Is Summer!

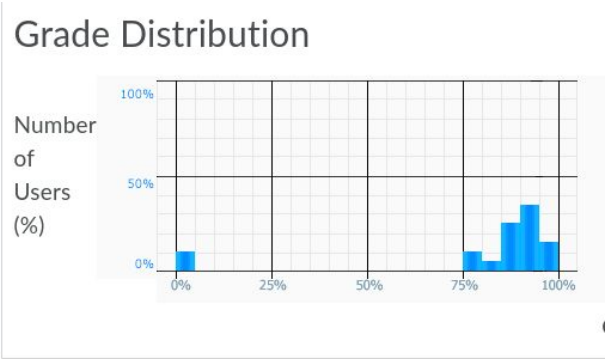
March 29th, 2021

Presentation Highlights

Median: 89.5 !!!

If there were any across the board suggestions...

1) Cohesive Story
    a) Breadth -> Depth -> Why -> Conclusion
2) Lots of time on high level analysis
3) Use of algorithms appropriately (not just because hey why not)



Quick Remaining Agenda

Today - Project 2 Begins

20% HW Review, Presentation Discussion

30% What is Machine Learning, EDA for Machine Learning, Classification (Prediction)

50% Logistic Regression, Intro to other algorithms

4/5

20% Topics in Pandas for Sci-Kit Learn

60% Decision Trees

20% SVM's (maybe)

Take Home EXAM 2

4/12

30% Review Models Thus Far

50% How good are models? How do we tell? What do we tweak? Feature Scaling, Feature Selection

20% What else is out there??

4/19

60% Deep Learning (Neural Networks)

20% Advanced Applications

20% Review, Feature Generation, Steve's Presentation

4/26 Project 2 - Also when are finals?

Thinking about the final project...

8 Minutes (subject to my maths)

1-2 Intro to data

2-3 EDA

3-4 ML & Analysis

Thinking about the final project...

Choosing Data

- Robust enough data to really dig into things and see how distributions of values in various features may/may not impact results.
- Enough to appropriately train a good/great model and be able to discuss success/failures.
- More features (columns) is not always better, but too few is probably not great.
    - Asterisks... images are really just a matrix of pixel values 28x28 for example. So flattening out an image to 28x28=784 columns is what it actually is. Same for text...one sentences is many words (features).
- Usually the more rows the better unless you want to show results in a constrained data problem setting.

Choosing the Problem:

- Most of the datasets I will send out have a clearly marked column for label (supervised learning). This usually comes with a task - classify types of animals or something. (You can do other things but this should be a consideration)
- We will discuss more next week but you want to start thinking about what is important - accuracy, precision, recall, F1 score - in other words, what are the what case scenarios if you are wrong.

Choosing Other Data:

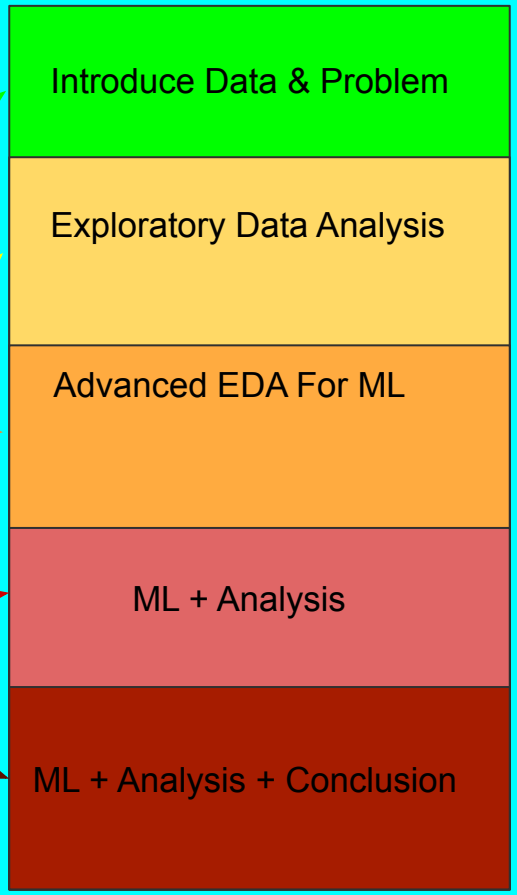
I highly recommend meeting with me or discussing via email why your dataset warrants this task and what problem (ML) you intend to solve with it. In form of HW or early/short 1-1.

