Data Science

Yogesh Kulkarni

Machine Learning Concepts

Machine Learning way of Solution, Mathematically

General Form

$Y = F(X_1, X_2, \ldots, X_p)$

- We have:
    - Observation of quantitative (numerical) response Y
    - Observation of p different predictors $X_1, X_2, \ldots, X_p$
- Find function F between Y and $X = X_1, X_2, \ldots, X_p$

General Form

As one may not get exact function F, approximate it to f, thus introducing some error.

- So, general form: $Y = f(X) + \epsilon$
- f is unknown function of $X_1, X_2, \ldots, X_p$
- ϵ is a random error term, Independent of X, Has mean equal to zero
- Statistical learning refers to approaches for estimating f

Estimate f

- Not concerned if f is linear, quadratic, etc.
- We only care that our predictions are "near accurate".
- So, f often treated as a black box.
- The aim is of minimizing reducible error

How do we estimate f?

Most statistical learning methods classified as:

- Parametric
- Non-parametric

How do we estimate f?

- Parametric:
  - Assume that the functional form, or shape, of f in linear in X, $f(X) = \sum_{i=1}^{p} \beta_i X_i$
  - This is a linear model, for p predictors $X = X_1, X_2, \ldots, X_p$
  - Model fitting involves estimating the parameters $\beta_0, \beta_1, \ldots, \beta_p$
  - Reduces the problem of estimating f to estimating a set of parameters
- Non-Parametric: No explicit assumptions about the functional form of f

Comparison

Parametric

- Only need to estimate set of parameters
- Bias such as linear model
- May not fit data well

Non-Paremetric

- Potential of many shapes for f
- Lots of observations needed
- Complex models can overfit

Simple Example

- Linear regression is a simple and useful tool for predicting a quantitative response. The relationship between input variables $X = (X_1, X_2, \dots X_p)$ and output variable Y takes the form: $Y \approx \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + \epsilon$
- $\beta_0, \beta_1, \dots \beta_p$ are the unknown coefficients (parameters) which we are trying to determine.
- The best coefficients will lead us to the best "fit", which can be found by minimizing the residual sum squares (RSS), or the sum of the difference between the actual *i*th value and the predicted *i*th value.
- $RSS = \sum_{i=1}^{n} = e_i$, where $e_i = y_i - y'$

How to find best fit?

Matrix Form:

- We can solve the closed-form equation for coefficient vector w: $w = (X^T X^{-1}) X^T Y$.
- X represents the input data and Y represents the output data.
- This method is used for smaller matrices, since inverting a matrix is computationally expensive.

How to find best fit?

Gradient Descent:

- First-order optimization algorithm.
- We can find the minimum of a convex function by starting at an arbitrary point and repeatedly take steps in the downward direction, which can be found by taking the negative direction of the gradient.
- After several iterations, we will eventually converge to the minimum. In our case, the minimum corresponds to the coefficients with the minimum error, or the best line of fit.
- The learning rate α determines the size of the steps we take in the downward direction.

How to find best fit?

Gradient descent algorithm in two dimensions (meaning with 2 features, as example):

- Repeat until convergence.

$$w_0^{t+1} = w_0^t - \alpha \frac{\partial J(w_0, w_1)}{\partial w_0}$$
$$w_1^{t+1} = w_1^t - \alpha \frac{\partial J(w_0, w_1)}{\partial w_1}$$

- For non convex functions, gradient descent no longer guarantees an optimal solutions since there may be local minimas.
- Instead, we should run the algorithm from different starting points and use the best local minima we find for the solution.

How to find best fit?

Stochastic Gradient Descent:

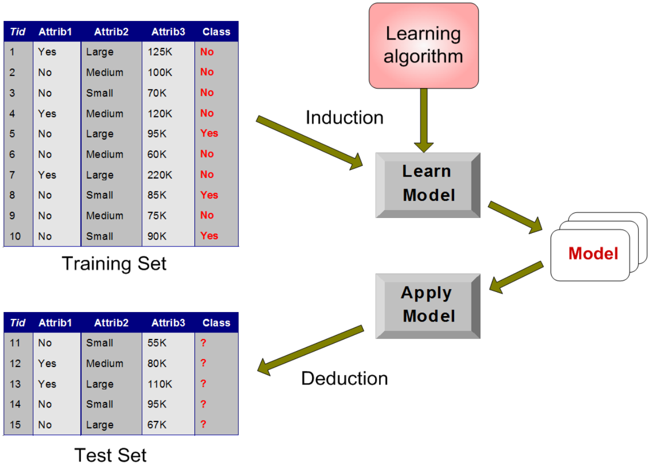
- Instead of taking a step after sampling the entire training set, we take a small batch of training data at random to determine our next step.
- Computationally more efficient and may lead to faster convergence.

