[1. Approximate\_nearest\_neighbors 12](#_Toc103004248)

[2. Brain\_tumor\_detection 12](#_Toc103004249)

[3. Breast\_cancer\_prediction 12](#_Toc103004250)

[4. Chess\_piece\_detection 12](#_Toc103004251)

[5. Eda\_for\_pubg 12](#_Toc103004252)

[6. Grid\_search\_text\_feature\_extraction 12](#_Toc103004253)

[7. House\_price\_prediction 12](#_Toc103004254)

[8. Mask\_or\_no\_mask 12](#_Toc103004255)

[9. Open\_images\_eda 12](#_Toc103004256)

[10. Plot\_adaboost\_hastie\_10\_2 12](#_Toc103004257)

[11. Plot\_adaboost\_multiclass 12](#_Toc103004258)

[12. Plot\_adaboost\_regression 12](#_Toc103004259)

[13. Plot\_adaboost\_twoclass 12](#_Toc103004260)

[14. Plot\_adjusted\_for\_chance\_measures 12](#_Toc103004261)

[15. Plot\_affinity\_propagation 12](#_Toc103004262)

[16. Plot\_agglomerative\_clustering 12](#_Toc103004263)

[17. Plot\_agglomerative\_clustering\_metrics 12](#_Toc103004264)

[18. Plot\_agglormerarive\_dendrogram 12](#_Toc103004265)

[19. Plot\_anomaly\_comparison 12](#_Toc103004266)

[20. Plot\_ard 12](#_Toc103004267)

[21. Plot\_bayesian\_ridge 12](#_Toc103004268)

[22. Plot\_bayesian\_ridge\_curvefit 12](#_Toc103004269)

[23. Plot\_beta\_divergence 12](#_Toc103004270)

[24. Plot\_bias\_variance 12](#_Toc103004271)

[25. Plot\_bicluster\_newsgroups 12](#_Toc103004272)

[26. Plot\_birch\_vs\_minibatchkmeans 12](#_Toc103004273)

[27. Plot\_caching\_nearest\_neighbors 13](#_Toc103004274)

[28. Plot\_calibration 13](#_Toc103004275)

[29. Plot\_calibration\_curve 13](#_Toc103004276)

[30. Plot\_calibration\_multiclass 13](#_Toc103004277)

[31. Plot\_changed\_only\_pprint\_parameter 13](#_Toc103004278)

[32. Plot\_classification 13](#_Toc103004279)

[33. Plot\_classification\_probability 13](#_Toc103004280)

[34. Plot\_classifier\_chain\_yeast 13](#_Toc103004281)

[35. Plot\_classifier\_comparison 13](#_Toc103004282)

[36. Plot\_cluster\_comparison 13](#_Toc103004283)

[37. Plot\_cluser\_iris 13](#_Toc103004284)

[38. Plot\_coin\_segmentation 13](#_Toc103004285)

[39. Plot\_coin\_ward\_segmentation 13](#_Toc103004286)

[40. Plot\_color\_quantization 13](#_Toc103004287)

[41. Plot\_column\_transformer 13](#_Toc103004288)

[42. Plot\_column\_transformer\_mixed\_types 13](#_Toc103004289)

[43. Plot\_compare\_colibration 13](#_Toc103004290)

[44. Plot\_compare\_cross\_decomposition 13](#_Toc103004291)

[45. Plot\_compare\_krr 13](#_Toc103004292)

[46. Plot\_compare\_methods 13](#_Toc103004293)

[47. Plot\_compare\_reduction 13](#_Toc103004294)

[48. Plot\_concentration\_prior 13](#_Toc103004295)

[49. Plot\_cost\_complexity\_pruning 13](#_Toc103004296)

[50. Plot\_covariance\_estimation 13](#_Toc103004297)

[51. Plot\_cv\_diabetes 13](#_Toc103004298)

[52. Plot\_cv\_digits 13](#_Toc103004299)

[53. Plot\_cv\_indices 14](#_Toc103004300)

[54. Plot\_cv\_predict 14](#_Toc103004301)

