RocketScience.ai

TENSOR.BY

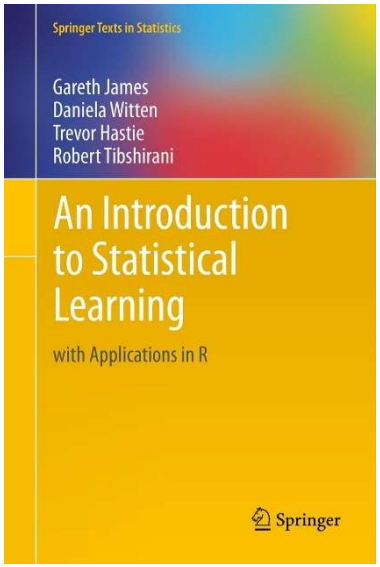
ML-course

3. Classification in Python Scikit-learn

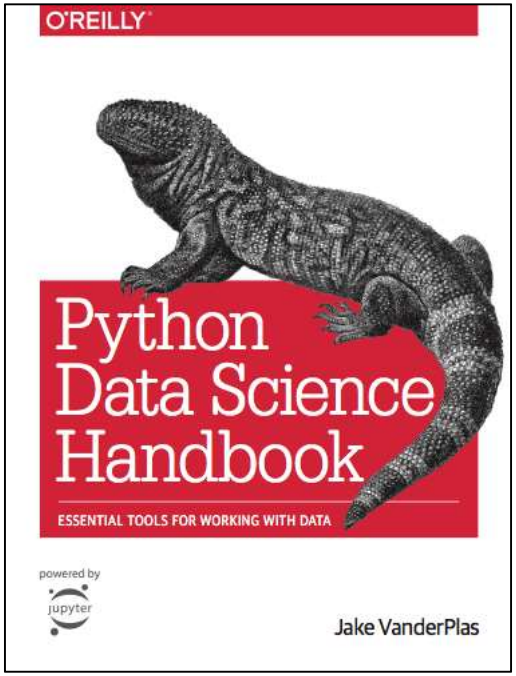
Kate Miniukovich (Data Scientist),
miniukovich@rocketscience.ai

Reference

An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, http://www-bcf.usc.edu/~gareth/ISL/ (available online for free)



Reference



Jake VanderPlas

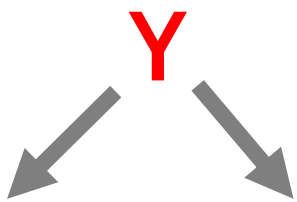
Python Data Science Handbook
https://jakevdp.github.io/PythonDataScienceHandbook/

Video
https://www.youtube.com/watch?v=L7R4HUQ-eQ0&t=6033s

Reference

http://scikit-learn.org

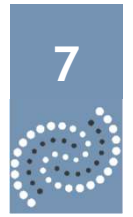
Supervised vs. Unsupervised Learning

Supervised	Unsupervised
Data: 1) n observations; 2) p variables X1, X2, . . .,Xp, measured on each observation; 3) response Y measured on same n observations  Continuous Regression Discrete Classification	Data: 1) n observations; 2) p variables X1, X2, . . .,Xp, measured on each observation Clustering…

Classification

Binary	**Multiclass or multinomial**
2 classes	more than 2 classes

Regression / Classification Problem



Steps to solve

- ***Working with data***
- ***Modeling***

Working with data

- Tidy data
- Types of variables and actions
- Missing data and imputation
- Feature engineering
- Data preprocessing for scikit-learn

Working with data Tidy Data

- Tidy data is a standard way of mapping the meaning of a dataset to its structure. This is Codd's 3rd normal form and the focus put on a single dataset rather than the many connected datasets common in relational databases.
- In tidy data:
  1. **Each variable forms a column.**
  2. **Each observation forms a row.**
  3. **Each type of observational unit forms a table.**

Which table below is tidy?

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

person	treatment	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

More about tidy data:

Original, code in R ftp://cran.r-project.org/pub/R/web/packages/tidyr/vignettes/tidy-data.html

Code in Python https://www.ibm.com/developerworks/community/blogs/jfp/entry/Tidy_Data_In_Python?lang=en

Working with data

Types of variables and actions

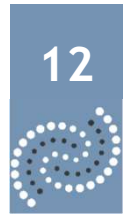
Types of variables	Actions
Categorical	Convert to n binary vars (n – number of labels)
Text	*Options:* • *Scrap a pattern and convert it to n binary vars* • *Convert text to numbers (Word2Vec)* • *Drop text variable*
Numerical	*Standardization of datasets is a common requirement for many machine learning estimators implemented in scikit-learn* *http://scikit-learn.org/stable/modules/preprocessing.html* *Standardization =* *= mean removal + variance scaling*

