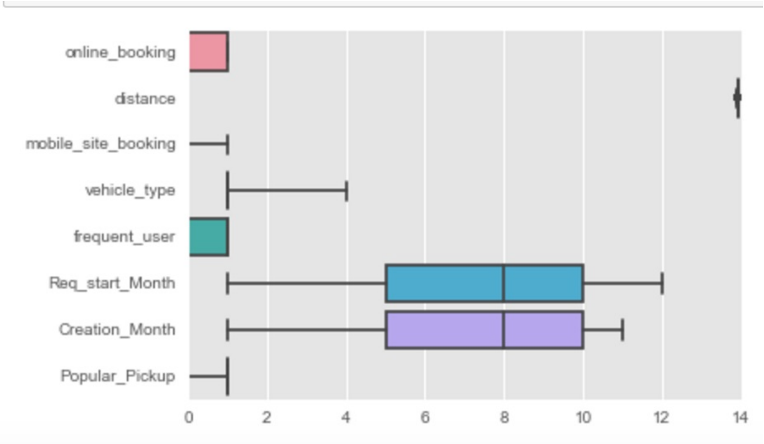
Feature Engineering

(Summary)



Raw Data set

Transformed to



Transformed Data
(final data set)

Stratified Sampling

- Uneven Data Set- less than 7% of the booking are cancelled
- Creating a balanced data set with equal distribution of dependent variable

- y_0 = df[df.Car_Cancellation == 0]
- y_1 = df[df.Car_Cancellation == 1]
- n = min([len(y_0), len(y_1)])
- y_0 = y_0.sample(n = n, random_state = 0)
- y_1 = y_1.sample(n = n, random_state = 0)df_strat = pd.concat([y_0, y_1])
- X_strat = df_strat[['online_booking','distance','mobile_site_booking','vehicle_type','frequent_user','Req_start_Month','Creation_Month','Popular_Pickup']]y_strat = df_strat.Car_Cancellation

Data Science Workflow – Parse, Mine, Refine the Data

Agenda

- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

Modelling-Stats Model

(Kitchen Sink Strategy)

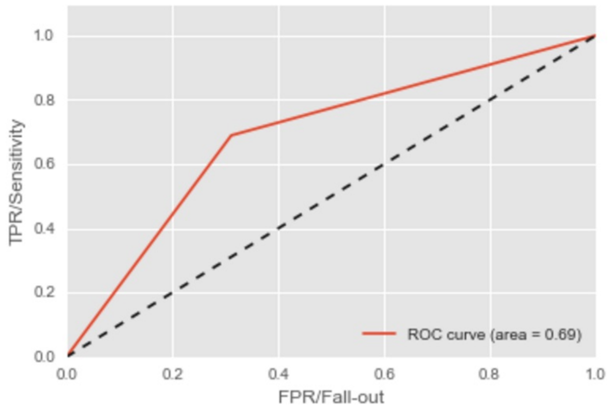
Output of Stats Model

	coef	std err	z	P>|z|	[95.0% Conf. Int.]
const	-908.5756	4799.228	-0.189	0.850	-1.03e+04 8497.738
online_booking	1.2302	0.047	26.333	0.000	1.139 1.322
distance	63.2429	2.440	25.923	0.000	58.461 68.024
mobile_site_booking	1.3237	0.080	16.562	0.000	1.167 1.480
vehicle_type	-0.8444	0.056	-15.117	0.000	-0.954 -0.735
travel_type_id	12.8902	2399.554	0.005	0.996	-4690.149 4715.929
frequent_user	-0.7271	0.043	-16.901	0.000	-0.811 -0.643
Req_start_Month	0.7830	0.077	10.134	0.000	0.632 0.934
Creation_Month	-0.5925	0.078	-7.583	0.000	-0.746 -0.439
Popular_Pickup	-0.3916	0.049	-7.946	0.000	-0.488 -0.295
Popular_Drop	-0.1377	0.048	-2.867	0.004	-0.232 -0.044

- Kitchen Sink strategy on the Data set further reduces the features
- Travel_type_id gets eliminated from further analysis due to the higher p value