[55. Plot\_cyclical\_feature\_engineering 14](#_Toc103004302)

[56. Plot\_dbscan 14](#_Toc103004303)

[57. Plot\_det 14](#_Toc103004304)

[58. Plot\_dict\_face\_patches 14](#_Toc103004305)

[59. Plot\_digits\_agglomeration 14](#_Toc103004306)

[60. Plot\_digits\_classification 14](#_Toc103004307)

[61. Plot\_digits\_classficiation\_exercies 14](#_Toc103004308)

[62. Plot\_digits\_denoising 14](#_Toc103004309)

[63. Plot\_digits\_kde\_sampling 14](#_Toc103004310)

[64. Plot\_digits\_last\_image 14](#_Toc103004311)

[65. Plot\_digits\_linkage 14](#_Toc103004312)

[66. Plot\_digits\_pipe 14](#_Toc103004313)

[67. Plot\_discretization 14](#_Toc103004314)

[68. Plit\_discretization\_classification 14](#_Toc103004315)

[69. Plot\_descretization\_strategies 14](#_Toc103004316)

[70. Plot\_display\_object\_visualization 14](#_Toc103004317)

[71. Plot\_document\_classification\_20newsgroups 14](#_Toc103004318)

[72. Plot\_document\_clustering 14](#_Toc103004319)

[73. Plot\_elastic\_net\_precomputed\_gram\_matrix\_with\_weighted\_samples 14](#_Toc103004320)

[74. Plot\_ensemble\_oob 14](#_Toc103004321)

[75. Plot\_f\_test\_vs\_mi 14](#_Toc103004322)

[76. Plot\_face\_compress 14](#_Toc103004323)

[77. Plot\_face\_recognition 14](#_Toc103004324)

[78. Plot\_faces\_decomposition 14](#_Toc103004325)

[79. Plot\_feature\_agglomeration\_vs\_univariate\_selection 15](#_Toc103004326)

[80. Plot\_feature\_selection 15](#_Toc103004327)

[81. Plot\_feature\_selection\_pipeline 15](#_Toc103004328)

[82. Plot\_feature\_transformation 15](#_Toc103004329)

[83. Plot\_feature\_union 15](#_Toc103004330)

[84. Plot\_forest\_importances 15](#_Toc103004331)

[85. Plot\_forest\_importances\_faces 15](#_Toc103004332)

[86. Plot\_forest\_iris 15](#_Toc103004333)

[87. Plot\_gmm 15](#_Toc103004334)

[88. Plot\_gmm\_covariances 15](#_Toc103004335)

[89. Plot\_gmm\_pdf 15](#_Toc103004336)

[90. Plot\_gmm\_selection 15](#_Toc103004337)

[91. Plot\_gmm\_sin 15](#_Toc103004338)

[92. Plot\_gpc 15](#_Toc103004339)

[93. Plot\_gpc\_iris 15](#_Toc103004340)

[94. Plot\_gpc\_isoprobability 15](#_Toc103004341)

[95. Plot\_gpc\_xor 15](#_Toc103004342)

[96. Plot\_gpr\_co2 15](#_Toc103004343)

[97. Plot\_gpr\_noisy 15](#_Toc103004344)

[98. Plot\_gpr\_noisy\_targets 15](#_Toc103004345)

[99. Plot\_gpr\_on\_structured\_data 15](#_Toc103004346)

[100. Plot\_gpr\_prior\_psterior 15](#_Toc103004347)

[101. Plot\_gradient\_boosting\_categorical 15](#_Toc103004348)

[102. Plot\_gradient\_boosting\_early\_stopping 15](#_Toc103004349)

[103. Plot\_gradient\_boosting\_oob 15](#_Toc103004350)

[104. Plot\_gradient\_boosting\_quantile 15](#_Toc103004351)

[105. Plot\_gradient\_boosting\_regression 16](#_Toc103004352)

[106. Plot\_gradient\_boosting\_regularization 16](#_Toc103004353)

[107. Plot\_grid\_search\_digits 16](#_Toc103004354)