Working with data

Missing data and imputation

- **Missing data:** NaN
- **Imputation**
  - Mean, median or mode
  - Prediction

Examples:
https://www.kaggle.com/kernels search on “Missing data imputation”



Feature Engineering

- Based on variables meaning
- Technical approaches

Examples:

https://www.kaggle.com/kernels search on “Feature engineering”

Working with data

Representation of in Scikit-learn

- **X**

  *two-dimensional **numpy array***
  shape – (n_samples, m_features)

- **Y**

  one-dimensional numpy array
  shape – (n_samples,)

Working with data

Example

- dataset: Titanic
  https://www.kaggle.com/c/titanic

- classification_titanic_simple.ipynb

Modeling

- Choose a class of model
- Fit the model to data
- Validate the model and optimize hyperparameters
- Predict for unknown data

Some models for binary classification in Python scikit-learn

- **Generalized Linear Models**
  - Logistic regression

example in classification_titanic_simple.ipynb

- **Ensemble methods**
  - Random Forests
  - Gradient Tree Boosting

Binary Classification

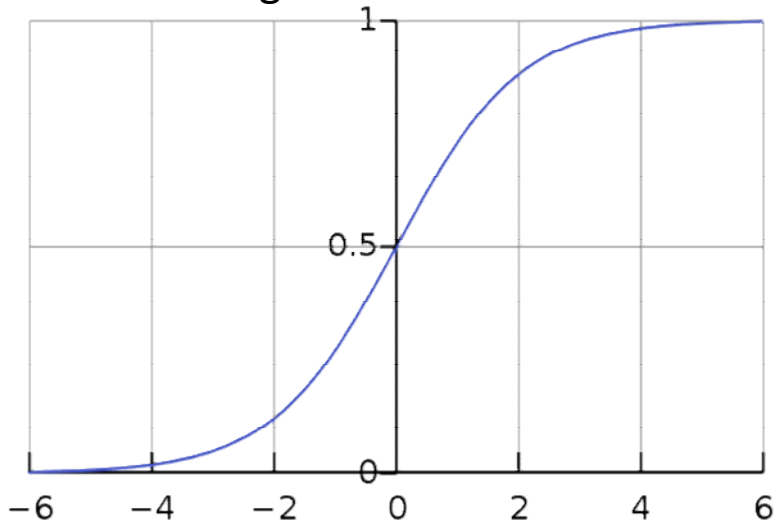
Logistic Regression

To model p(X) = Pr(Y = 1|X) we need function that gives outputs between 0 and 1 for all values of X

$$\hat{y} = p(X) = \frac{e^{\theta_0+\theta_1 x_1+\cdots+\theta_m x_m}}{1+e^{\theta_0+\theta_1 x_1+\cdots+\theta_m x_m}} = \frac{e^{X\theta}}{1+e^{X\theta}}$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = X\theta$$

