Model Assessment

Dr. Ratna Babu Chinnam

Industrial & Systems Engineering

Wayne State University

Motivation

Google Says It Will Address AI, ML Model Bias with Technology called TCAV

By Larry Dignan | May 7, 2019 | ZDNet | Link

- Google CEO Sundar Pichai said the company is working to make its AI and ML models more transparent as a way to defend against bias.
- Pichai outlined a bevy of AI enhancements and moves to put more ML models on devices, but the bigger takeaway for developers and data scientists may be something called TCAV.
- TCAV is short for Testing with Concept Activation Vectors. In a nutshell, TCAV is an interpretability method to understand what signals your neural network models use for prediction.
- In theory, TCAV's ability to understand signals could surface bias because it would highlight whether males were a signal over females and surface other issues such as race, income and location. Using TCAV, one can see how high value concepts are valued.

Judging Model Performance & Robustness

- Accuracy isn’t Everything
  - Timeliness / Lead time to action
  - Class imbalances / Bias
  - Cost differences: Need cost-sensitive learning
- Performance Repeatability
  - Ideal conditions vs real-world
- Variable Selection/Influence Consistency
- False Alarms Fatigue!
- Purely Data-Driven vs Mechanistic/Physics Based Models
- Ensembles: Combine Approaches for Robustness

Model Assessment

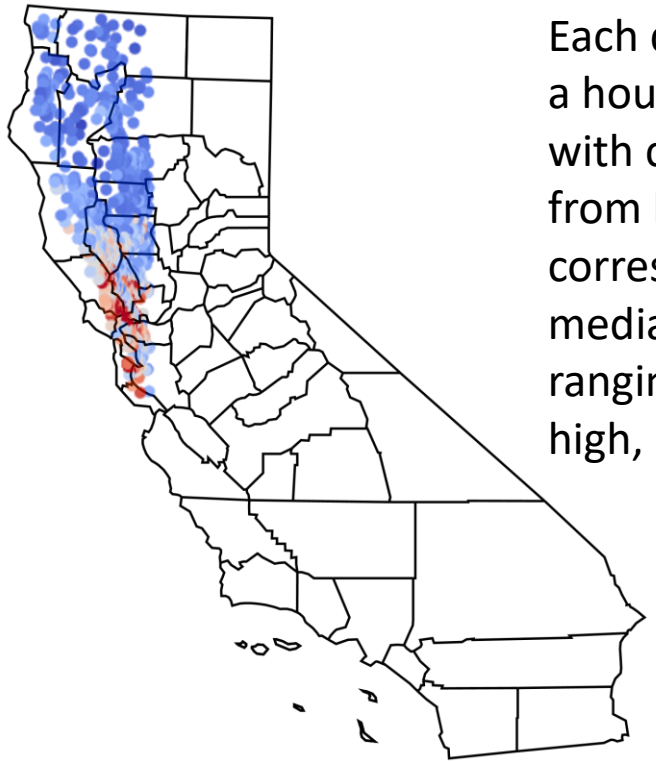
- First Requirement: Dataset offers (best) features to carry out task
- Feature selection/extraction might be necessary
- Small datasets: *K*-fold cross-validation strategy and/or regularization
- Build a model that best addresses the *bias-variance dilemma*
  - For “small” datasets, as one decreases bias, variance increases (and vice versa)
- Effective Model:
  - Selecting most appropriate modeling approach
  - Optimize model complexity (e.g., in MLPs, # layers, # nodes/layer), model parameters (e.g., transfer function)
  - Appropriate learning algorithm and parameters
- Vigorous Testing & Sensitivity Analysis

Some Sources of Model Bias

- Missing Feature Values
  - If your data set has features that have missing values for a large number of examples, that could be an indicator that certain key characteristics of your data set are under-represented.
- Unexpected Feature Values
- Data Skew
  - Geographical Bias Example: If this unrepresentative sample were used to train a model to predict California housing prices statewide, the lack of housing data from southern portions of California would be problematic. The geographical bias encoded in the model might adversely affect homebuyers in unrepresented communities.

	longitude	latitude	total_rooms	population	households	median_income	median_house_value
count	17000.0	17000.0	17000.0	3000.0	3000.0	3000.0	17000.0

California Housing Dataset



Each dot represents a housing block, with colors ranging from blue to red corresponding to median house price ranging from low to high, respectively.