Final Project Outline

8 minute presentations to ensure completion on Final Day
HARD STOP...if you are well under...why...

This is a suggested breakdown

- What is your data, what is your problem? 1min
- EDA - Midterm Presentations (what's in the data?) 1-2min
- EDA with a purpose for ML 1-2min
  - Distributions
  - Bootstrapping
  - Balancing
  - Train / Test / K Folds Splits?
- ML Baseline - what comes out of the box 1-2min
  - Describe the algorithm to us - teach me!
- ML Improvement - how did you improve a model? 1-2min
  - Comparisons from 'EDA with a purpose'
  - Explanations + teach me!
  - Wrap up - lessons learned



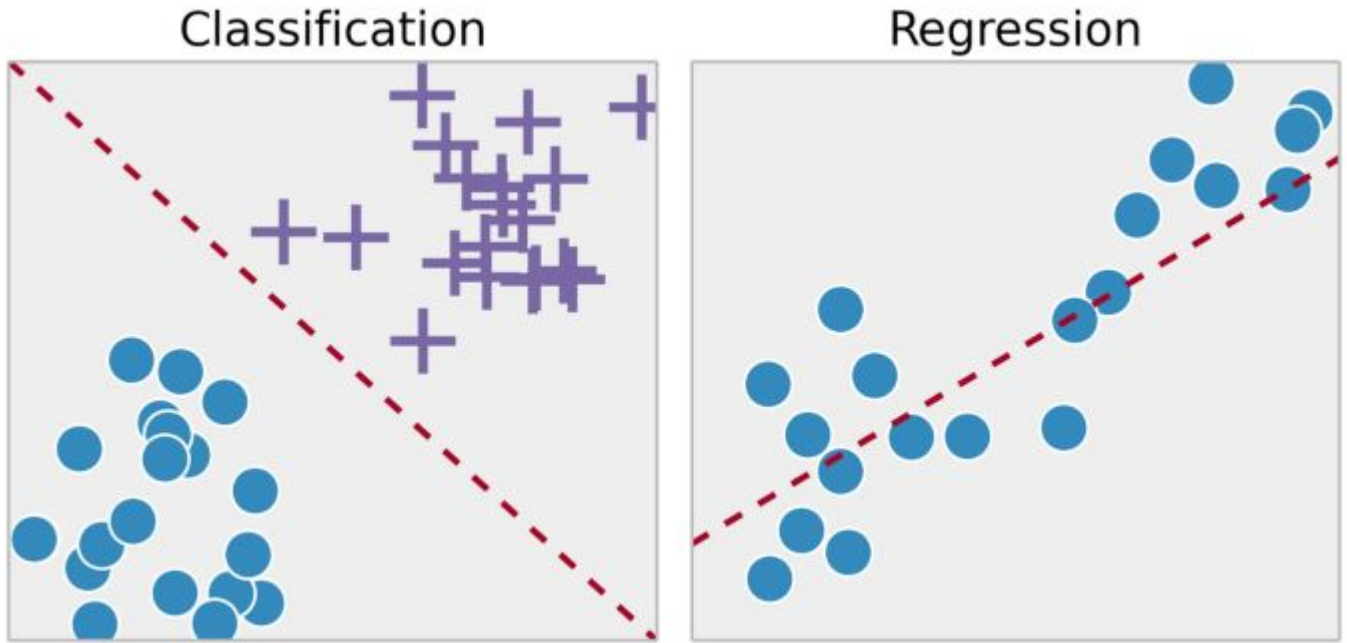
Guest Speaker: Cynthia Frelund

HW Review - Jupyter

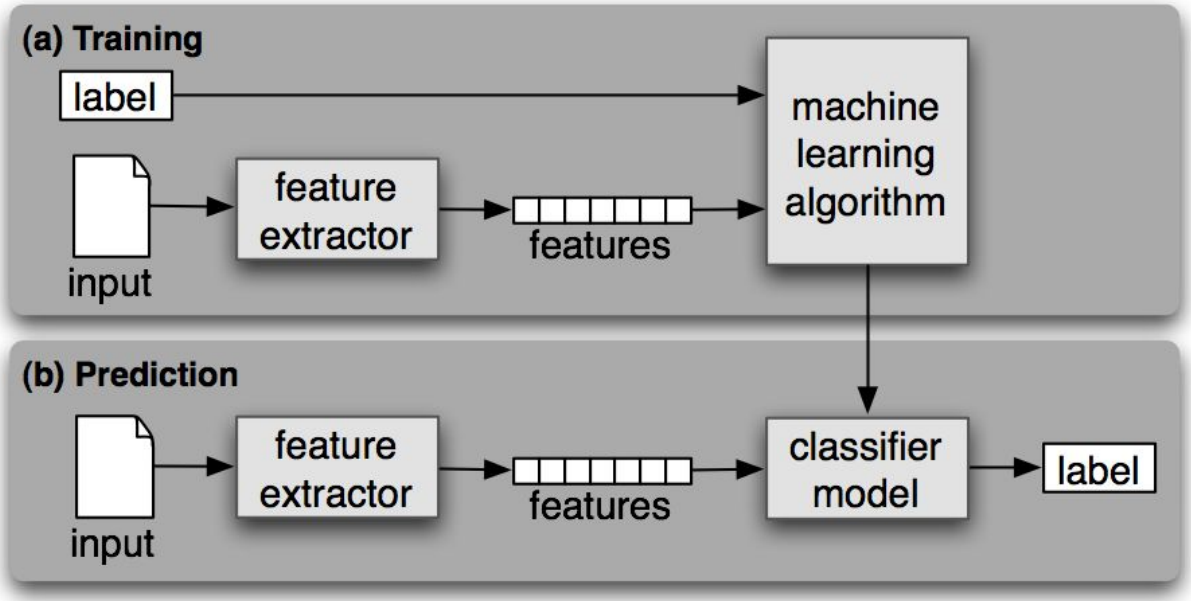
Last EDA Note - Case Study

Reporting at Martiz

Supervised Learning: Classification vs. Regression



Recall our algorithm for algorithms



Remember - the algorithm for our algorithms is also an algorithm...or something like that...

In [24]:
```
from sklearn.svm import SVC

model_name = 'Kernel SVM Classifier'

svmClassifier = SVC(kernel='rbf', gamma= 'auto')

svm_model = Pipeline(steps=[('preprocessor', preprocessorForFeatures),('classifier', svmClassifier)])

svm_model.fit(X_train,y_train)

y_pred_svm = svm_model.predict(X_test)
```

In [18]:
```
from sklearn.linear_model import LogisticRegression

model_name = "Logistic Regression Classifier"

logisticRegressionClassifier = LogisticRegression(random_state=0,multi_class='auto',solver='lbfgs',max_iter=1000)

lrc_model = Pipeline(steps=[('preprocessor', preprocessorForCategoricalColumns),
                            ('classifier', logisticRegressionClassifier)])

lrc_model.fit(X_train,y_train)

y_pred_lrc = lrc_model.predict(X_test)
```