[108. Plot\_grid\_search\_refit\_callable 16](#_Toc103004355)

[109. Plot\_grid\_search\_stats 16](#_Toc103004356)

[110. Plot\_hashing\_vs\_dict\_vectorizer 16](#_Toc103004357)

[111. Plot\_huber\_vs\_ridge 16](#_Toc103004358)

[112. Plot\_ica\_blind\_source\_separation 16](#_Toc103004359)

[113. Plot\_ica\_vs\_pca 16](#_Toc103004360)

[114. Plot\_inductive\_clustering 16](#_Toc103004361)

[115. Plot\_iris\_dataset 16](#_Toc103004362)

[116. Plot\_iris\_dic 16](#_Toc103004363)

[117. Plot\_iris\_logistic 16](#_Toc103004364)

[118. Plot\_isolation\_forest 16](#_Toc103004365)

[119. Plot\_isotonic\_regression 16](#_Toc103004366)

[120. Plot\_iterative\_imputer\_variants\_comparison 16](#_Toc103004367)

[121. Plot\_johnson\_lindenstrauss\_bound 16](#_Toc103004368)

[122. Plot\_kde\_1d 16](#_Toc103004369)

[123. Plot\_kernel\_approximation 16](#_Toc103004370)

[124. Plot\_kmeans\_assumptions 16](#_Toc103004371)

[125. Plot\_kmeans\_digits 16](#_Toc103004372)

[126. Plot\_kmeans\_plusplus 16](#_Toc103004373)

[127. Plot\_kmeans\_silhouette\_analysis 16](#_Toc103004374)

[128. Plot\_kmeans\_stability\_low\_dim\_dense 16](#_Toc103004375)

[129. Plot\_label\_propagation\_digits 16](#_Toc103004376)

[130. Plot\_label\_propagation\_digits\_active\_learning 16](#_Toc103004377)

[131. Plot\_label\_propagation\_structure 17](#_Toc103004378)

[132. Plot\_lasso\_and\_elasticnet 17](#_Toc103004379)

[133. Plot\_lasso\_coordinate\_descent\_path 17](#_Toc103004380)

[134. Plot\_lasso\_dense\_vs\_sparse\_data 17](#_Toc103004381)

[135. Plot\_lasso\_lars 17](#_Toc103004382)

[136. Plot\_lasso\_lars\_ic 17](#_Toc103004383)

[137. Plot\_lasso\_model\_selection 17](#_Toc103004384)

[138. Plot\_Ida 17](#_Toc103004385)

[139. Plot\_Ida\_qda 17](#_Toc103004386)

[140. Plot\_learning\_curve 17](#_Toc103004387)

[141. Plot\_linear\_model\_coefficient\_interpretation 17](#_Toc103004388)

[142. Plot\_linearsvc\_support\_vectors 17](#_Toc103004389)

[143. Plot\_linkage\_comparison 17](#_Toc103004390)

[144. Plot\_lIe\_digits 17](#_Toc103004391)

[145. Plot\_lot\_novelty\_detection 17](#_Toc103004392)

[146. Plot\_lof\_outlier\_detection 17](#_Toc103004393)

[147. Plot\_logistic 17](#_Toc103004394)

[148. Plot\_logistic\_l1\_l2\_sparsity 17](#_Toc103004395)

[149. Plot\_logistic\_multinomial 17](#_Toc103004396)

[150. Plot\_logistic\_path 17](#_Toc103004397)

[151. Plot\_lw\_vs\_oas 17](#_Toc103004398)

[152. Plot\_mahalanobis\_sphere 17](#_Toc103004399)

[153. Plot\_manifold\_sphere 17](#_Toc103004400)

[154. Plot\_map\_data\_to\_normal 17](#_Toc103004401)

[155. Plot\_mds 17](#_Toc103004402)

[156. Plot\_mean\_shift 17](#_Toc103004403)

[157. Plot\_mini\_batch\_kmeans 18](#_Toc103004404)

[158. Plot\_missing\_values 18](#_Toc103004405)

[159. Plot\_mlp\_alpha 18](#_Toc103004406)