Evaluating for Bias: Example

- Consider a new model developed to predict the presence of tumors evaluated against a validation set of 1,000 patients' medical records
  - 500 records are each from female and male patients
- Calculated metrics separately for female and male patients show stark differences in model performance for each group

Confusion Matrix

True Positives (TPs): 16	False Positives (FPs): 4
False Negatives (FNs): 6	True Negatives (TNs): 974

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{16}{16 + 4} = 0.800$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{16}{16 + 6} = 0.727$$

Female Patient Results

True Positives (TPs): 10	False Positives (FPs): 1
False Negatives (FNs): 1	True Negatives (TNs): 48

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{10}{10 + 1} = 0.909$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{10}{10 + 1} = 0.909$$

Male Patient Results

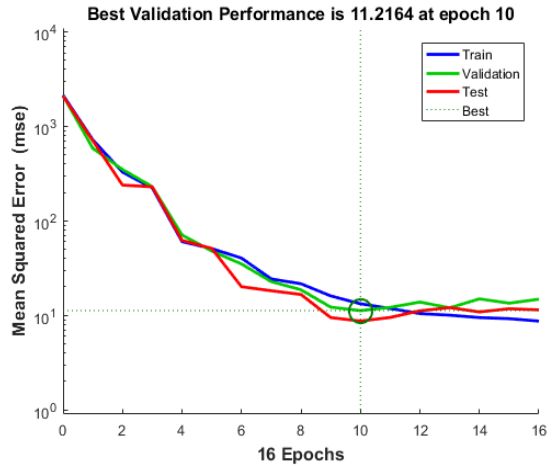
True Positives (TPs): 6	False Positives (FPs): 3
False Negatives (FNs): 5	True Negatives (TNs): 486

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{6}{6 + 3} = 0.667$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{6}{6 + 5} = 0.545$$

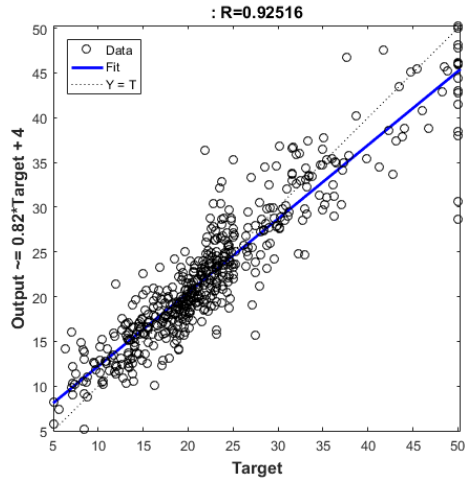
Function Approximation & Regression

- Performance criterion for learning model should be a complete measure
  - Accuracy and precision (e.g., MSE and MAD are complete measures but not expected error)
- Model errors across training, validation, and testing datasets should be consistent
- Regression plots are useful
  - For training, validation, and testing datasets
  - Ideally have a high R^2 and an intercept of zero and slope of 1



Sample Cross-Validation MSE Plot

`Matlab:-> [net,tr] = train(net,x,t); plotperform(tr)`

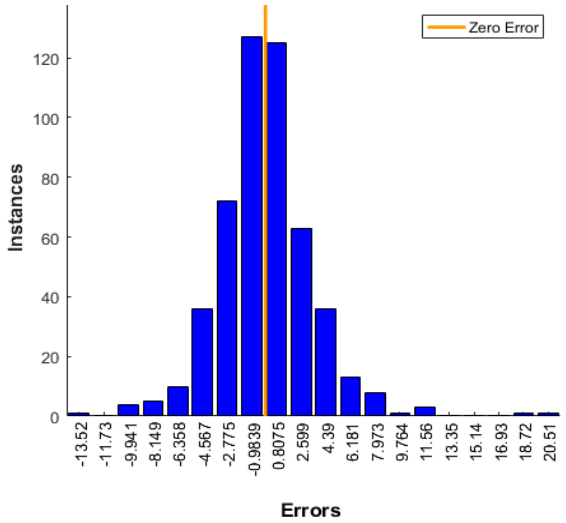


Sample Plot of Actual Vs. Predicted

`Matlab:-> y = net(x); plotregression(t,y)`

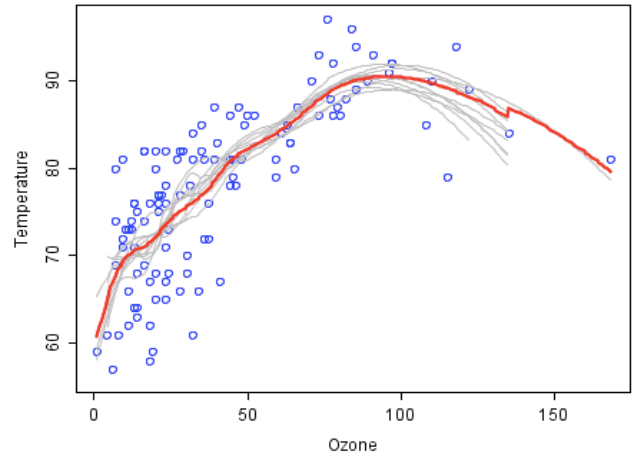
Function Approximation & Regression …

- Error histograms are also quite useful
  - Most errors should be close to zero
  - Significant outliers (large errors) should be investigated further
    - Data collection errors, missing variables
- Track performance across replications
  - Many machine learning models can get stuck in local optimum solutions
  - Training process has to be repeated several times to demonstrate that results are indeed consistent