Cross Validation

Background: Imp Terms

- **Learning algorithm:** identifies a model that best fits the relationships between inputs and outputs.
- **Training set:** consists of records with features and with/without class labels
- **Test set:** consists of records with only features, class labels to be computed.

Training - Testing



Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

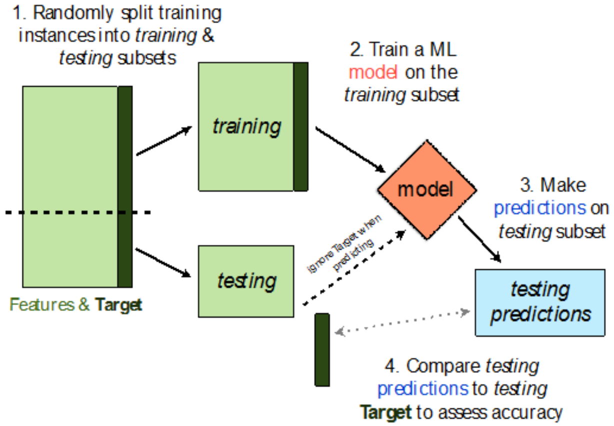
Test Set

(Reference: Data Mining Classification - Gun Ho Lee)

Cross Validation: Measuring Performance

- Labeled training data is finite.
- The model built needs to be validated.
- So, training data itself is split
  - Use one part for model building: Training Data.
  - Use the other part: Testing Data. (actually called as Validation Data, but within Cross Validation process, it can be called as Testing data. Otherwise Testing data is the one which does not have output values. Validation data has output values for comparison.)

Machine Learning Technique

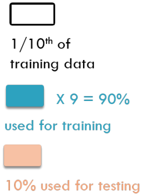

1. Randomly split training instances into *training* & *testing* subsets
Features & **Target**
training
testing
2. Train a ML model on the *training* subset
model
ignore Target when predicting
3. Make predictions on *testing* subset
testing predictions
4. Compare *testing* predictions to *testing* **Target** to assess accuracy

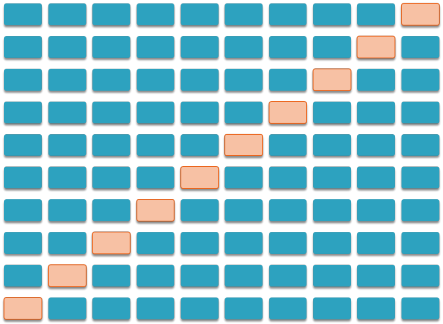
Cross Validation

How to divide dataset into Training, Validation, Testing sets?

- Problems?
  - Validation set should “represent” the Training set.
  - “Lucky split” is possible: Difficult instances were chosen for the training set, Easy instances put into the testing set
- Multiple evaluations using different portions of data for training and validating/testing
- k-fold Cross Validation

K-Fold





Cross Validation Error Estimate: $= \frac{1}{k} \sum Error_i$

Note: Here weights or all 10 models are neither averaged nor the last weights are taken as final model. [PTO]

Practical Tip

- Cross Validation is NOT used for Model Building, but choosing best model amongst choices such as Linear models, decision tree, SVM, etc.
- Once a specific model with known hyper parameters is finalized using least average error, a new model is built using FULL training data.
- Not splits are done there.

Machine Learning Evaluation

Evaluation

Given Actuals and Predictions, how to find out how good the performance was?

- Accuracy: Misclassification Rate
- Confusion Matrix
- Precision, Recall, F1

Evaluation Metrics: Accuracy

Accuracy

Model Evaluation on Test Set (Regression) - Mean Squared Error

- Mean Squared Error: measuring the “quality of fit”
- Will be small if the predicted responses are very close to the true responses $MSE = \frac{1}{n}\sum(y_i - f(x_i))^2$

Accuracy

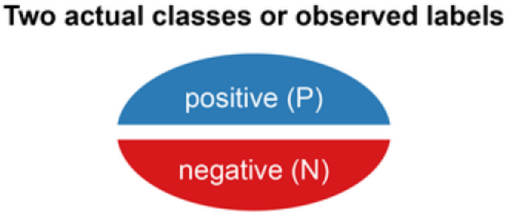
How should classifier be quantitatively evaluated?