In [217]:
```
from sklearn.naive_bayes import GaussianNB
```

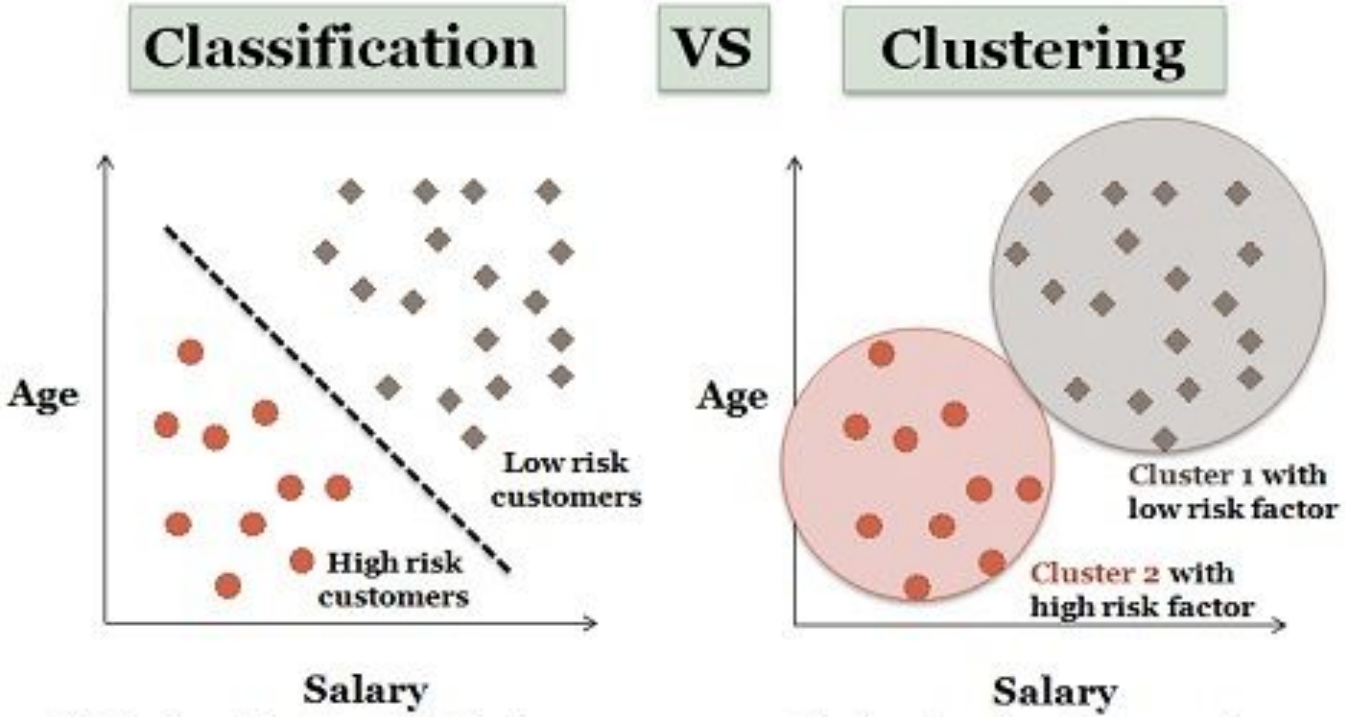
In [218]:
```
classifier = GaussianNB()
```

In [226]:
```
features = zip(data.W[:1600], data.ADJOE[:1600], data.WAB[:1600])
test = zip(data.W[1600:], data.ADJOE[1600:], data.WAB[1600:])
#classifier.fit(features, data.SEED[:1600])
classifier.fit(np.array(data.W[:1600]).reshape(-1,1), data.SEED[:1600].astype(int))
```

Out[226]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [222]:
```
preds = classifier.predict(test)
print(preds)
```