Sample residual error histogram

`Matlab:-> e = t - y; ploterrhist(e)`

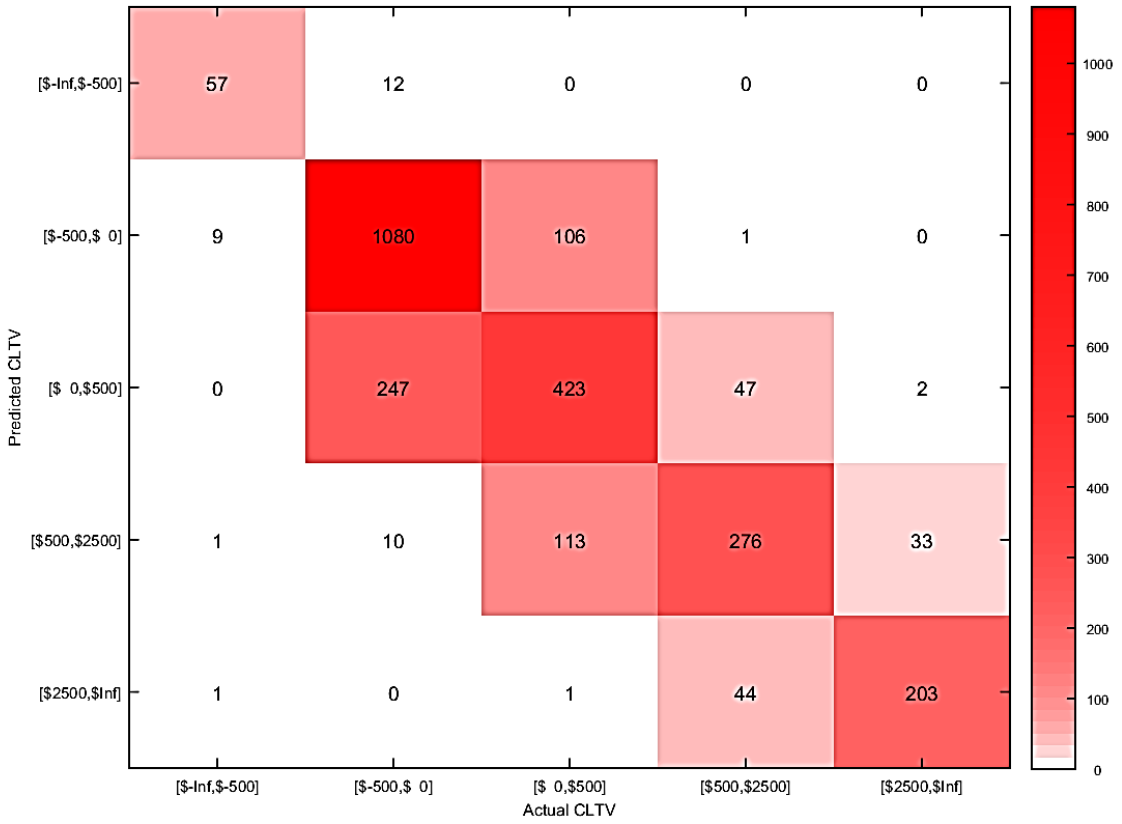


Results from different runs (gray lines)

`Matlab:-> y = net(x); plotregression(t,y)`

Function Approximation & Regression …

- Binning errors by range might be of interest for certain applications
  - Certain types of errors might be more important
  - Example: Table reports actual versus estimated Customer Life-Time Values (CLTV) from a model

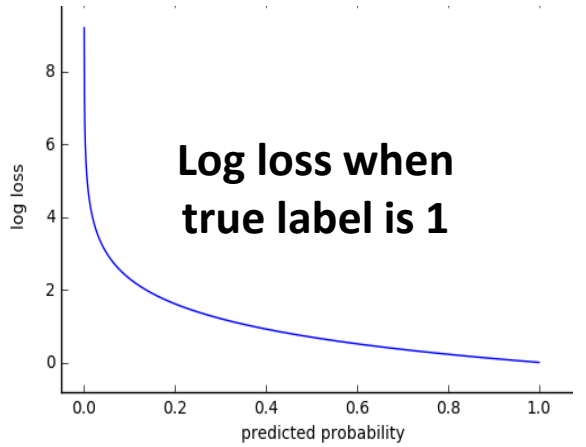


Sample matrix of predicted versus actual output

Pattern Recognition (Classifiers)