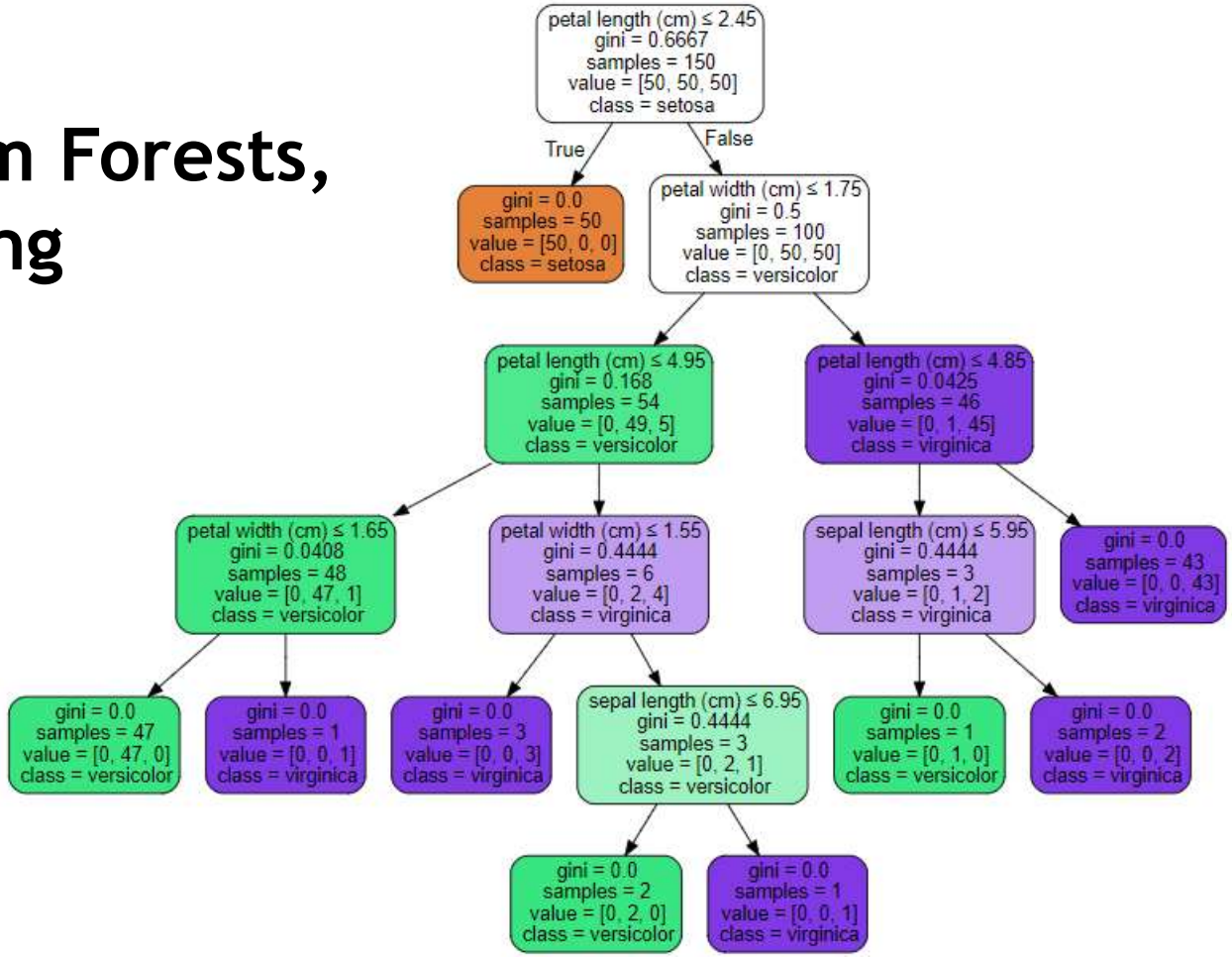
Logistic function



$$f(x) = \frac{e^x}{1+e^x}$$

Binary Classification

Decision trees
Bagging, Random Forests,
Gradient Boosting

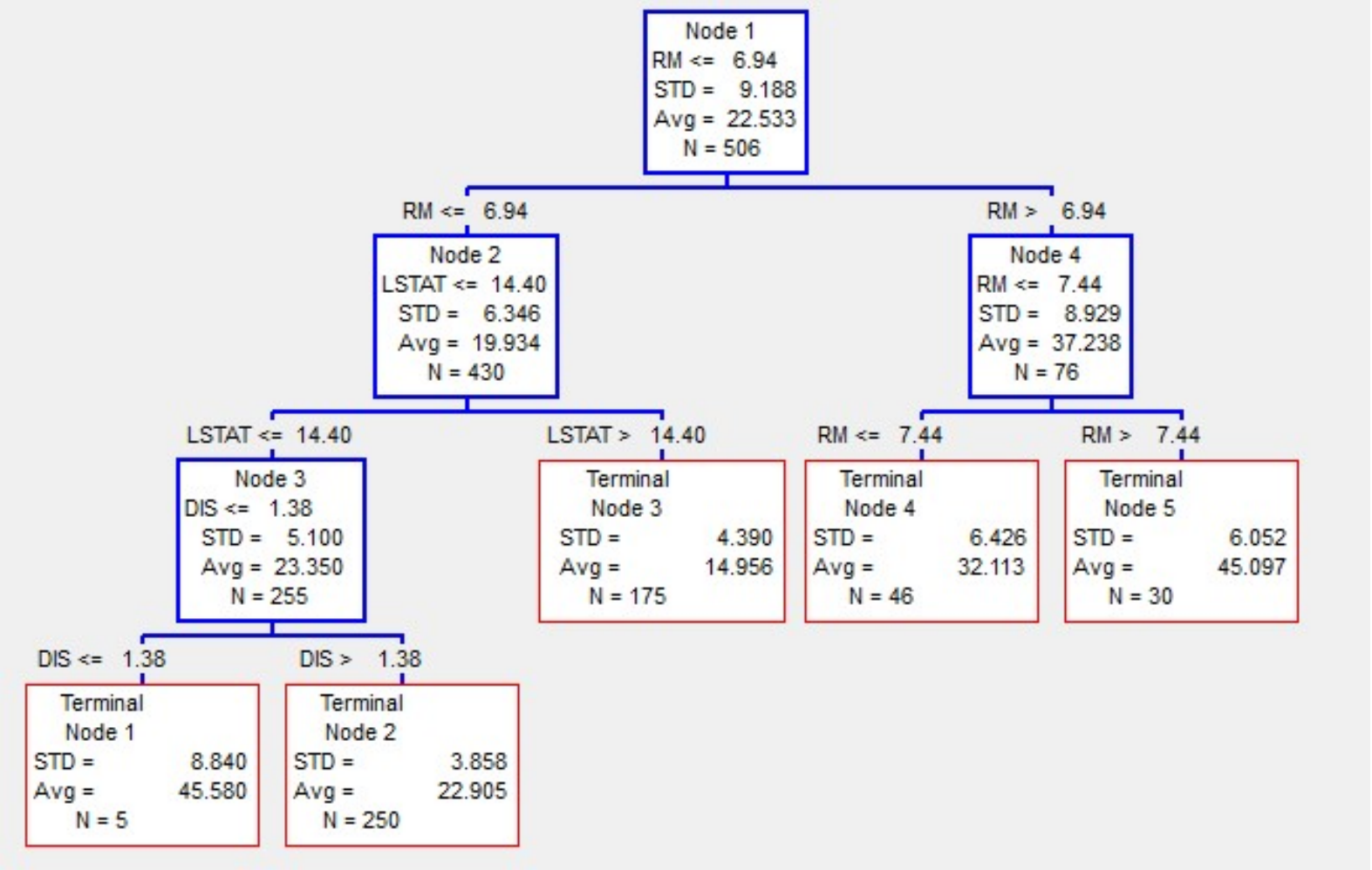


Good explanation of Boosted Trees
http://xgboost.readthedocs.io/en/latest///model.html

Regression

Decision trees

Bagging, Random Forests,Gradient Boosting



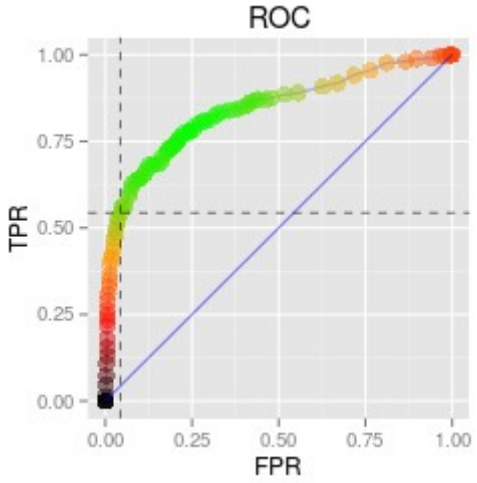