[160. Plot\_mlp\_training\_curves 18](#_Toc103004407)

[161. Plot\_mnist\_filters 18](#_Toc103004408)

[162. Plot\_model\_complexity\_influence 18](#_Toc103004409)

[163. Plot\_monotonic\_constraints 18](#_Toc103004410)

[164. Plot\_multi\_metric\_evaluation 18](#_Toc103004411)

[165. Plot\_multi\_task\_lasso\_support 18](#_Toc103004412)

[166. Plot\_nca\_classification 18](#_Toc103004413)

[167. Plot\_nca\_dim\_reduction 18](#_Toc103004414)

[168. Plot\_nca\_illustration 18](#_Toc103004415)

[169. Plot\_nearest\_centroid 18](#_Toc103004416)

[170. Plot\_nested\_cross\_validation\_iris 18](#_Toc103004417)

[171. Plot\_nnls 18](#_Toc103004418)

[172. Plot\_ols 18](#_Toc103004419)

[173. Plot\_ols\_3d 18](#_Toc103004420)

[174. Plot\_ols\_ridge\_variance 18](#_Toc103004421)

[175. Plot\_omp 18](#_Toc103004422)

[176. Plot\_oneclass 18](#_Toc103004423)

[177. Plot\_optics 18](#_Toc103004424)

[178. Plot\_out\_of\_core\_classification 18](#_Toc103004425)

[179. Plot\_outlier\_detection\_wine 18](#_Toc103004426)

[180. Plot\_particial\_dependence 18](#_Toc103004427)

[181. Plot\_particial\_dependence\_visualization\_api 18](#_Toc103004428)

[182. Plot\_pca\_3d 18](#_Toc103004429)

[183. Plot\_pca\_iris 19](#_Toc103004430)

[184. Plot\_pca\_vs\_fa\_model\_selection 19](#_Toc103004431)

[185. Plot\_pca\_vs\_Ida 19](#_Toc103004432)

[186. Plot\_pcr\_vs\_pls 19](#_Toc103004433)

[187. Plot\_permutation\_importance 19](#_Toc103004434)

[188. Plot\_permuation\_tests\_for\_classification 19](#_Toc103004435)

[189. Plot\_pipeline\_display 19](#_Toc103004436)

[190. Plot\_poisson\_regression\_non\_normal\_loss 19](#_Toc103004437)

[191. Plot\_polynomial\_interpolation 19](#_Toc103004438)

[192. Plot\_precision\_recall 19](#_Toc103004439)

[193. Plot\_quantile\_regression 19](#_Toc103004440)

[194. Plot\_random\_dataset 19](#_Toc103004441)

[195. Plot\_random\_forest\_embedding 19](#_Toc103004442)

[196. Plot\_random\_forest\_regression\_multioutput 19](#_Toc103004443)

[197. Plot\_random\_mulilabel\_dataset 19](#_Toc103004444)

[198. Plot\_randomized\_search 19](#_Toc103004445)

[199. Plot\_ransac 19](#_Toc103004446)

[200. Plot\_rbf\_parameters 19](#_Toc103004447)

[201. Plot\_rbm\_logistic\_classification 19](#_Toc103004448)

[202. Plot\_regression 19](#_Toc103004449)

[203. Plot\_release\_highlights\_0\_22\_0 19](#_Toc103004450)

[204. Plot\_release\_highlights\_0\_23\_0 19](#_Toc103004451)

[205. Plot\_release\_highlights\_0\_24\_0 19](#_Toc103004452)

[206. Plot\_release\_highlights\_1\_0\_0 19](#_Toc103004453)

[207. Plot\_rfe\_digits 19](#_Toc103004454)

[208. Plot\_rfe\_with\_cross\_validation 19](#_Toc103004455)

[209. Plot\_rfe\_digits 20](#_Toc103004456)

[210. Plot\_rfe\_with\_cross\_validation 20](#_Toc103004457)

[211. Plot\_ridge\_coeffs 20](#_Toc103004458)

[212. Plot\_ridge\_path 20](#_Toc103004459)