- Misclassification Rate $= \frac{NumIncorrectPrediction}{TotalPrediction}$
- Issues with misclassification rate?
  - If Accuracy is used: Example: in CC fraud domain, there are many more legitimate transactions than fraudulent transactions
  - Then: Classifier that predicts every transaction as GOOD would have 99% accuracy! Seems great, but it's not

Evaluation Metrics: Confusion Matrix

Evaluation Metrics

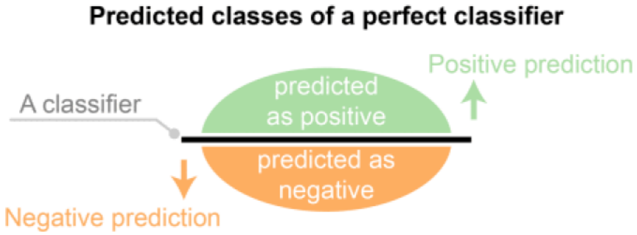
There are predicted labels (Positive / negative) as well as actual labels (Positive / negative) .



In binary classification, a test dataset has two labels; positive and negative.

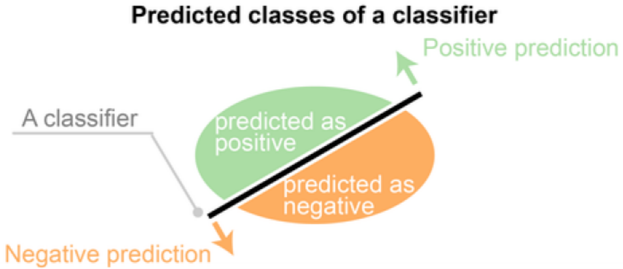
Evaluation Metrics

The predicted labels will be exactly the same if the performance of a binary classifier is perfect, but it is uncommon to be able to develop a perfect binary classifier that is practical for various conditions.



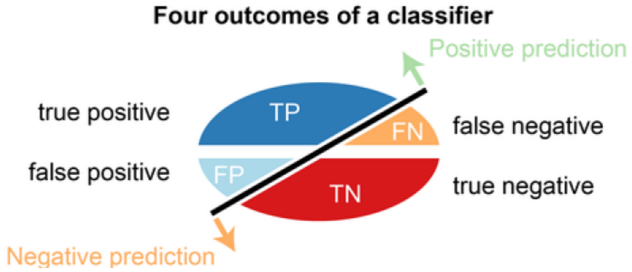
Evaluation Metrics

Hence, the predicted labels usually match with part of the observed labels



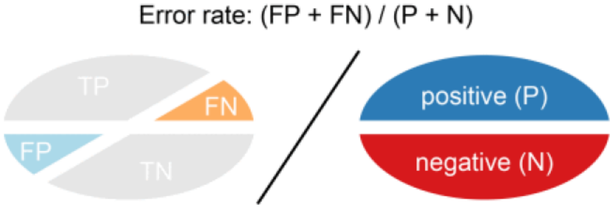
Evaluation Metrics

Classification of a test dataset produces four outcomes - true positive, false positive, true negative, and false negative.



Error rate (ERR) and accuracy (ACC) are the most common and intuitive measures derived from the confusion matrix.

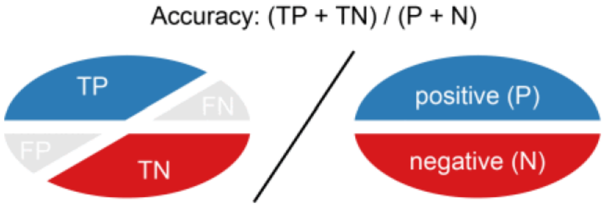
Error rate (ERR)



It is calculated as the number of all incorrect predictions divided by the total number of the dataset. The best error rate is 0.0, whereas the worst is 1.
$ERR = \frac{FP+FN}{TP+TN+FN+FP} = \frac{FP+FN}{P+N}$

Accuracy

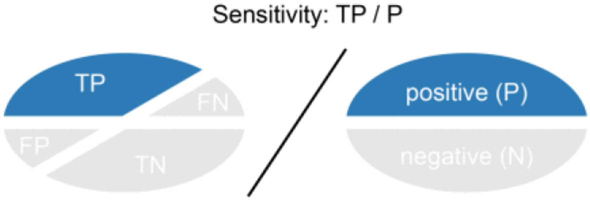
It is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.0, whereas the worst is 0.0. It can also be calculated by $1 - ERR$.



$$Accuracy = \frac{TP+TN}{P+N}$$

Sensitivity (Recall or True positive rate)

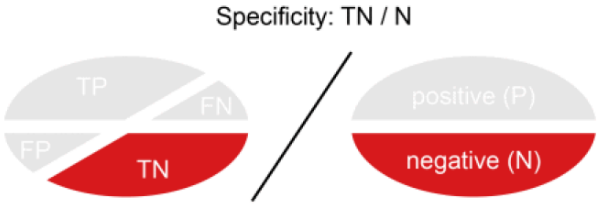
Sensitivity (SN) is calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall (REC) or true positive rate (TPR). The best sensitivity is 1.0, whereas the worst is 0.0.