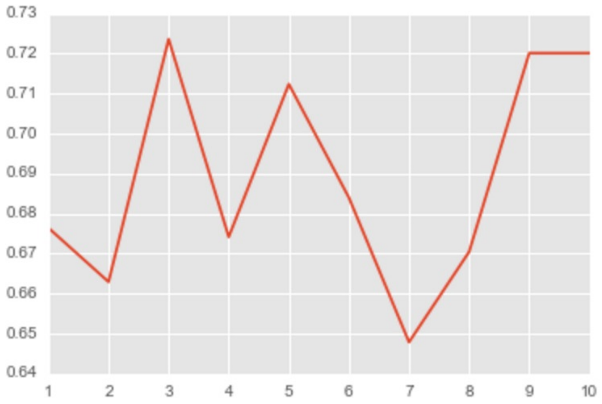
Modelling
(Logistic Regression)

Training



- 69% Accuracy on the Training Data

Cross Validation



- 69% mean Accuracy on the CV Data(10 folds)

Test Data

```
model.score(test_X_strat,test_y_strat)
```

0.69999999999999996

Modelling

(Decision Trees)

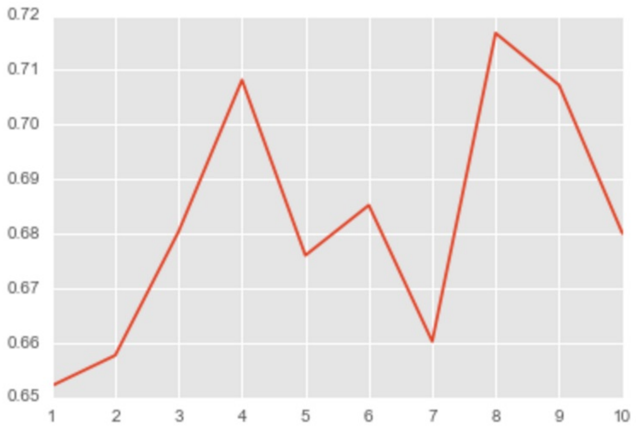
Training

```
model_tree.score(train_X_strat, train_y_strat)
```

```
0.96877189424135701
```

- 97 % Accuracy on the Training Data

Cross Validation



- 68.2% mean Accuracy on the CV Data(10 folds)

Test Data

```
model_tree.score(test_X_strat , test_y_strat)
```

```
-0.20076622358025387
```

```
(tree_y_hat == test_y_strat).mean()
```

```
0.67927927927927922
```

Modelling

(Random Forests - no of trees=10000)

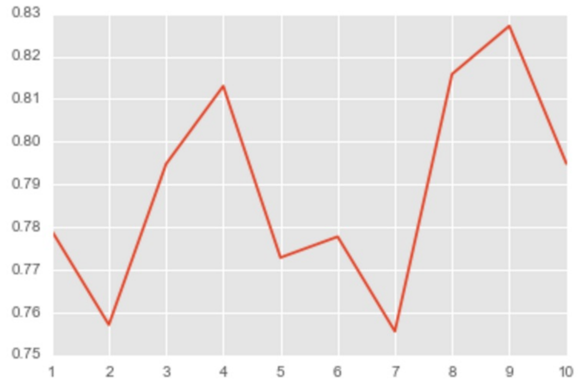
Training

```
model_forest.score(train_X_strat, train_y_strat)
```

0.9862612612612612

- 98 % Accuracy on the Training Data

Cross Validation



- 79% mean Accuracy on the CV Data(10 folds)

Test Data

```
model_forest.score(test_X_strat, test_y_strat)
```

0.7162162162162162

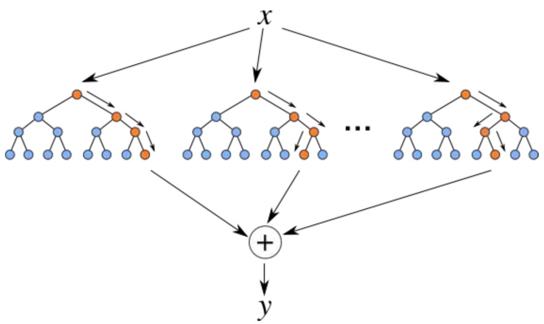
Modelling

(Random Forests-Feature Importance & Co-relation)