- **Cross-entropy loss** appropriate for training classifiers (Link1, Link2)
  - Penalizes extremely inaccurate outputs (p near $1 - t$) with little penalty for fairly correct classifications (p near t)

$$\text{CE Loss} = -\sum_{c=1}^{M} t_{o,c} \log(p_{o,c})$$



M: number of classes

$t_{o,c}$: Binary target (0 or 1) if label c is correct for observation o

$p_{o,c}$: predicted probability observation o is of class c

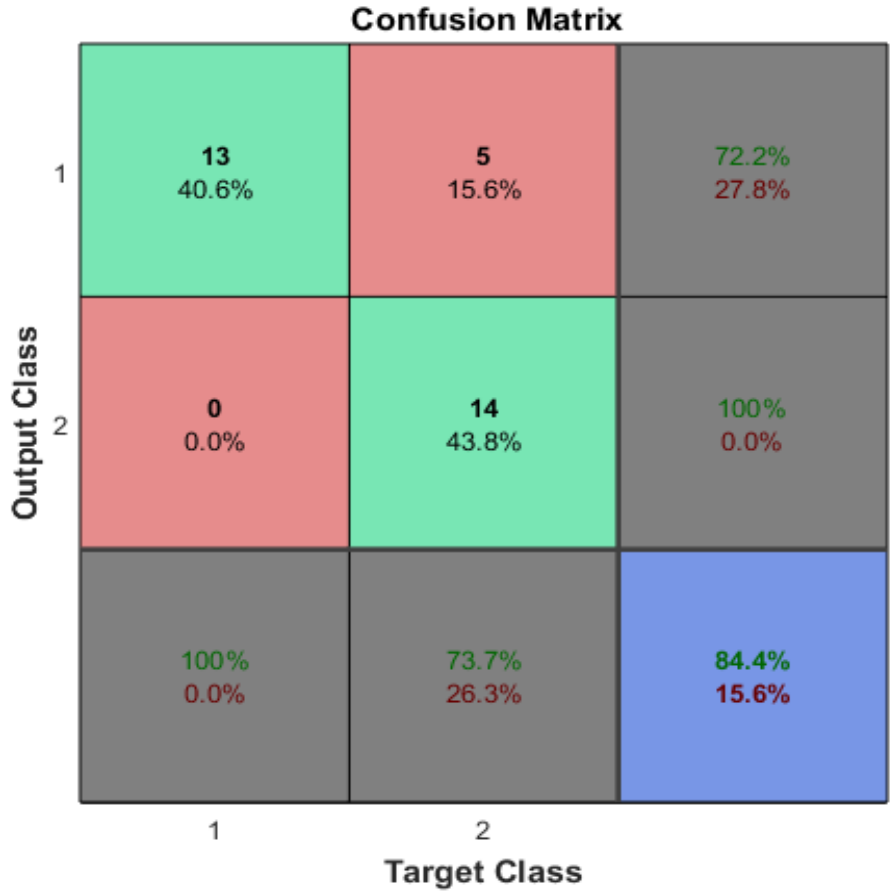
- Example: Let us take a three-class classification problem
  - True Output (Target) for a particular observation: [0 0 1]
  - Predicted Output for this observation: [0.04 0.13 0.83]

$$\text{CE Loss} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

$$= -[0 \times \log(0.04) + 0 \times \log(0.13) + 1 \times \log(0.83)] = -\log(0.83)$$

Pattern Recognition (Classifiers): Confusion Matrix

- An effective performance measure is "***confusion matrix***"
  - Shows %s of correct and incorrect classifications
  - Diagonal entries are correct (green)