$SN = \frac{TP}{TP+FN} = \frac{TP}{P}$

Specificity (True negative rate)

Specificity (SP) is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR). The best specificity is 1.0, whereas the worst is 0.0.



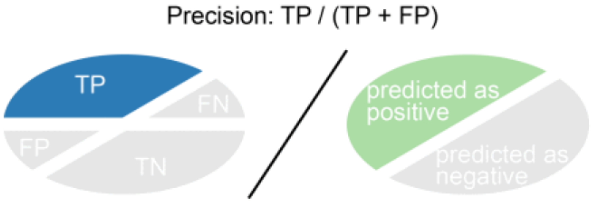
$SP = \frac{TN}{TN+FP} = \frac{TN}{N}$

Sensitivity vs Specificity

- For sensitivity we will look oat only *P, all that were predicted positive, rest are all in the negatives group.
- For highly sensitive test, 100% sensitive, within the Negatives group, there is no mistake.
- Meaning anyone who has tested negative, they surely don't have disease.
- All are truly negative. Meaning FN are 0. Recall thus can be made 1 by marking all positive. So there is no one in the Negatives group. So no FN. But thats not good.
- But still its not a perfect test, because within Positives, you may have some FPs.
- Similarly 100% Specific test: We will send all negative tested guys to a different group. Within them we had FN guys as well. The Positives were perfect. All of them had disease. $FP = 0$. So those who tested positive surely had the disease.
- In reality there is no perfect test!!

Precision (Positive predictive value)

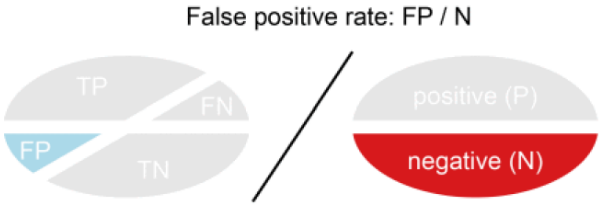
Precision (PREC) is calculated as the number of correct positive predictions divided by the total number of positive predictions. It is also called positive predictive value (PPV). The best precision is 1.0, whereas the worst is 0.0.



$PREC = \frac{TP}{TP+FP}$

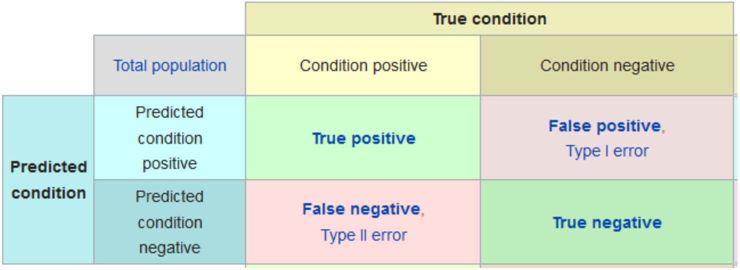
False positive rate

False positive rate (FPR) is calculated as the number of incorrect positive predictions divided by the total number of negatives. The best false positive rate is 0.0 whereas the worst is 1.0. It can also be calculated as 1 - specificity.



$FPR = \frac{FP}{TN+FP} = 1 - SP$

Example



		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

Type I Error

Type I Error is equivalent to False Positive (FP).

- Test is Positive but actually diesese is not there.
- A Fire alarm going off when in fact there is no fire.

This kind of error is synonymous to “believing a lie” or “a false alarm”.

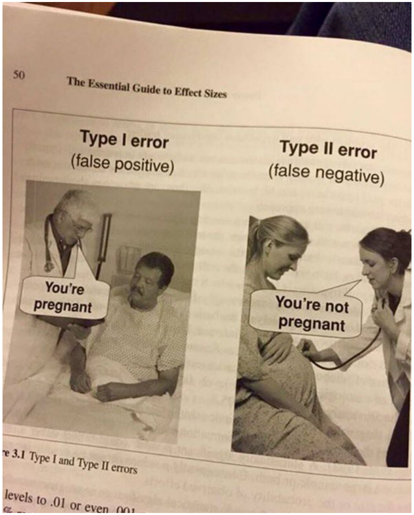
Type II Error

Type II Error is equivalent to False Negative (FN).

- Test is Negative but actually diesese is there..
- A Fire breaking out and the fire alarm does not ring.

This kind of error is synonymous to “failing to believe a truth” or “a miss”.

Errors



50 The Essential Guide to Effect Sizes

re 3.1 Type I and Type II errors

levels to .01 or even .001

Evaluation Metrics: Precision and Recall

Precision and Recall

- Precision: fraction of records that are positive in the set of records that classifier predicted as positive
- Interpretation: If the prediction is very stringent then getting all of them correct is likely, so more precise. Less (no) mis-predictions (FP) happens.
- Recall: fraction of positive records that are correctly predicted by classifier, out of all actual.
- Interpretation: Criteria is relaxed so that maximum catch is made. Less (no) mis-classification (FN) happens.

