twiml

AI Enterprise Workflow Study Group

Course 4, Week 1

4/18/2020

Agenda

- Check in
- Discussion
- Next steps

Course & Study Group Schedule

AI Enterprise Workflow Study Group		
Session	**Topic**	**Date**
Overview Webinar	Webinar with instructor, Ray Lopez	15-Feb
Course 1 Week 1	Course intro	22-Feb
Course 1 Week 2	Data ingestion, cleaning, parsing, assembly	29-Feb
Course 2 Week 1	Exploratory data analysis & visualization	7-Mar
Course 2 Week 2	Estimation and NHT	14-Mar
Course 3 Week 1	Data transformation and feature engineering	21-Mar
Course 3 Week 2	Pattern recognition and data mining best practices	28-Mar
Course 4 Week 1	Model evaluation and performance metrics	18-Apr
Course 4 Week 2	Building machine learning and deep learning models	25-Apr
Course 5 Week 1	Deploying models	2-May
Course 5 Week 2	Deploying models using Spark	9-May
Course 6 Week 1	Feedback loops and monitoring	16-May
Course 6 Week 2	Hands on with OpenScale and Kubernetes	23-May
Course 6 Week 3	Captsone project week 1	30-May
Course 6 Week 4	Captsone project week 2	6-Jun

Course 4 Week 1 learning objectives

1. Discuss common regression, classification, and multilabel classification metrics
2. Describe common strategies for grid searching and cross-validation
3. Explain the use of linear models in supervised learning applications
4. Create and test an instance of Watson Natural Language Understanding
5. Employ evaluation metrics to select models for production use

Regression Metrics

- RMSE $\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2}$
- MAE $\text{MAE} = \frac{1}{N}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|$

`sklearn.metrics.mean_squared_error`

`sklearn.metrics.mean_absolute_error`

RMSE vs MAE

- Both same units as data
- RMSE penalizes large errors
- RMSE differentiable

CASE 1: Evenly distributed errors

ID	Error	|Error|	Error^2
1	2	2	4
2	2	2	4
3	2	2	4
4	2	2	4
5	2	2	4
6	2	2	4
7	2	2	4
8	2	2	4
9	2	2	4
10	2	2	4

MAE	RMSE
2.000	2.000

CASE 2: Small variance in errors

ID	Error	|Error|	Error^2
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	3	3	9
7	3	3	9
8	3	3	9
9	3	3	9
10	3	3	9

MAE	RMSE
2.000	2.236

CASE 3: Large error outlier

ID	Error	|Error|	Error^2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	20	20	400

MAE	RMSE
2.000	6.325

Classifier Metrics

Recall from C3W1:

		True condition				
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$		

F_β Score

- Combines precision and recall into one metric

  - Precision = $\frac{TP}{TP+FP}$: proportion called true that are correct
  - Recall = $\frac{TP}{TP+FN}$: proportion of true that are called correctly

  The F1_score is the harmonic mean of precision and recall.

  $$\text{F1\_score} = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}$$

  The F1_score is actually a special case of the F_β score, where the weight of recall and precision is evenly balanced. The F-score can also be written as:

  $$F_\beta = (1+\beta^2)\frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}$$

