

Master Machine Learning with Python

by
Ted Petrou

© 2019 Ted Petrou All Rights Reserved

Contents

I	Environment Setup and Jupyter Notebooks	1
1	Installing Python and Setting up an Environment for Data Science	3
1.1	Conda	3
1.2	Miniconda	3
1.3	Python and Conda installation complete	6
1.4	Creating a new environment just for data analysis	8
1.5	Create a new environment	9
1.6	Other Considerations	11
1.7	Summary of Steps	13
1.8	Using the book with Jupyter Notebooks	13
2	Introduction to Jupyter Notebooks	15
2.1	Jupyter Notebook Basics	15
2.2	Getting Started with Jupyter Notebooks	16
2.3	Executing Cells	17
2.4	Keyboard Shortcuts	18
2.5	Completing exercises in this book	20
2.6	Getting help in the notebook	20
3	Markdown Tutorial	23
3.1	Headers	23
3.2	Italics and bold	23
3.3	Code formatting	23
3.4	Lists	24
3.5	Hyperlinks	24
3.6	Images	25
3.7	New lines	25
4	Jupyter Notebooks More	27
4.1	Where Jupyter Notebooks Excel	27
4.2	Where Jupyter Notebooks Fail	27
4.3	On-demand Jupyter Notebooks	28
5	Jupyter Notebook Extensions	29
5.1	The NBextensions tab	29
5.2	The Skip-Traceback Extension	30
5.3	Skip-Traceback in the Notebook	30
II	The Machine Learning Model	33
6	1. Exploratory Data Analysis	35
6.1	Introduction	35
6.2	Introducing the Ames, Iowa Housing Dataset	35
6.3	The data dictionary	36
6.4	Automated EDA with the Pandas Profiling Package	38

7	2. The Machine Learning Model	39
7.1	What is a model?	39
7.2	Machine learning models	39
7.3	All of these models learn from data	40
7.4	Plotting a range of predictions	41
8	3. Learning vs Machine Learning	43
8.1	What is Learning?	43
8.2	What is Machine Learning?	43
8.3	The two types of machine learning	44
8.4	Terminology	44
8.5	Assessing task performance	45
8.6	Make prediction and calculate error for each observation	45
8.7	Comparing model performance against one another	48
	III Linear Regression	51
9	4. Linear Regression	53
9.1	Introduction	53
9.2	The Linear Regression Model	53
9.3	Learn the coefficients from the data	53
9.4	Propose search area for w_0 and w_1	55
9.5	Plot the model along with the actual data	56
9.6	Absolute vs Relative measures of performance	57
9.7	A slightly different error metric	57
9.8	Visualizing R-Squared	57
10	5. Linear Regression Widget	61
11	6. Linear Regression with scikit-learn	63
11.1	Introduction	63
11.2	The three step process to train a machine learning model in scikit-learn	63
11.3	Import, Instantiate, Train - The three-step process	63
11.4	Trained model	67
11.5	What happens during <code>fit</code> ?	67
11.6	Types of features	67
11.7	Exercises	67
12	7. Prediction and Performance Evaluation	69
12.1	Repeating the three step machine learning process	69
12.2	Write our own function to calculate predictions	70
12.3	Evaluating the performance of our predictions	70
12.4	Exercises	71
13	8. Multiple Linear Regression	73
13.1	Introduction	73
13.2	Same Goal - Minimize squared error	73
13.3	Choose two variables to build a model	73
13.4	Import, Instantiate, Train	75
13.5	Make predictions	75
13.6	We can still be extrapolating	76
13.7	Evaluating model performance	78
13.8	Comparing our multivariate model to the single variable case	78
13.9	Exercises	79
14	9. Establishing a Baseline	81
14.1	How to establish a baseline?	82
14.2	A baseline is built into linear regression	82
14.3	The <code>dummy</code> module	82

14.4	Slowly build more complex models	83
14.5	Simple Linear Regression Model Interpretation	83
14.6	Multiple Linear Regression Model Interpretation	83
14.7	No statistical output with scikit-learn	83
14.8	Machine Learning vs Statistics	83
14.9	Exercises	83
15	10. What it means to be a linear model	85
15.1	Introduction	85
15.2	Linear regression is more flexible than a straight line	85
15.3	Create new input data	87
15.4	Feature Engineering	88
15.5	Exercises	88
IV	More Supervised Learning Models	89
16	11. K-Nearest Neighbors	91
16.1	Introduction	91
16.2	How KNN works	91
16.3	KNN in Pandas	91
16.4	KNN with multiple features	93
16.5	Create a function to do KNN with plots	94
16.6	Use Scikit-Learn	95
16.7	Measuring Performance of KNN	96
16.8	Notes on KNN	96
16.9	Using proportional distance	96
16.10	Distance Calculation on Variables of different scale	97
16.11	The fewer the neighbors the higher the variance	97
16.12	Exercises	97
17	12. Decision Trees	99
17.1	Introduction	99
17.2	How a decision tree is created	100
17.3	Manually create a short tree	100
17.4	Step 3: Split on FullBath	102
17.5	Step 4: Repeat until stopping criterion is met	103
17.6	Use scikit-learn to create a decision tree	105
17.7	Visualize it with graphviz	106
17.8	Exercises	107
18	13. Random Forests	109
18.1	Introduction	109
18.2	Random forests are a collection of decision trees	109
18.3	Bagging = Bootstrap Aggregating	109
18.4	Random Forest in Scikit-Learn	110
18.5	Visualize each tree with graphviz	111
18.6	Random forests build weak learners, why are they good?	113
18.7	Evaluate the random forest	113
18.8	Extremely random forests	113
18.9	Exercises	113
V	Model Evaluation and Selection	115
19	14. Evaluating Model Performance	117
19.1	Introduction	117
19.2	Goal: Estimate Performance on Future Unseen Data	117
19.3	First idea, create a “holdout” test set that is not used during training	117
19.4	Fit model just on the training data	118