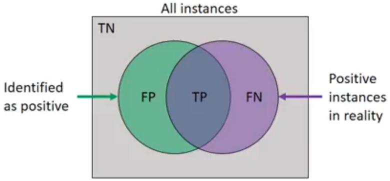
Precision and Recall

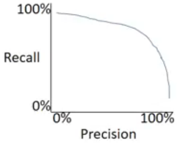
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{F-measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

High precision: relevant results >> irrelevant results
High recall: we got most of the relevant results
F-measure: harmonic mean of precision and recall





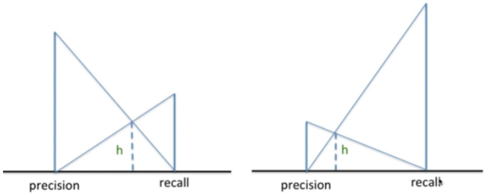
10

1: (Powers, 2007)

(Reference: Text Mining - Jeff Shaul)

F1 Measure

- F-measure: combines precision and recall into a single metric, using the harmonic mean
- Harmonic Mean of two numbers tends to be closer to the smaller of two numbers, so the only way F1 is high is for both precision and recall to be high. $F_1 = \frac{2rp}{(r+p)} = \frac{2\times TP}{(2\times TP+FP+FN)}$

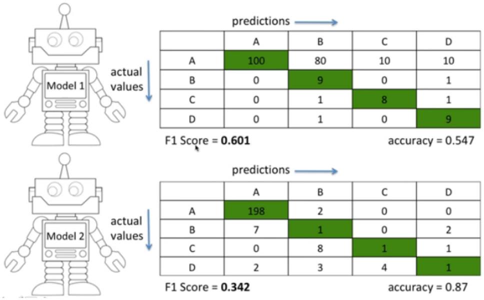


h is half the harmonic mean

Even if anyone is very high, we get balanced F1 score
(Ref: Performance measure on multiclass classification - Minsuk Heo)

F1 Measure

- Model 1's F1 is better but Accuracy is lower than Model 2. Which is better?
- F1 is better as it takes care of any imbalance in the targets.


Model 1
actual values
predictions
Model 2
actual values
predictions

	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

F1 Score = **0.601** accuracy = 0.547

	A	B	C	D
A	198	2	0	0
B	7	1	0	2
C	0	8	1	1
D	2	3	4	1

F1 Score = **0.342** accuracy = 0.87

(Ref: Performance measure on multiclass classification - Minsuk Heo)

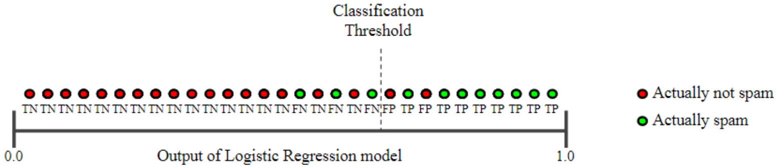
Precision and Recall: Tug of War

Precision and Recall: A Tug of War

- Recall and Precision seem to be opposing each other.
- Just knowing one value is not enough, but both
- And we need to do better at both.

Precision and Recall: A Tug of War

- Explore this notion by looking at the following figure, which shows 30 predictions made by an email classification model.
- Those to the right of the classification threshold are classified as "Spam", while those to the left are classified as "not Spam."


Classification Threshold
TN TN TN TN TN TN TN TN TN TN TN TN TN TN TN FN TN FN TN FN FP TP FP TP TP TP TP TP TP TP
0.0
Output of Logistic Regression model
1.0
Actually not spam
Actually spam

Precision and Recall trade-off

- Typically to increase precision for a given model implies lowering recall, though this depends on the precision-recall curve of your model.
- Generally, if you want higher precision you need to restrict the positive predictions to those with highest certainty in your model, which means predicting fewer positives overall (which, in turn, usually results in lower recall).
- If you want to maintain the same level of recall while improving precision, you will need a better classifier.

Precision and Recall trade-off : Summary

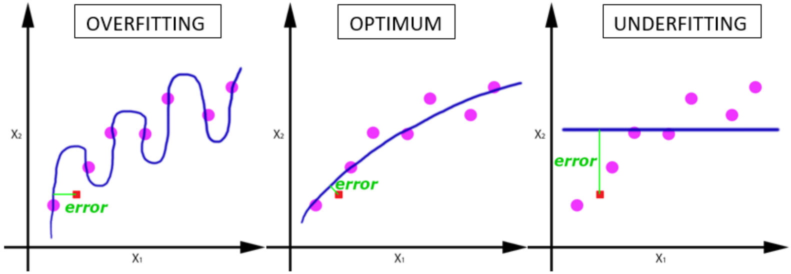
- Intuitively, if you cast a wider net (meaning your classifier threshold is relaxed), you will detect more relevant documents/positive cases (higher recall)
- But you will also get more false alarms (lower precision).
- If you classify everything in the positive category, you have 100% recall (no FN here), a bad precision because some of them could be wrong, so there will be many FPs, making precision lower.
- If you adjust threshold strict so as to have good precision, you will have lower making of positives and mark many actually positive values as negative, falsely. Making more FNs, thus lower recall.