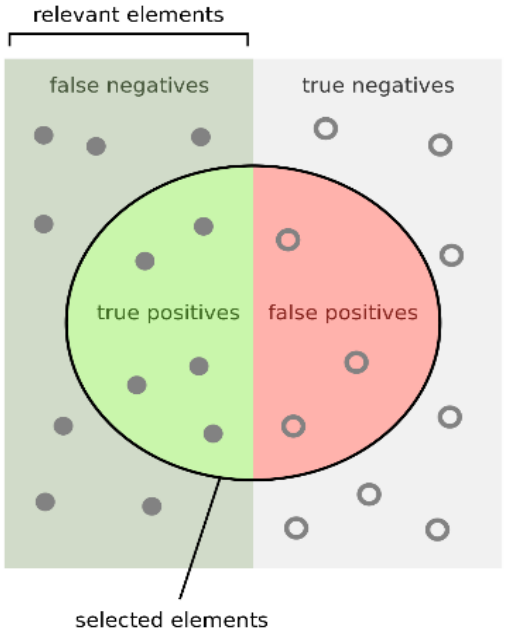
Sample Confusion Matrix for Testing Dataset With Two Target Classes

```
Matlab:-> [net,tr] = train(net,x,t);
testX = x(:,tr.testInd); testT = t(:,tr.testInd); <- Testing Observations
testY = net(testX); plotconfusion(testT,testY)
```

Pattern Recognition (Classifiers): Confusion Matrix

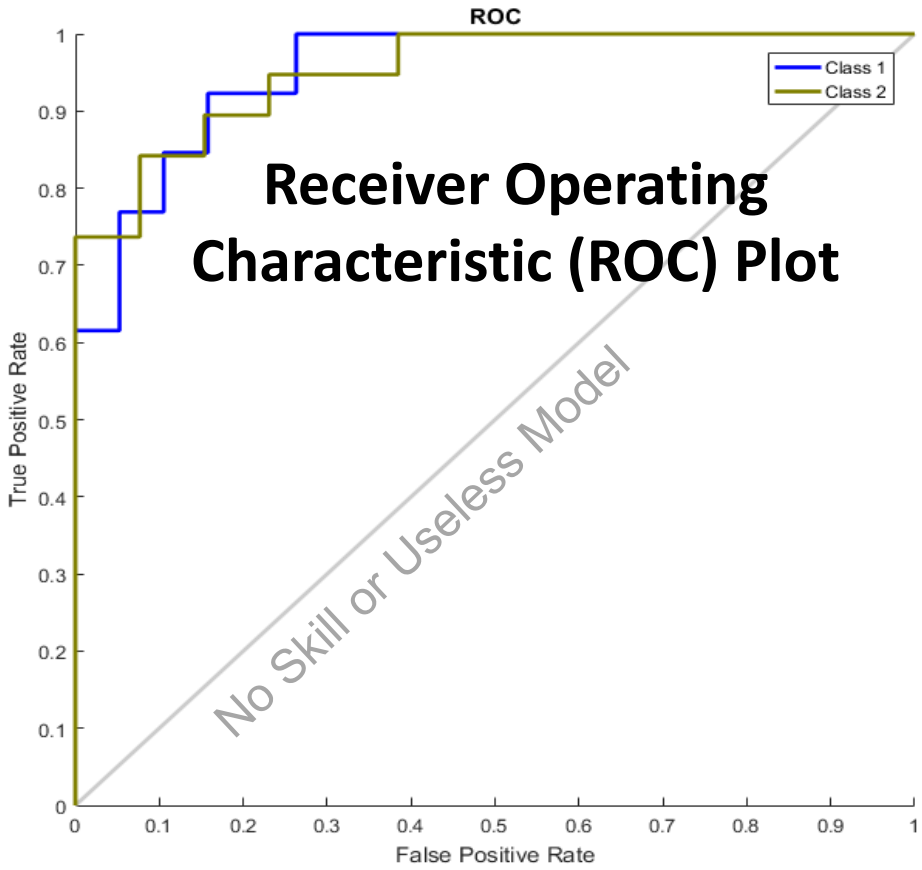
Binary Classification Example

True Condition		Predicted Condition: Predicted +ve	Predicted Condition: Predicted –ve	
	Condition +ve	**True⁺**	**False⁻ (Type-II error)**	**Sensitivity** or **Recall**= $\frac{\sum True^{+}}{\sum Condition^{+}}$
	Condition –ve	**False⁺ (Type-I error)**	**True⁻**	**Specificity** = $\frac{\sum True^{-}}{\sum Condition^{-}}$
	Prevalence $= \frac{\sum Condition^{+}}{\sum Total}$	**Precision** = $\frac{\sum True^{+}}{\sum Predicted^{+}}$	**-ve Predictive Value** $= \frac{\sum True^{-}}{\sum Predicted^{-}}$	**Accuracy** = $\frac{\sum True^{+} + \sum True^{-}}{\sum Total}$



Pattern Recognition (Classifiers): ROC Plot

- Shows how false +ve and true +ve rates change as “threshold” for output classification is varied from 0 to 1
  - Straightforward for binary classification
- Ideally, we are looking for the curve to bow towards the top left corner
  - A completely random model with no utility will yield an ROC plot that is diagonal at 45%
- One should pick a threshold that best balances the two rates to meet the needs of the application
- For multi-class problems, each class will be compared against all other classes
  - One ROC curve for each class (not effective)