19.5	Evaluating (incorrectly) on the training data	119
19.6	Evaluate on the test data	119
19.7	Next Idea: Cross Validation	119
19.8	Many flavors of cross validation	119
19.9	K-Fold Cross Validation in Scikit-Learn	120
19.10	No model is returned	120
19.11	Other flavors of cross validation	121
19.12	Shuffling the data	121
19.13	Exercise	121
20	15. Evaluation Metrics	123
20.1	Introduction	123
20.2	With Cross Validation	124
20.3	Exercises	126
21	16. Hyperparameter Tuning	127
21.1	Hyperparameters vs Parameters	127
21.2	Overfitting	127
21.3	Inspecting the decision tree	128
21.4	Change hyperparameters to reduce overfitting	129
21.5	Exercises	129
22	17. Model Selection	131
22.1	Automating Hyperparameter Tuning	131
22.2	Execute the exhaustive grid search	132
22.3	Get results from our grid search object	132
22.4	Getting the best trained estimator back	133
22.5	More	134
22.6	Exercises	135
23	18. Regularization	137
23.1	Overfitting and Underfitting	137
23.2	Flexible vs Inflexible models	138
23.3	The Bias/Variance Tradeoff	138
23.4	Linear Regression with all numeric variables	138
23.5	Not much overfitting	139
23.6	Regularization - Penalized Regression	140
23.7	Regularization in linear regression	140
23.8	Ridge and Lasso Regression	140
23.9	Must Scale Data Before Regularization	140
23.10	Standardizing the data	140
23.11	Transforming data in scikit-learn	140
23.12	Refit the data with the scaled data	142
23.13	Parameter values are huge - Transform Sale Price	143
23.14	Ridge Regression in scikit-learn	145
23.15	Repeat with Lasso	149
23.16	Plot the coefficients with respect to α	152
VI	Data Transformations	153
24	19. Missing Values	155
24.1	Imputation in scikit-learn	156
24.2	Find columns with missing values	156
25	20. Handling Categorical Data	159
25.1	Encoding	160
25.2	One-hot encoding in scikit-learn	160
25.3	That's a lot of features - what happened?	161
25.4	Let's use only the string columns	161

25.5	Apply separate transformations to different parts of the data	162
25.6	Scale the numeric variables	163
25.7	Exercises	164
VII	The Machine Learning Pipeline	165
26	21. Building a Pipeline	167
26.1	Introduction	167
26.2	What is a scikit-learn pipeline?	167
26.3	Pipeline within a ColumnTransformer within a Pipeline	170
27	22. Model Persistence	173
28	23. Loading a Model	177
VIII	Feature Engineering	179
29	24. Feature Engineering	181
29.1	Introduction	181
29.2	Use KBinsDiscretizer to create the bins	182
29.3	Handling columns dominated by a single string	183
29.4	Handling low-frequency strings	185
IX	Kaggle Competitions	187
30	25. Kaggle Competitions	189
30.1	Introduction	189
X	Classification	191
31	26. Intro to Classification	193
31.1	Introduction	193
31.2	Heart Disease Dataset	193
31.3	Verifying the type of problem we have	194
31.4	Identifying the target variable	195
31.5	Classification or Regression	195
31.6	Different models for classification	195
31.7	Begin by using a single feature	195
31.8	Use the max_hr column as feature	195
31.9	Import, Instantiate, Train	195
31.10	Make a prediction	196
31.11	Interpretation of prediction	196
31.12	Make a prediction for all inputs	196
31.13	Measure performance by calculating the accuracy	196
32	27. Logistic Regression	199
32.1	Redo our logistic regression output	199
32.2	Attributes and methods of our Estimator	199
32.3	Making a prediction for all possible values	200
32.4	Logistic Regression returns a probability	201
32.5	View the underlying probabilities with predict_proba	201
32.6	Isn't logistic regression supposed to draw an S curve?	202
32.7	The parameter values of the logistic regression model	203
32.8		

$$y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

.....	203
32.9 Calculating the probabilities manually	203
32.10 Test out our function	204
32.11 Verify the output is the same	204
33 28. Establishing a Baseline for Classification	205
33.1 Taking a step back	205
33.2 The simplest classification model - Guessing the most common class	206
33.3 Establish the baseline before doing machine learning	206
33.4 Using the simplest model as a baseline	206
33.5 Our logistic regression vs the baseline	206
33.6 Building a Dummy estimator - a baseline model in scikit-learn	207
34 29. Practicing Classification	209