Machine Learning Model Generalization

Generalization in Machine Learning

- Generalization refers to how well the concepts learned by a machine learning model apply to specific examples not seen by the model when it was learning.
- The goal of a good machine learning model is to generalize well from the training data to any data from the problem domain.
- This allows us to make predictions in the future on data the model has never seen
- In statistics, a fit refers to how well you approximate a target function.

Under-fitting vs Over-fitting



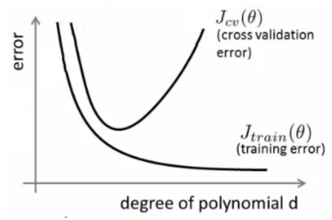
(Reference: What is Machine Learning? - An Introduction- itdadao)

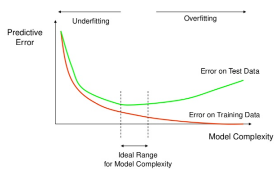
Over-fitting

- Over-fitting: occurs when model “memorizes” training data
- Model does not generalize to the overall problem
- This is bad! We wish to avoid over-fitting
- Techniques: Regularization (Penalty) and Drop Out (Neural Network)

Detection

Roughly speaking, over-fitting typically occurs when the ratio $\frac{ComplexityOfTheModel}{TrainingSize}$ is too high.




How Overfitting affects Prediction

- If your data is in two dimensions, you have 10000 points in the training set

Solution

- In many texts, “iterations” is used on X axis, which is easy to imagine in case of Neural Networks, where each epoch refines model by fitting better question/weights.
- In case of Machine Learning, there no iterations. How to detect Over-fitting in case of Polynomial Regression?
- Initially when, polynomial equation to start with is simple, it has not fit the data yet, so its under fitting phase. Losses are HIGH for both, training as well as testing (validation)
- As more levels get added in tree or more degrees in polynomial regression, under-fitting reduces and losses come down.
- At one point, training loss still reduces but validation los starts jumping up. Thats over-fitting. Model is fitting training data TOO tightly and is OFF the validation data.

(Ref:Why Is Over-fitting Bad in Machine Learning? - StackOverflow)

Regularization

- In regularization, apart from cost function, weighted sum of parameters and features is added
- For the features which need to be suppressed, their corresponding coefficient is made large. So their cost component increases
- As the Cost needs to be minimized, to reduce these weighted sum, corresponding feature values are reduced,
- So their importances gets lowered dramatically.

Regularization

- Instead of simply aiming to minimize loss (empirical risk minimization): $minimize(Loss(Data|Model))$
- We'll now minimize loss+complexity, which is called structural risk minimization: $minimize(Loss(Data|Model) + complexity(Model))$

Regularization

- Complexity of model can be represented by many ways, here, Model complexity is taken as a function of the weights of all the features in the model.
- Then, a feature weight with a high absolute value is more complex than a feature weight with a low absolute value.

Under-fitting

- Under-fitting: occurs when model cannot fit training data well
- Model does generalize to the overall problem but not that well
- This is also bad! We wish to avoid under-fitting
- Techniques: more features

Bias and Variance

Evaluation Metrics: Prediction Bias

Types of Errors

The prediction error can be broken down into:

- Bias Error
- Variance Error

error(X) = noise(X) + bias(X) + variance(X)

Bias

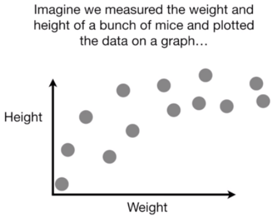
- Bias is the simplifying assumption made by a model to make the target function easier to learn.
- A high bias makes it fast to learn and easier to understand but is generally less flexible. In turn, it has lower predictive performance on complex problems.
- Low Bias: Suggests less assumptions about the form of the target function.
- High-Bias: Suggests more assumptions about the form of the target function.

Variance

- Variance is the amount that the estimate of the target function will change if different training data was used.
- Ideally, it should not change too much from one training dataset to the next, meaning that the algorithm is good at picking out the hidden underlying mapping between the inputs and the output variables.
- Low Variance: Suggests small changes to the estimate of the target function with changes to the training dataset.
- High Variance: Suggests large changes to the estimate of the target function with changes to the training dataset.