[213. Plot\_robust\_fit 20](#_Toc103004460)

[214. Plot\_roblust\_vs\_empirical\_covariance 20](#_Toc103004461)

[215. Plot\_roc 20](#_Toc103004462)

[216. Plot\_roc\_crossval 20](#_Toc103004463)

[217. Plot\_roc\_curve\_visualization\_api 20](#_Toc103004464)

[218. Plot\_scalable\_poly\_kernels 20](#_Toc103004465)

[219. Plot\_segmentation\_toy 20](#_Toc103004466)

[220. Plot\_select\_from\_model\_diabetes 20](#_Toc103004467)

[221. Plot\_self\_training\_varying\_threshold 20](#_Toc103004468)

[222. Plot\_semi\_supervised\_newsgroups 20](#_Toc103004469)

[223. Plot\_semi\_supervised\_versus\_svm\_iris 20](#_Toc103004470)

[224. Plot\_separating\_hyperplane 20](#_Toc103004471)

[225. Plot\_separating\_hyperplane\_unbalanced 20](#_Toc103004472)

[226. Plot\_sgd\_comparison 20](#_Toc103004473)

[227. Plot\_sgd\_early\_stopping 20](#_Toc103004474)

[228. Plot\_sgd\_iris 20](#_Toc103004475)

[229. Plot\_sgd\_loss\_functions 20](#_Toc103004476)

[230. Plot\_sdf\_penalities 20](#_Toc103004477)

[231. Plot\_sgd\_seperating\_hyperplane 20](#_Toc103004478)

[232. Plot\_sgd\_weightted\_samples 20](#_Toc103004479)

[233. Plot\_sgdocsvm\_vs\_ocsvm 20](#_Toc103004480)

[234. Plot\_sparse\_coding 20](#_Toc103004481)

[235. Plot\_sparse\_cov 21](#_Toc103004482)

[236. Plot\_sparse\_logistic\_regression\_20newsgroups 21](#_Toc103004483)

[237. Plot\_sparse\_logistic\_regression\_mnist 21](#_Toc103004484)

[238. Plot\_species\_distribution\_modeling 21](#_Toc103004485)

[239. Plot\_species\_kde 21](#_Toc103004486)

[240. Plot\_spectral\_biclustering 21](#_Toc103004487)

[241. Plot\_spectral\_coclustering 21](#_Toc103004488)

[242. Plot\_stack\_predictors 21](#_Toc103004489)

[243. Plot\_succesive\_halving\_heatmap 21](#_Toc103004490)

[244. Plot\_succesive\_halving\_iterations 21](#_Toc103004491)

[245. Plot\_svm\_anova 21](#_Toc103004492)

[246. Plot\_svm\_kernels 21](#_Toc103004493)

[247. Plot\_svm\_margin 21](#_Toc103004494)

[248. Plot\_svm\_nonlinear 21](#_Toc103004495)

[249. Plot\_svm\_regression 21](#_Toc103004496)

[250. Plot\_svm\_scale\_c 21](#_Toc103004497)

[251. Plot\_svm\_tie\_breaking 21](#_Toc103004498)

[252. Plot\_swissroll 21](#_Toc103004499)

[253. Plot\_t\_sne\_perlexity 21](#_Toc103004500)

[254. Plot\_theilsen 21](#_Toc103004501)

[255. Plot\_tomography\_l1\_reconstruction 21](#_Toc103004502)

[256. Plot\_topics\_extraction\_with\_nmf\_Ida 21](#_Toc103004503)

[257. Plot\_train\_error\_vs\_test\_error 21](#_Toc103004504)

[258. Plot\_transformed\_target 21](#_Toc103004505)

[259. Plot\_tree\_regression 21](#_Toc103004506)

[260. Plot\_tree\_regression\_multioutput 21](#_Toc103004507)

[261. Plot\_tweedie\_regression\_insurance\_claims 22](#_Toc103004508)

[262. Plot\_underfitting\_overfitting 22](#_Toc103004509)

[263. Plot\_unveil\_tree\_structure 22](#_Toc103004510)