Feature	%age	Co-Relation with the dependent variable
distance	62.4	0.261690
Creation_Month	10.4	0.262376
Req_start_Month	9.1	0.262179
online_booking	6.2	0.255332
frequent_user	4.1	-0.158572
vehicle_type	3.3	-0.154804
mobile_site_booking	2.2	0.104083
Popular_Pickup	1.9	-0.056936
Total	96	

Modelling

Conclusion

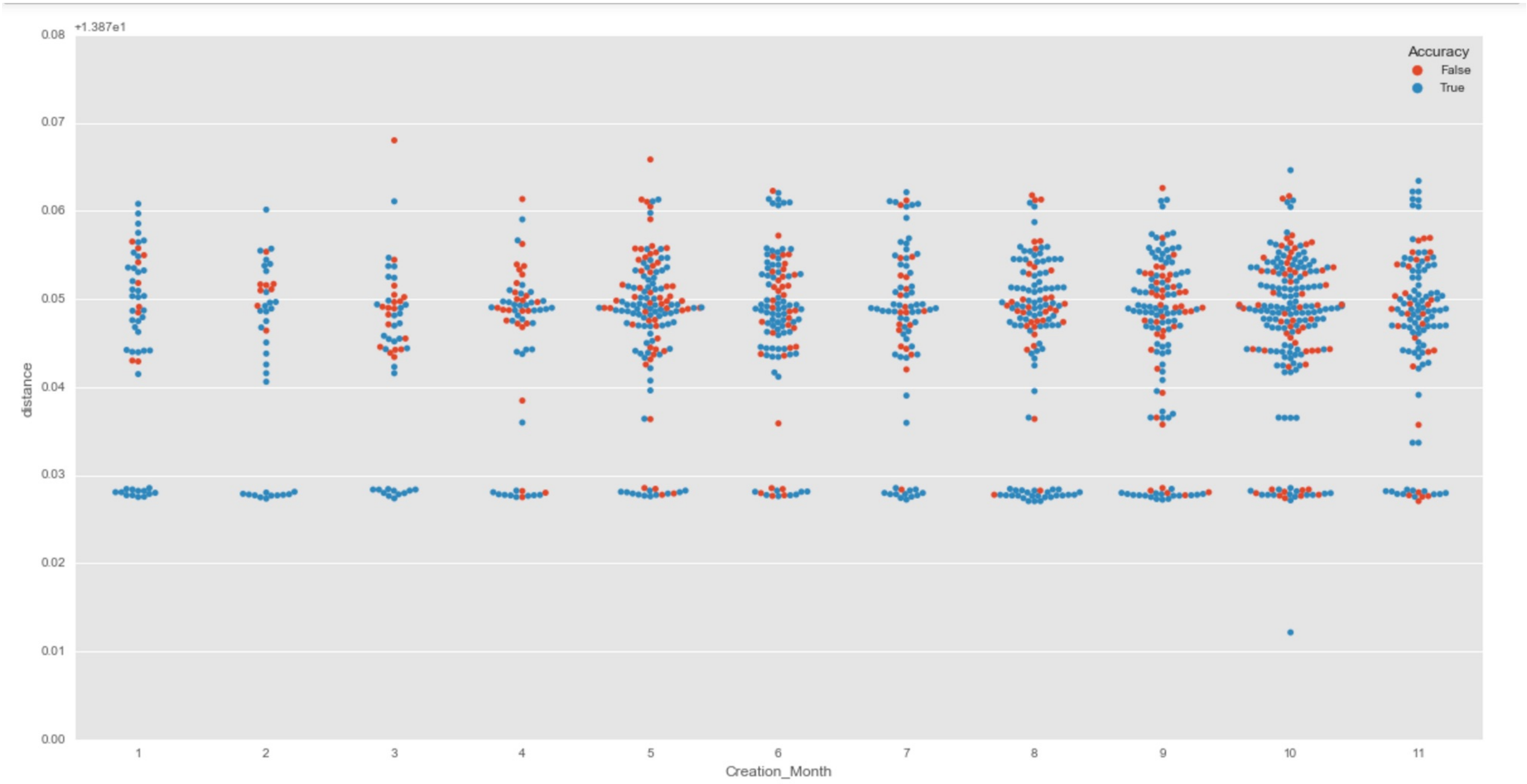


- Random forest seems to be the best amongst all the models
- Random forest also seem to cut off the nose and make the best decision on the important features
- Chance of over -fitting is less as compared to Decision trees(which is most likely to have overfit – Training score of 97%)