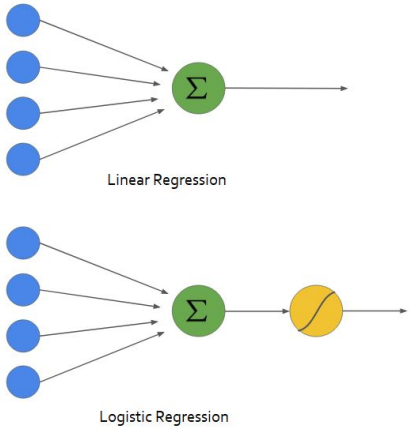
Classification vs. Clustering

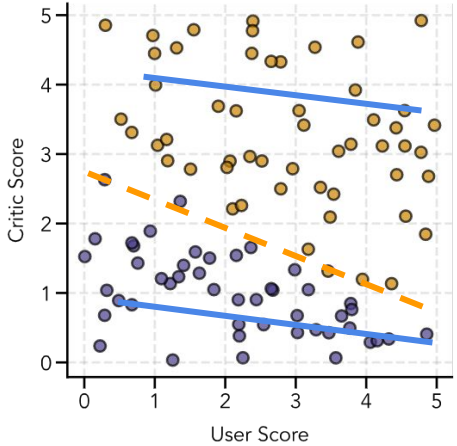


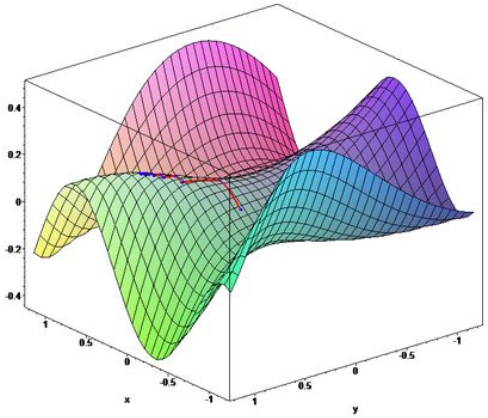
Risk classification for the loan payees on the basis of customer salary

Logistic Regression

$$\mathbf{0} = \beta 0 + \beta 1.x1 + \cdots + \beta n.xn$$







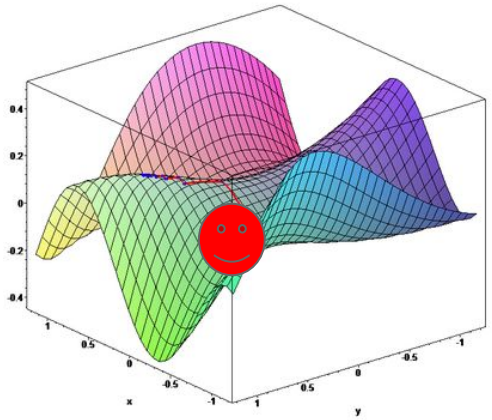
Gradient Descent - ML!

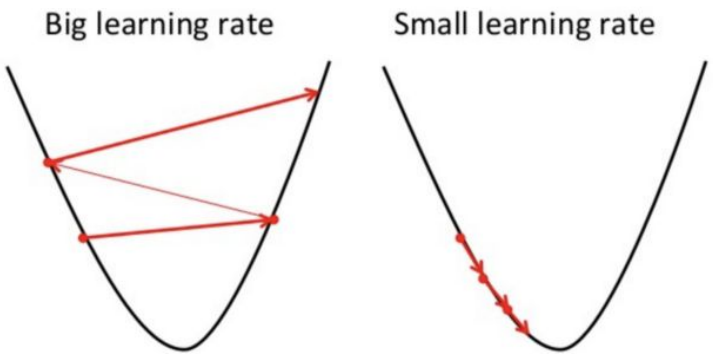
Baby Intro - Gradient Descent

$$J(w_0, w_1, w_2) = \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(h_w(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_w(x^{(i)})) \right]$$

where our **hypothesis function** is:

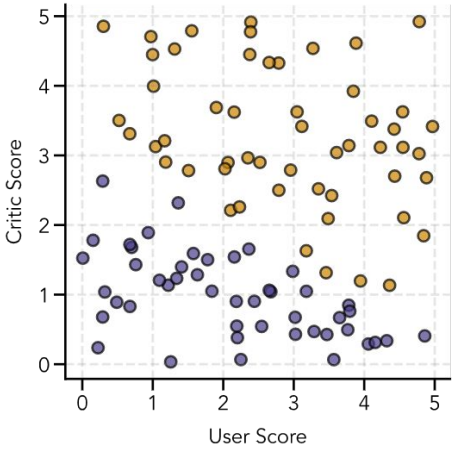
$$h_w(x_1^{(i)}, x_2^{(i)}) = \frac{1}{1 + \exp(-z)} \quad \text{where} \quad z = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)}$$