[264. Plot\_validation\_curve 22](#_Toc103004511)

[265. Plot\_varimax\_fa 22](#_Toc103004512)

[266. Plot\_voting\_decision\_regions 22](#_Toc103004513)

[267. Plot\_voting\_regressor 22](#_Toc103004514)

[268. Plot\_ward\_structured\_vs\_unstructued 22](#_Toc103004515)

[269. Plot\_weighted\_samples 22](#_Toc103004516)

[270. Scene\_classification 22](#_Toc103004517)

[271. Svm\_gui 22](#_Toc103004518)

[272. Tray\_food\_segmentation 22](#_Toc103004519)

[273. Wikipedia\_principal\_eigenvector 22](#_Toc103004520)

# URL

http://175.197.5.7:20013/tree?

# Approximate\_nearest\_neighbors

|  |  |
| --- | --- |
| 제목 | Approximate nearest neighbors in TSNE |
| 현재 상태 | Install 에러 발생 |
| Approximate nearest neighbor |  |
| TSNE | - 유사성이 높은 점들을 다차원 공간에서 가까운 점들로 매핑하는 알고리즘  즉 벡터가 아니 어떤 대상을 벡터화 해주는 알고리즘  - 예를 들어 데이터의 연관성만 있을 때 연관성 그래프를 다차원 공간에 매핑  - 연관성이 많은 것은 가까이 모이고 먼 대상은 멀리 배치함 |
| annoy | Approximate Nearest Neighbors  - 가장 비슷한 벡터를 빠르게 찾아주는 알고리즘  - spotify에서 만든 알고리즘으로 spotify에서 실제로 음악을 추천할 때 사용 |
| nearest neighbor | - 최근접 이웃  - 공간에서 가장 가까운 벡터를 찾는 알고리즘  - word2vec에서 유사한 단어를 검색할 때 사용  - vector 간의 거리를 계산할 때 사용  - 추천 시스템에서 유사한 벡터를 찾아서 추천할 때 사용  - 현재 사용장이 사용 기록을 벡터로 만든 다음 다른 사용자들의 사용 기록에 대한 벡터와 nearest neighbor 알고리즘을 통해 유사한 벡터 검색  - vector의 수가 1M가 넘어가면 비용도 많이 들어가고 성능도 떨어짐  -5 nearest neighbors 예제    -20 nearest neighbors 예제    -100 nearest neighbors |
| MNIST 데이터 측정 | 이미지를 벡터로 표현할 수 있다는 것이 중요한 아이디어임 |
| MNIST neighbors |  |
| Nearest food pics |  |
| k-nearest neighbor(knn) | 새로운 데이터를 입력 받을 때 가장 가까이 있는 것이 무엇이냐를 중심으로 새로운 데이터의 종류를 정해주는 알고리즘    위의 그림에서 물음표에 무엇이 들어가야 하는가 판단할 때 가장 가까운 도형이 세모이므로 세모로 판단하는 알고리즘    K=1일 때는 물음표를 동그라미로 판단하지만 k=4일 때는 세모로 판단 |
|  |  |
| k-nn 분류기 | -데이터가 아래과 같이 주어졌을 경우    -입력데이터가 들어오면    -k=4일 경우 입력 데이터는 파란색으로 판단 |
| MNIST | 필기체 숫자 데이터 분류에 사용되는 샘플 |
| t-sne 사례 | t-sne를 사용해 MNIST 데이터를 2차원 공간에 매핑 |
|  |  |

# Brain\_tumor\_detection

# Breast\_cancer\_prediction

# Chess\_piece\_detection

# Eda\_for\_pubg

# Grid\_search\_text\_feature\_extraction

# House\_price\_prediction

# Mask\_or\_no\_mask

# Open\_images\_eda

# Plot\_adaboost\_hastie\_10\_2

|  |  |
| --- | --- |
| 제목 | Discrete versus Real AdaBoost |
| 결과 | - Discrete SAMME AdaBoost는 예측된 클래스 레이블의 오류를 기반으로하고 real SAMM.R AdaBoost는 예측된 클래스의 확률을 기반으로 학습한다. |
| 목적 | Discrete SAMME AdaBoost 알고리즘과 real SAMM.R AdaBoost 알고리즘의 성능 차이를 보여줌 |