Sample ROC Curve Plot for Binary Classification Problem (Cancer Detection)

(Class 1: cancer patients; class 2: normal patients)

`Matlab:-> plotroc(testT,testY)`

Pattern Recognition (Classifiers): Class Imbalance

- **Class Imbalance**: Observations not uniformly distributed across classes
  - Example: Outcome of "strep test" for patients with sore throat condition is mostly "negative" for Streptococcal pharyngitis
- $\boldsymbol{F_1}$ **Score**: Option for binary classification with class imbalance

$$F_1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

- $\boldsymbol{F_1}$ **Score vs. Accuracy**: Weather Prediction Example
  - Suppose only 10% of days are "sunny"
  - A model that (almost) always predicts "not sunny" (say 99.999% of the time, randomly) has an accuracy of 90% whereas it's F_1 score will be 0
    - Precision will be $\approx 0/(0+0)$ NaN and Sensitivity will be $0/(0+10) = 0$

Pattern Recognition (Classifiers): Class Imbalance

Model "Always" Predicts Not-Sunny (randomly, 99.9% of the time):

			Predicted Condition	
		Cases	P(Sunny)	P(Not-Sunny)
			0.001	0.999
True Condition	Sunny	10	0.01	9.99
	Not-Sunny	90	0.09	89.91

Accuracy:	**89.92%**
Sensitivity:	**0.001**
Precision:	**0.1**
F1 Score:	**0.00198**

Perfect Model:

			Predicted Condition	
		Cases	P(Sunny)	P(Not-Sunny)
True Condition	Sunny	10	10	0
	Not-Sunny	90	0	90

Accuracy:	**100.00%**
Sensitivity:	**1**
Precision:	**1**
F1 Score:	**1**

"Decent" Model:

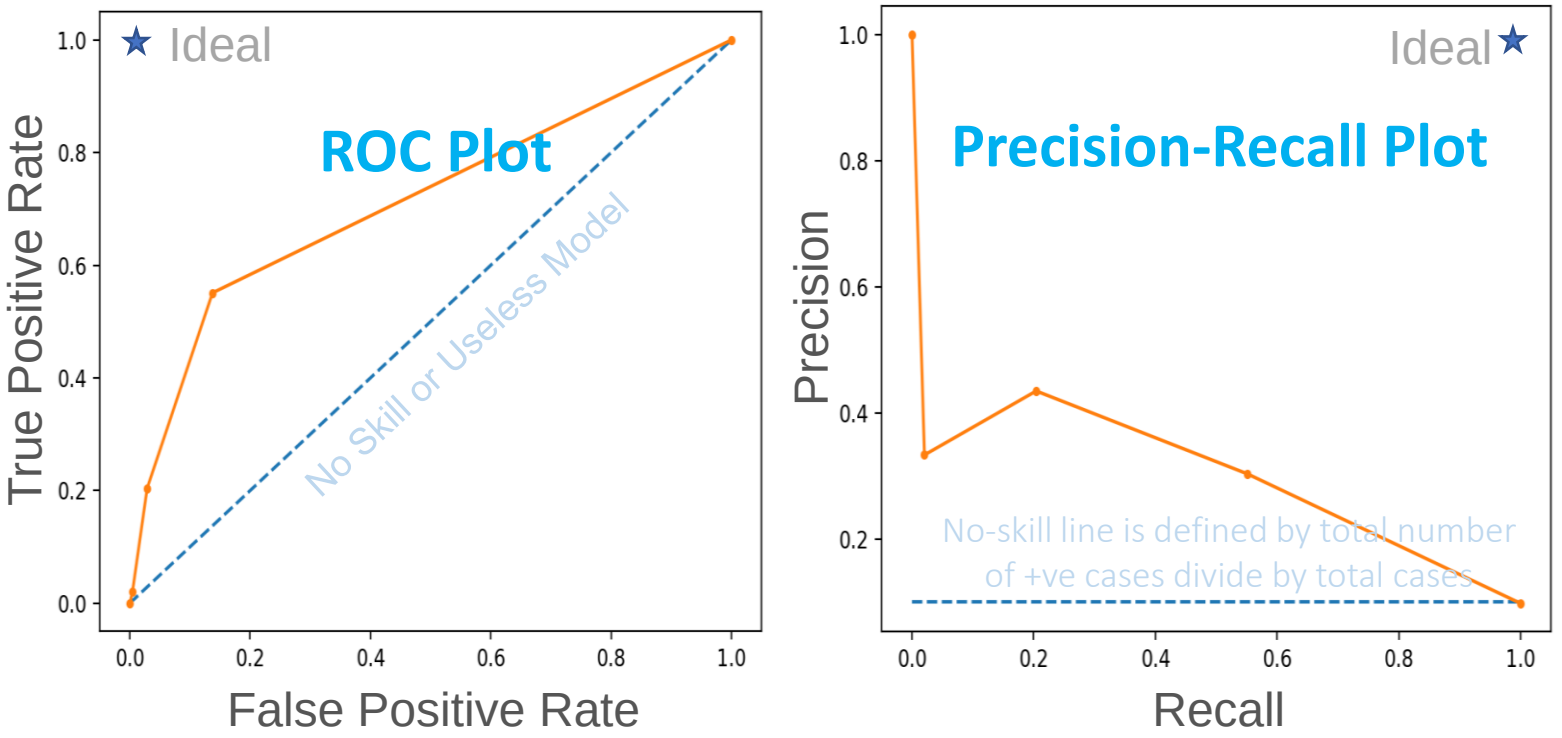
			Predicted Condition	
		Cases	P(Sunny)	P(Not-Sunny)
True Condition	Sunny	10	9	1
	Not-Sunny	90	2	88