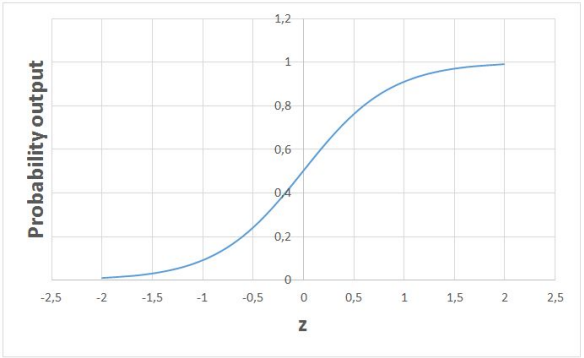


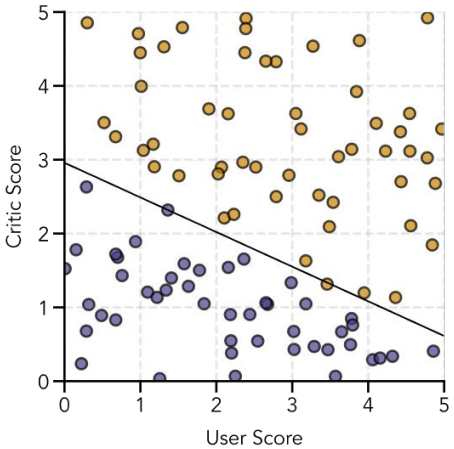


https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

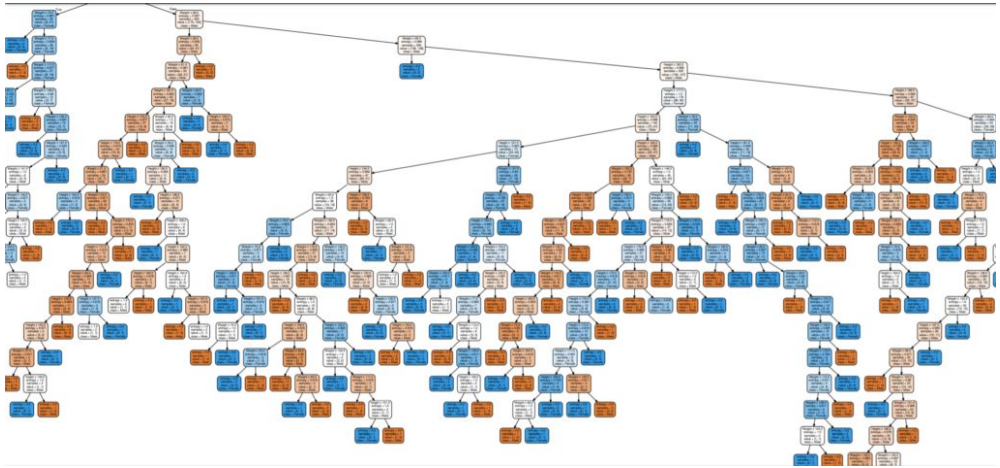
Logistic Regression Output

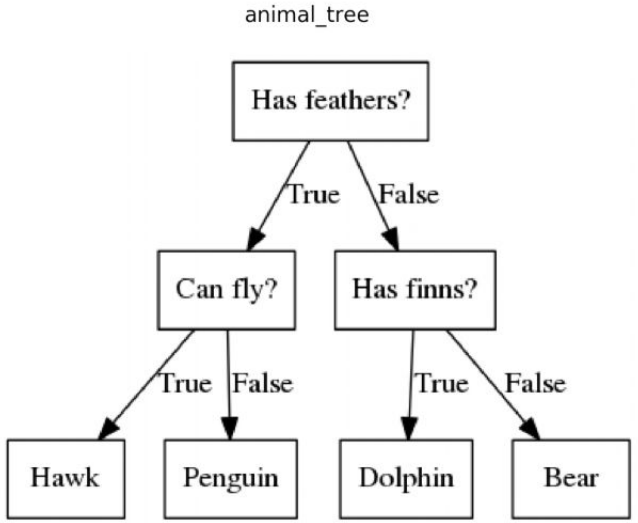




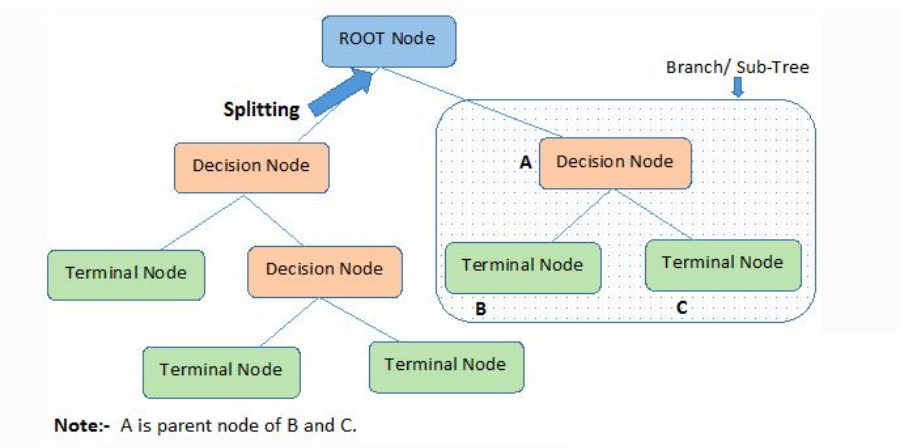