- β essentially a multiplier on importance of recall

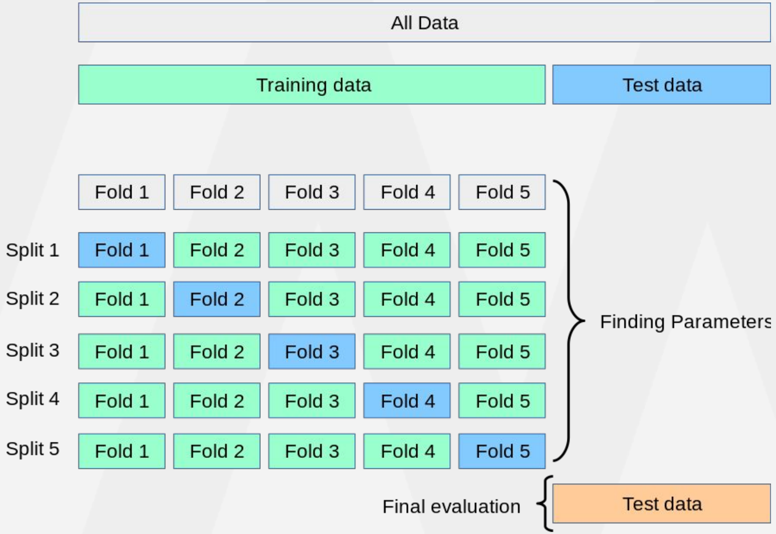
Model Validation

K-Fold Validation

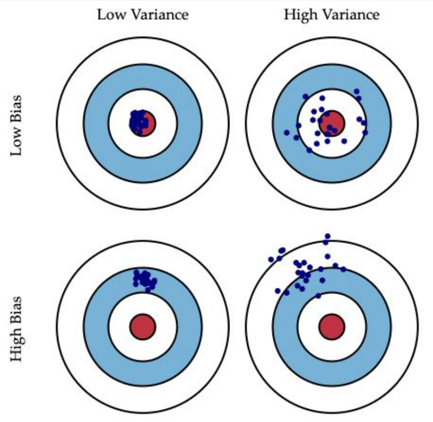
- Shuffle & Split: Shuffles before split
- Stratified: Balance classes across folds

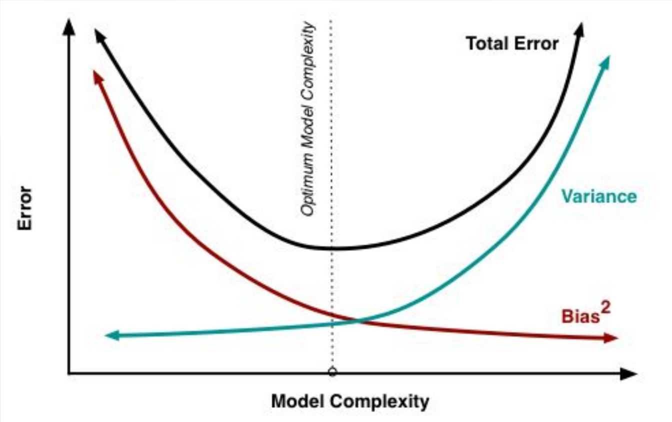
Parameter Tuning

- Grid, random search
- Also Bayesian methods



Bias vs Variance





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

Linear Models

Linear Models in Python

- `LinearRegression` (sklearn.linear_model.LinearRegression)
- `LogisticRegression` (sklearn.linear_model.LogisticRegression)
- `RidgeCV` (sklearn.linear_model.RidgeCV)
- `LassoCV` (sklearn.linear_model.LassoCV)
- `MultiTaskLasso` (sklearn.linear_model.MultiTaskLasso)
- `ElasticNetCV` (sklearn.linear_model.ElasticNetCV)
- `BayesianRidge` (sklearn.linear_model.BayesianRidge)
- `statsmodels` (for GLMs and situations where inference is the goal)
- `PyMC3` (for GLMMs)

In [19]:

```
from sklearn import linear_model
slide_print([lm for lm in dir(linear_model) if not re.search("_",lm)])
```

```
['ARDRegression', 'BayesianRidge', 'ElasticNet', 'ElasticNetCV', 'Hinge', 'Huber', 'HuberRegressor', 'Lars', 'LarsCV', 'Lass
o', 'LassoCV', 'LassoLars', 'LassoLarsCV', 'LassoLarsIC', 'LinearRegression', 'Log', 'LogisticRegression', 'LogisticRegression
CV', 'ModifiedHuber', 'MultiTaskElasticNet', 'MultiTaskElasticNetCV', 'MultiTaskLasso', 'MultiTaskLassoCV', 'OrthogonalMatchin
gPursuit', 'OrthogonalMatchingPursuitCV', 'PassiveAggressiveClassifier', 'PassiveAggressiveRegressor', 'Perceptron', 'RANSACRe
gressor', 'Ridge', 'RidgeCV', 'RidgeClassifier', 'RidgeClassifierCV', 'SGDClassifier', 'SGDRegressor', 'SquaredLoss', 'TheilSe
nRegressor', 'base', 'bayes', 'huber', 'logistic', 'omp', 'perceptron', 'ransac', 'ridge', 'sag']
```

Play

Additional Discussion

What did you learn?

What stumbling blocks did you run into?

How do these lessons relate to your experience?

What did you learn/find interesting in this week’s lesson?

What are you doing as homework?

What interesting resources have you found?

Other?