Agenda

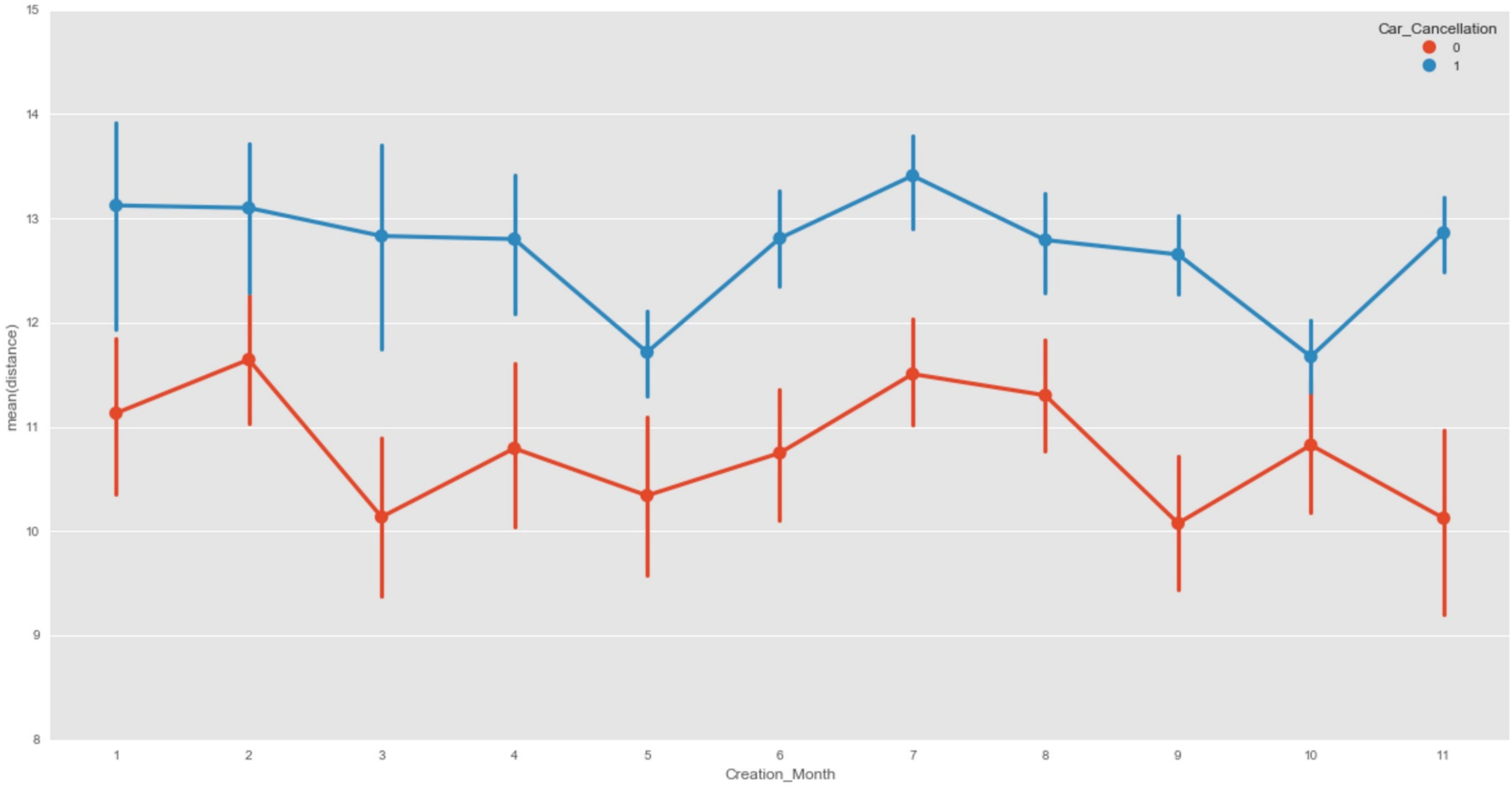
- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

Model Accuracy
(Random Forest on Test set)



- Appears that the Maximum number of misclassifications are occurring in Apr,May

Interpretation



- Appears that the chances for the cancellation is maximum in Jul when the mean travel distance is between 13 -14 KMs

Questions/Feedback