# Plot\_adaboost\_multiclass

|  |  |
| --- | --- |
| 제목 | Multi-class AdaBoosted Decision Trees |
| 결과 | -왼쪽: 테스트 에러가 tree수가 증가하면서 감소한다.  -가운데: 분류 오류는 크게 변함이 없다.  -오른쪽: 각 tree의 가중치는 tree 수가 증가면서 점차 감소한다. |
| 목적 | 다중 분류 문제에 boosting이 예측 정확도 개선에 도움이 된다는 것을 보여줌 |
| boosting | 이전 분류기의 학습 결과를 토대로 다음 분류기의 학습 데이터의 샘풀가중치를 조정  -절차 |
| bootstrap sampling | -비용과 시간이 많이 드는 데이터 수집을 스스로 해결할 수 있는 샘플링 방법  -N개의 sample data를 가지고 있을 때 1000개의 bootstrap samples를 만들고자 하면, 복원 추출을 N번 실행하여 새로운 sample data set을 만들고 이 작업을 1000번 반복한다. |

# Plot\_adaboost\_regression

|  |  |
| --- | --- |
| 제목 | Decision Tree Regression with AdaBoost |
| 결과 | - boost 수가 증가함에 따라 sample을 더 정확히 예측할 수 있다. |
| 목적 | Decision Tree의 성능을 향상시키기 위해 Adaboost.R2 알고리즘을 사용한다. |

# Plot\_adaboost\_twoclass

|  |  |
| --- | --- |
| 제목 | Two-class AdaBoost |
| 결과 | - 두 개의 “Gaussian quantiles” clusters로 구성된 비선형 데이터에 대한 결정 경계(좌)와 결정 점수(우)  - 결정점수가 0보다 크면 B, 작으면 A로 분류 |

# Plot\_adjusted\_for\_chance\_measures

# Plot\_affinity\_propagation

# Plot\_agglomerative\_clustering

|  |  |
| --- | --- |
| 1 | agglomerative clustering:  -시작할 때 각 포인트를 하나의 클러스터로 지정하고, 종료 조건을 만족할 때까지 가장 비슷한 두 클러스터를 합치는 방법  -병합군집이라고도 한다. |
| 2 | Connectivity matrix:  그래프에서 어느 꼭짓점들이 변으로 연결되었는지 나타내는 정사각 행렬 |
| 3 | 병합군집에서 connectivity matrix를 사용하지 않았을 때(즉 각 노드가 이웃 노드와 연결되어 있는지에 대한 정보를 사용하지 않을 경우) |
| 4 | connectivity matrix를 사용할 경우 |

# Plot\_agglomerative\_clustering\_metrics

|  |  |
| --- | --- |
| 1 | wave form 데이터: |
|  |  |
|  |  |

# Plot\_agglormerarive\_dendrogram

# Plot\_anomaly\_comparison

# Plot\_ard

|  |  |
| --- | --- |
| 제목 | Automatic Relevance Determination Regression (ARD) |
| 결과 | - 가중치를 OLS estimate와 비교하면 ARD estimate는 안정화 된 것을 확인할 수 있다.    - 가중치의 히스토그램이 뾰족한 이유는 sparsity-inducing prior이 암시되기 때문이다.    - ARD에 대한 예측 및 불확실성을 나타낸다. 우측으로 갈수록 불확실성이 증가하는 이유는 test sample이 train sample의 범위를 벗어났기 때문이다. |

# Plot\_bayesian\_ridge

|  |  |
| --- | --- |
| 제목 | Bayesian Ridge Regression |
| 결과 | - 가중치를 OLS estimate와 비교하면 ARD estimate는 안정화 된 것을 확인할 수 있다.  - ARD에 대한 예측 및 불확실성을 나타낸다. 우측으로 갈수록 불확실성이 증가하는 이유는 test sample이 train sample의 범위를 벗어났기 때문이다. |