Decision Trees



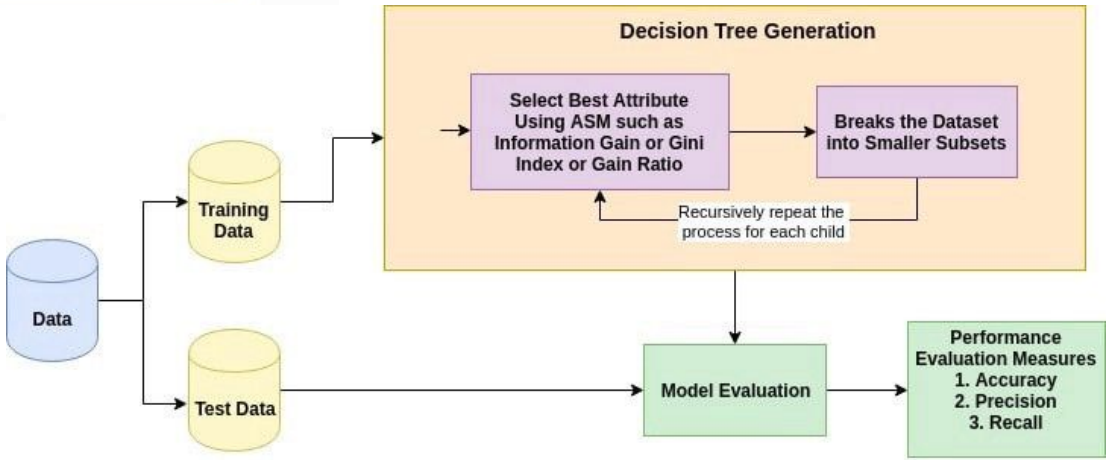


Decision Tree Mechanics

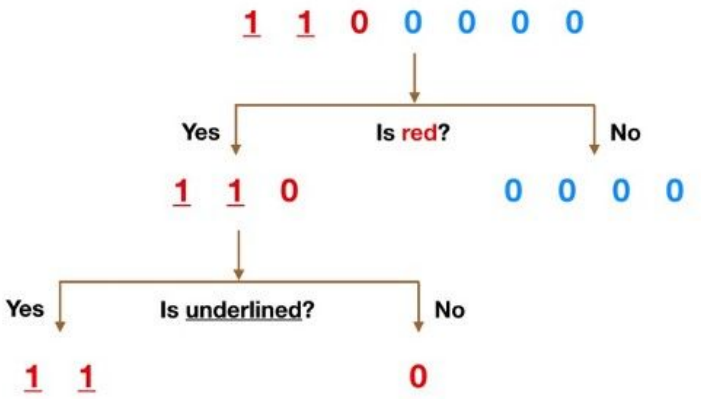


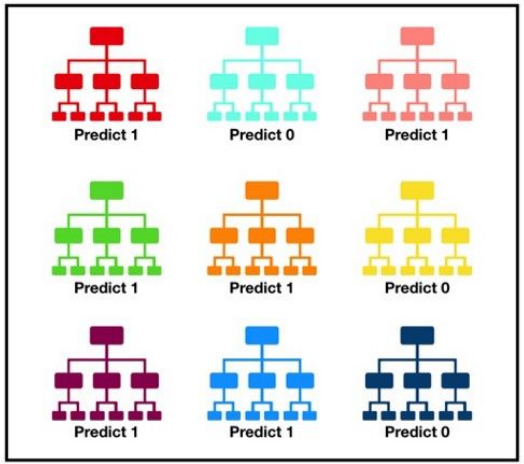
$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$



Decision Trees Become Forests?