Classification metrics

Confusion matrix

Survived (S) -1; Not Survived (NS) - 0

Actual/Predicted	*0*	*1*	*Error*
0 (N)	**TN** (NS as NS)	**FP** (NS as S)	**FPR=FP/N** (False Positive Rate)
1 (P)	**FN** (S as NS)	**TP** (S as S)	**FNR=FN/P**(False Negative Rate)

Receiver operating characteristic curve



Accuracy = (TP+TN)/(P+N)

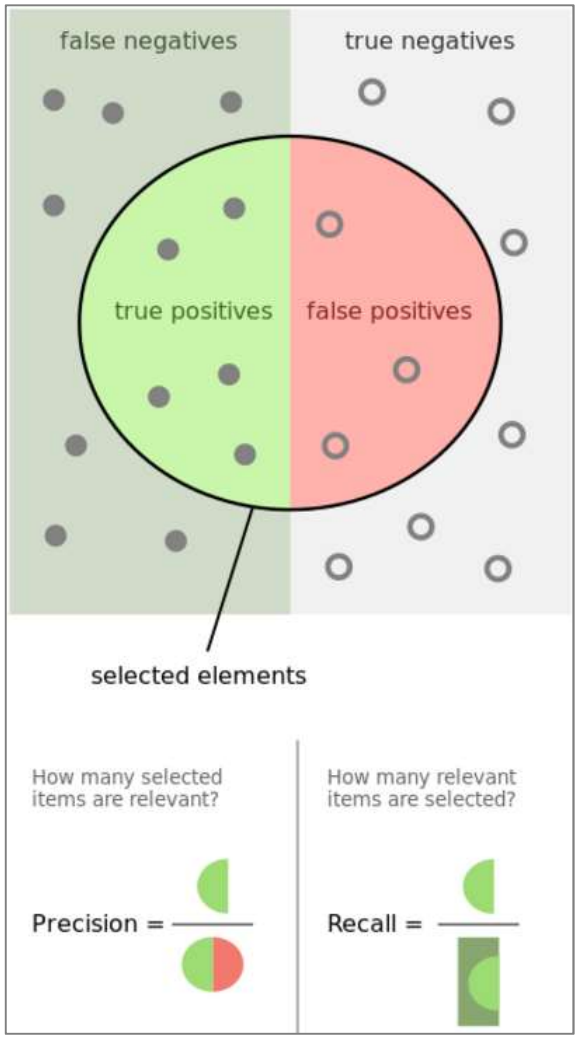
Precision = TP/(TP+FP) **Recall** = **TPR** = TP/P