Accuracy:	**97.00%**
Sensitivity:	**0.9**
Precision:	**0.818182**
F1 Score:	**0.857143**

Pattern Recognition (Classifiers): Precision-Recall Plot

- ROC plot not that effective under class imbalance
- Precision-Recall curve is more useful
  - Does not make use of true negatives; only concerned with correct prediction of minority class
- For more discussion, check out this link.

This model is useful as suggested by ROC curve. However, it is mostly useful for making few false +ve predictions and there are a lot of false –ve predictions.



ROC and Precision-Recall Curves for a Weak Model Learnt from a Highly Imbalanced Dataset (Source: Link)

Tactics for Addressing Class Imbalance

- Collect more data
- Try different algorithms
- Change performance metric during model optimization (e.g., F_1 Score)
- Resample your dataset (Link)
  - Over-sampling: Add copies of instances from under-represented class
  - Under-sampling: Delete instances from over-represented class
- Generate synthetic examples
  - Synthetic Minority Over-sampling Technique (SMOTE): Creates synthetic samples from minor class by selecting two or more similar instances (using a distance measure) and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances (Python Module | R DMwR)
- Cost-sensitive learning: Add more cost for minority classes
- Try different perspective: Anomaly detection (e.g., 1-class classification)
- Try getting creative: Redefine the problem or classes!

Pattern Recognition (Classifiers): Cost Sensitivities

- **Setting**: In cases where the cost of false +ves is different from cost of false –ves, "***cost sensitive learning***" will be necessary
  - Example: Cost of a false negative for cancer screening could be a 1,000 times higher than a false positive
- **Vigorous Approach**: Objective for the learning model should explicitly account for differences in costs of different error types
- **Heuristic Approach**: (Binary Classification Example)
  - Model learnt giving equal weight for both types of errors
  - Pick a classifier (threshold) based selecting the point on the ROC that minimizes overall cost

REFERENCE: Elkan (2001) Foundations of Cost-Sensitive Learning | PDF

Time-Series Forecasting

- Partition time-series as consecutive segments for training, validation, and testing (with no overlaps) to demonstrate generalization ability
- MSE plots during training are meaningful as well
- Some popular performance measures:
  - ***MAPE (Mean Absolute Percent Error)***:
    - MAPE is scale sensitive and should not be used with low-scale data
      - MAPE will often take on extreme values for small changes
    - MAPE is undefined when Actual value is zero (enters denominator)
  - ***MAD (Mean Absolute Deviation)***:
    - Calculated as average of unsigned errors