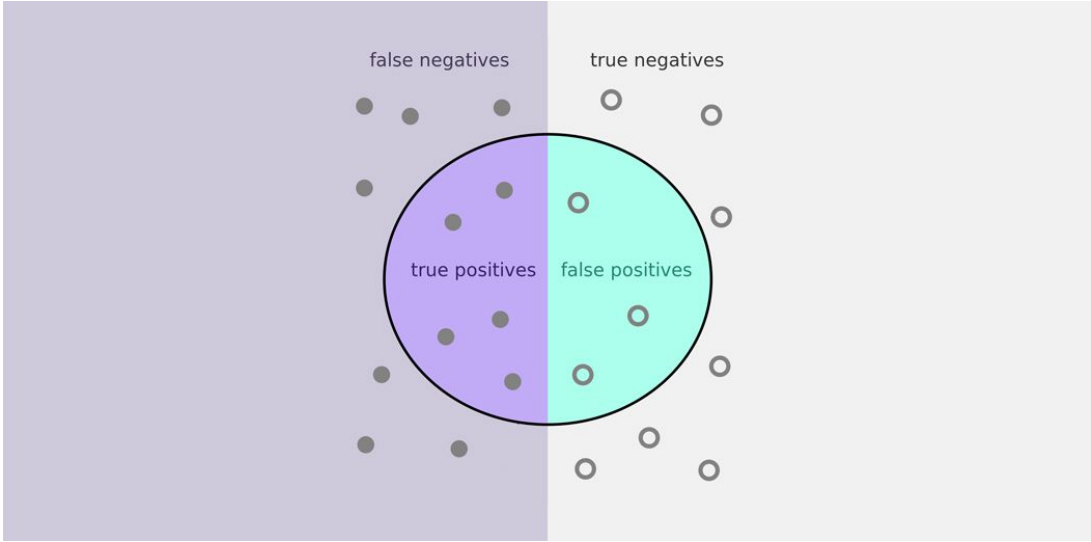


Tally: Six 1s and Three 0s
Prediction: 1

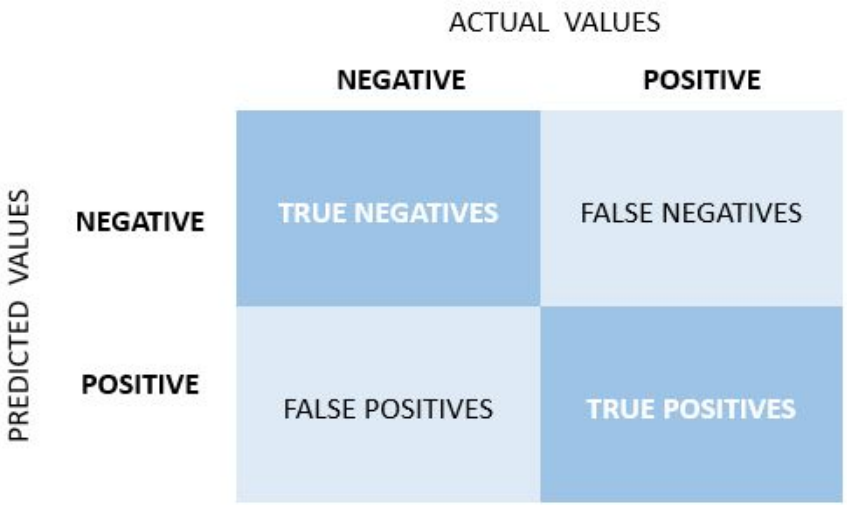
Measure of Goodness - Accuracy, Precision,

Create a model entirely in their head to identify terrorists trying to board flights with greater than 99% accuracy?

Simply label every single person flying as not a terrorist. Given the [800 million average passengers on US flights per year] and the [19 (confirmed) terrorists who boarded US flights from 2000–2017], this model achieves accuracy of 99.9999999%!



How Good Is A Model



Metric Name	Formula from Confusion Matrix
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP+FP}$
Recall, Sensitivity, TPR	$\frac{TP}{TP+FN}$
Specificity, 1-FPR	$\frac{TN}{TN+FP}$
F1	$\frac{2 * precision * recall}{precision + recall}$