F1 = 2*Precision*Recall/(Precision+Recall) - *harmonic mean Precision and Recall*

AUC - Area Under ROC Curve (**the closer to 1, the better a model is**)

More: https://en.wikipedia.org/wiki/Precision_and_recall

Classification metrics



Confusion matrix

Survived (S) -1; Not Survived (NS) - 0

Actual/Predicted	***0***	***1***	***Error***
0 (N=438)	TN=365	FP=?	FPR=FP/N = ?
1 (P=274)	FN=?	TP=212	FNR=FN/P = ?
Total			(FN+FP)/(N+P) = ?

FN - ошибка первого рода; FP - ошибка второго рода

Accuracy = (TP+TN)/(P+N) - ?

Precision = TP/(TP+FP) -?

Recall = **TPR** = TP/P-?

http://scikit-learn.org/stable/modules/classes.html#classification-metrics

Binary Classification

Example: binary models for Titanic dataset

Models comparison based on Accuracy

Model	Train	CV	Test
LgR	*0.82*	*0.81*	*0.77*
RF			
GB			

Multiclass Classification

Some classification algorithms naturally permit the use of more than two classes

- **Logistic Regression**
- **Random Forests, Gradient Boosting**

example in mclass_classification.ipynb

Techniques of transformation to binary

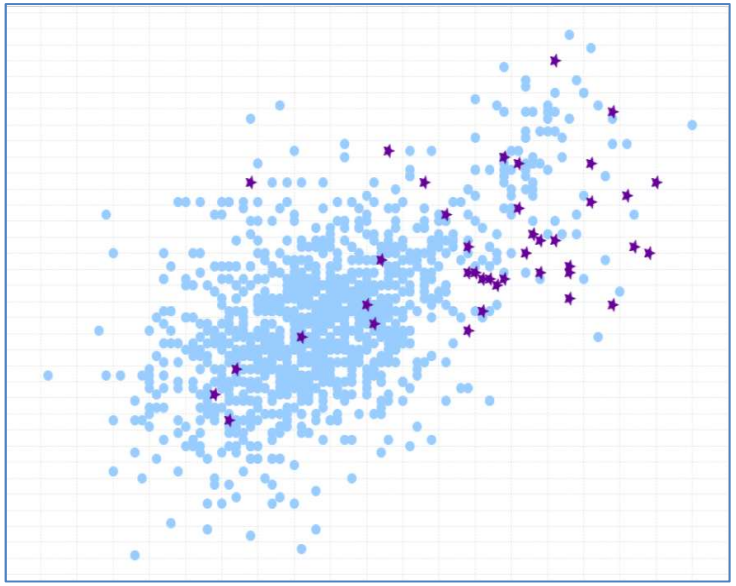
- One vs. All
- One vs. One

Read more:
https://en.wikipedia.org/wiki/Multiclass_classification
http://scikit-learn.org/stable/modules/multiclass.html

Classification: Unbalanced classes

Unbalanced classes - classes are not represented equally

Accuracy Paradox



Tactics to Combat Unbalanced Classes

1) Collect more data
2) Resample Your Dataset
3) Generate Synthetic Samples
4) Change Your Performance Metric
5) Use special hyperparameters
(e.g. *class_weight* in *sklearn.ensemble.RandomForestClassifier*)

Read more: 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset
http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Modeling

Hyperparameters optimization

- Parameters to optimize
- Good range of values

More about parameters to optimize and good range of values
https://www.linkedin.com/pulse/approaching-almost-any-machine-learning-problem-abhishek-thakur?trk=hp-feed-article-title-like

Q & A

Thank you!