Intuition: Example

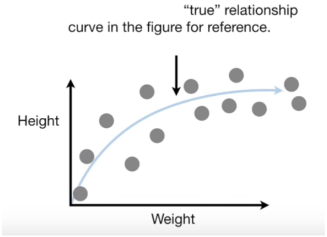
- We have collected past observations of Mice's Heights and Weights.
- Clearly,the relationship is sort of curved.
- Wish to predict Height given Weight.
- Different algorithms will have different underlying structures to fit this regression data.



(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Intuition: True Relationship

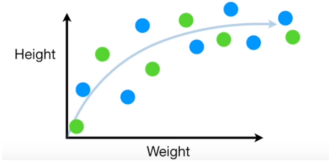
- We have collected past observations of Mice's Heights and Weights.
- Clearly,the relationship is sort of curved.
- Different algorithms will have different underlying structures to fit this regression data.



(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Intuition: Machine Learning

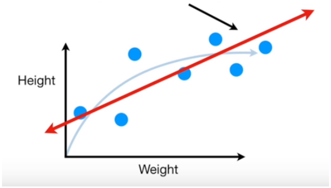
- Split data into training and testing (validation)
- Blue for training
- Green for testing



(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Intuition: Linear Regression

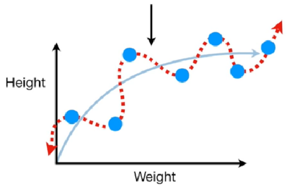
- Fitting Linear Regression model on the training data (blue dots)
- It fits a line, whatever the underlying data is.
- But it finds best line that minimizes Least Square error.
- Anyway, it can not bend, so said to have HIGH bias. It is not very accurate.
- Inability of Machine Learning model to capture the TRUE relationship is called as **Bias**.



(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Intuition: Non-Linear (Tree) Regression

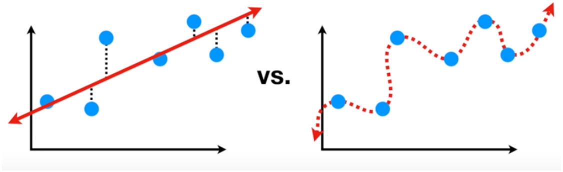
- Some other Machine Learning model can give high degree and flexible model.
- As it passes through points, its very accurate (as there is just no error)
- Being flexible, it has ability to represent true relationship, so very little bias.



(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Intuition: Training set Comparison

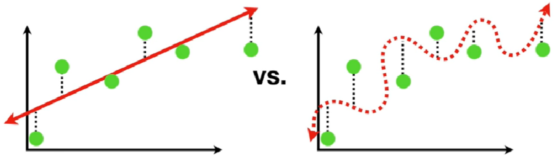
- Linear model has error on Training set. High Bias.
- Non-linear model has almost no error on Training set. Low Bias.



(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Intuition: Testing set Comparison

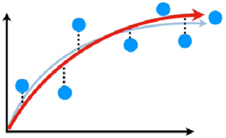
- When same model is applied on the testing set, how does it fair?
- Which model looks better? Linear or Non-Linear?
- Linear is not accurate, but the Non-linear is far too bad.
- This is called as **Variance**.
- Difference in fits on different datasets is the Variance.
- Linear model has less Variance and Non Linear model has High Variance.

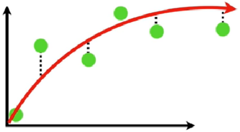


(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Intuition: Ideal

- Linear model gives good predictions but not great predictions, but its consistent.
- Non linear model gives great predictions but not consistent many a times.
- Ideal model captures underlying true relationship. So, low Bias and low Variance.
- It is done by finding a Sweet spot between simple model and complex model.
- 3 such methods of finding sweet spot are : regularization, bagging and boosting.





(Ref: Machine Learning Fundamentals: Bias and Variance -StatQuest with Josh Starmer)

Bias and Variance (Recap)

- Bias: the error introduced by modeling for complex situation by a much simpler model
- **The more flexible (complex) a method is, the less bias it will generally have.**
- Variance: how much the learned model will change if the training set was different
- Does changing a few observations in the training set, dramatically affect the model?
- **Generally, the more flexible (complex) a method is, the more variance it has.**

Learning Method Bias

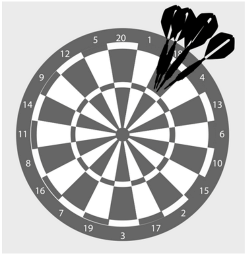
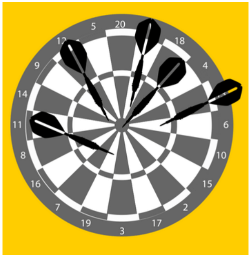
- Bias: the error introduced by modeling a real-life problem (usually extremely complicated) by a much simpler model.
- High bias is a strong view of modeling with simplistic solution.
  - Example: linear regression assumes a linear relationship between the target variable Y and the predictor variables X
  - It's unlikely that the relationship is exactly linear, so some bias will be present in the model
- The more flexible (complex) a method is, the less bias it will generally have.