$$\left(\frac{1}{n}\sum\frac{|Actual - Forecast|}{|Actual|}\right)*100$$

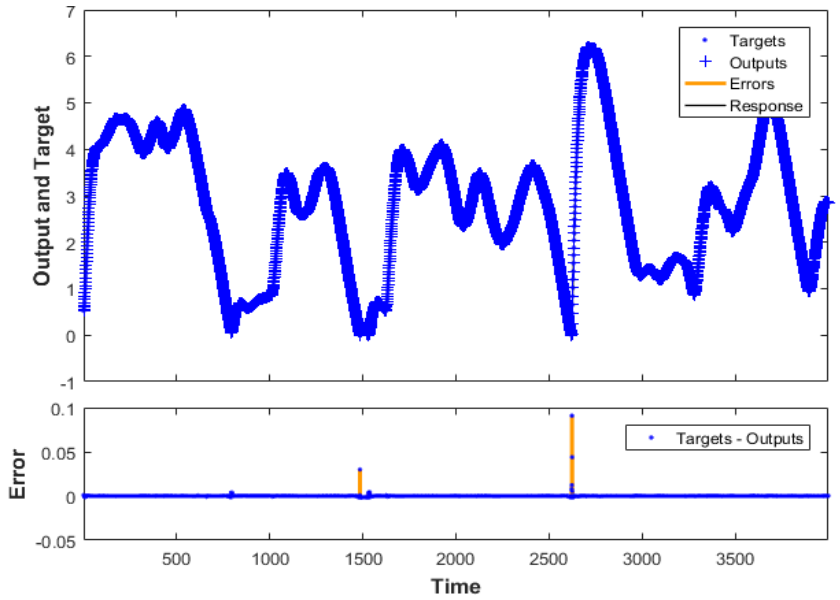
Month	Actual	Forecast	Absolute Percent Error
1	112.3	124.7	11.0%
2	108.4	103.7	4.3%
3	148.9	116.6	21.7%
4	117.4	78.5	33.1%
MAPE			**17.6%**

$$\frac{1}{n}\sum|Actual - Forecast|$$

Month	Actual	Forecast	Absolute Error
1	112.3	124.7	12.4
2	108.4	103.7	4.7
3	148.9	116.6	32.3
4	117.4	78.5	38.9
MAD			**22.08**

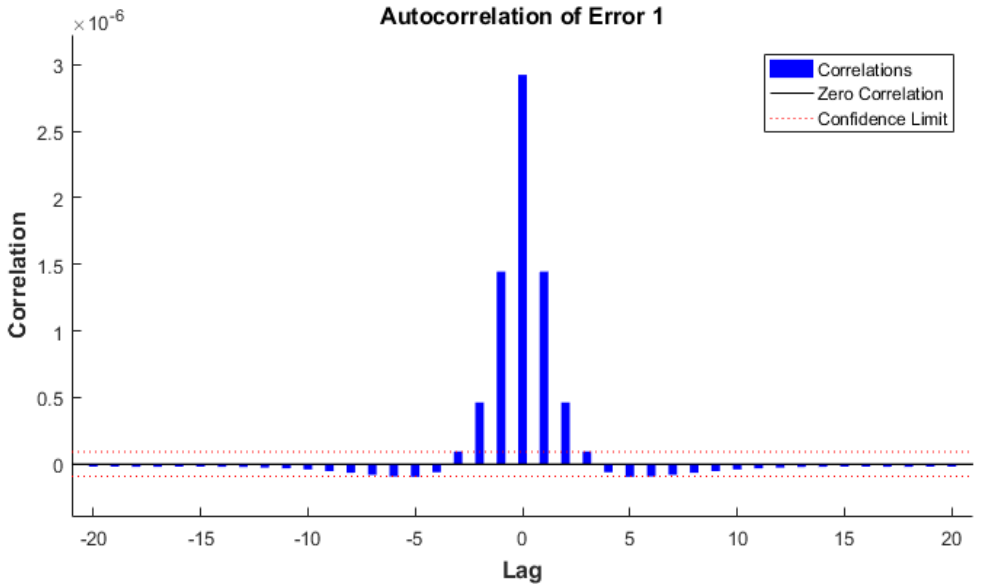
Time-Series Forecasting …

- Plot actual vs predicted values for testing times-series segment along with errors
- Plot correlation of error at time t, $e(t)$ with errors over varying lags, $e(t + lag)$
  - All lines should be short



Sample Forecasting Plot

`Matlab:-> plotresponse(Ts,Y)`



Autocorrelation of Error

`Matlab:-> E = gsubtract(Ts,Y); ploterrcorr(E)`

- Model should outperform a naïve model that predicts $\hat{Y}(n+1) = Y(n)$
- Test accuracy of "recursive" predictions if relevant

Good Data Analysis: Overall Guidelines

Technical

- Look at your distributions
- Consider outliers
- Consider noise
- Look at examples in detail
- Consider practical significance
- Check for consistency over time
- Acknowledge any data filtering

Process

- Separate Validation, Description, and Evaluation/Impact
- Confirm experiment and data collection setup
- Check for what shouldn't change and reproducibility
- Make hypotheses and look for evidence
- Exploratory analysis benefits from end-to-end iteration
- Seek feedback

Mindset

- Data analysis starts with questions, not data or a technique
- Be both skeptic and champion
- Correlation != Causation
- Share with peers first, external consumers second
- Expect and accept ignorance and mistakes

Source: Good Data Analysis | Link