Learning Method Variance

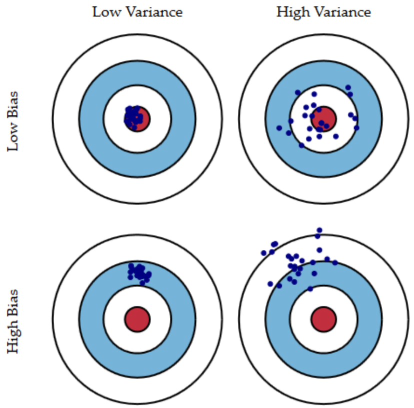
- Variance: how much the learned model will change if the training set was different
  - Does changing a few observations in the training set, dramatically affect the model?
  - Ideally, answer is no.
- Generally, the more flexible (complex) a method is, the more variance it has.

Example

- Imagine you have 5 training datasets of the same problem
- Imagine using an machine learning algo on these sets, so we have 5 models
- High bias algo gives off the mark result
- Low variance gives converged results
- Low bias algo gives towards the mark result
- High variance gives spread results

Bias vs Variations



http://scott.fortmann-roe.com/docs/BiasVariance.html http://scott.fortmann-roe.com/docs/BiasVariance.html

On a Lighter Note (Don't start imagining the individuals)

A person with high bias is someone who starts to answer before you can even finish asking. A person with high variance is someone who can think of all sorts of crazy answers. Combining these gives you different personalities:

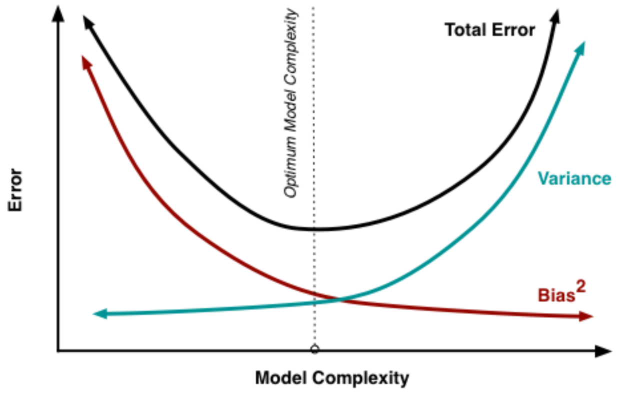
- High bias/low variance: this is someone who usually gives you the same answer (low variance), no matter what you ask, and is usually wrong about it (high bias)
- High bias/high variance: someone who takes wild guesses (high variance), all of which are sort of wrong; (high bias)
- Low bias/high variance: a person who listens to you and tries to answer the best they can (low bias, looks at lots of data), but that daydreams a lot and may say something totally crazy; (high variance)
- Low bias/low variance: a person who listens to you very carefully and gives you good answers pretty much all the time.

Bias Variance Trade-Off

What is the trade-off?

- Generally, as the model becomes more computational (e.g. when the number of free parameters grows), the variance (dispersion) of the estimate also increases, but bias decreases.
- Due to the fact that the training set is memorized completely instead of generalizing, small changes lead to unexpected results (over-fitting).
- On the other side, if the model is too weak, it will not be able to learn the pattern, resulting in learning something different that is offset with respect to the right solution.

What is the trade-off?



But Why is there a trade-off?

- Low bias algos tend to be more complex and flexible e.g. Decision Tree (non-linear) so that it can fit the data well.
- But if the algo is too flexible, it will fit each training data set differently, and hence have high variance.
- A key characteristic of many supervised learning methods is a built-in way to control the bias-variance trade-off either automatically or by providing a special parameter that the data scientist can adjust.

Summary

- Statistical learning reveals hidden data relationships.
- Relationships between dependent and independent data.

Summary

- Model is the transformation engine.
- Parameters are the ingredients that enable the transformation.
- A model uses the training data to learn.
- A model uses the testing data to evaluate.

Summary

- Bias-variance trade-off is a balancing act.
- Balance to find optimal model.
- Balance to find the sweet spot.

Finally

All models are wrong; but some are useful - George E P Box

Thanks ...

- Search **"Yogesh Haribhau Kulkarni"** on Google and follow me on LinkedIn and Medium
- Office Hours: Saturdays, 2 to 5pm (IST); Free-Open to all; email for appointment.
- Email: yogeshkulkarni at yahoo dot com