# Plot\_bayesian\_ridge\_curvefit

|  |  |
| --- | --- |
| 제목 | Curv Fitting with Bayesian Ridge Regression |
| 결과 | - log marginal likelihood(L)이 큰 모델이 더 피팅이 잘 된것을 볼 수 있다. |
| 목적 | 일반적으로 Bayesian ridge regression으로 다항식 곡선을 피팅할 때 정규화 매개변수 (alpha, lambda)의 초기값 선택이 중요하기 때문에 사용 |

# Plot\_beta\_divergence

# Plot\_bias\_variance

|  |  |
| --- | --- |
| 제목 | Single estimator versus bagging: bias-variance decomposition |
| 결과 | - 좌측 상단 : random dataset LS(파란점)에 대한 예측(진한 빨간선)을 보여준다. 이 그림에서 편차(빨간 영역)가 큰 반면 편향(청록색 곡선과 파란색 곡선의 차이)이 매우 낮음을 볼 수 있다.  - 좌측 하단 : 예상 평균 제곱 오차를 표시한다. 편향(파란색)이 낮고 분산(녹색)이 큰 것을 확인할 수 있다.  - 우측 : 동일한 플롯에 해당하지만 bagging 앙상블을 사용한다. 편향이 이전보다 크고 분산은 이전보다 낮다  - 편향은 이전보다 크지만 분산을 크게 줄일 수 있어 bagging이 더 나은 것을 확인 할 수 있다. |
| 목적 | 배깅 앙상블에 대한 single estimator의 예상 평균 제곱오차의 bias-variance 분해를 설명하고 비교하기 위해 사용 |

# Plot\_bicluster\_newsgroups

# Plot\_birch\_vs\_minibatchkmeans

# Plot\_caching\_nearest\_neighbors

|  |  |
| --- | --- |
| 제목 | Caching nearest neighbors |
| 결과 | - pipeline의 caching 속성을 사용하여 가장 가까운 neighbors 그래프를 caching한다. 첫 번째 call은 neighbors 그래프를 계산하기 때문에 느리지만 후속 call은 그래프를 다시 계산할 필요가 없기 때문에 더 빠르다. |
| 목적 | KNeighborsClassifier에서 사용하기 전에 K-nearest neighbors를 미리 계산하기 위해 사용한다. K-nearest neighbors를 미리 계산하면 더 정밀하게 매개변수를 제어할 수 있기 때문 |

# Plot\_calibration

|  |  |
| --- | --- |
| 제목 | Probability calibration of classifiers |
| 결과 | - 보정이 없는 Gaussian naive Bayes 분류기를 사용하여 추정된 확률을 sigmoid, isotonic 보정이 있는 경우와 비교한다. Isotonic 보정이 있는 모델이 이질적인 레이블이 있는 중간 cluster에 속하는 대부분의 샘플에 대해 0.5에 가까운 확률을 반환하는 확률 보정을 제공할 수 있음을 확인할 수 있다. |
| 목적 | 분류를 수행할 때 클래스 레이블 뿐만 아니라 관련 확률을 예측하려는 경우가 많다. 그러나 모든 분류기가 잘 보정된 확률을 제공하는 것은 아니며 과적합의 가능성이 있다. 따라서 예측된 확률의 별도 보정을 후처리로 수행하기 위해 사용한다. |

# Plot\_calibration\_curve

|  |  |
| --- | --- |
| 제목 | Probability Calibration Curves |
| 결과 | - Isotonic 보정은 거의 대각선 calibration 곡선에 가까운 것을 확인 할 수 있다. |
| 목적 | Calibration curve를 사용하여 예측 확률이 얼마나 잘 보정되었는지 시각화하기 위해 사용 |

# Plot\_calibration\_multiclass

# Plot\_changed\_only\_pprint\_parameter

# Plot\_classification

# Plot\_classification\_probability

# Plot\_classifier\_chain\_yeast

# Plot\_classifier\_comparison

|  |  |
| --- | --- |
| 제목 | Classifier comparison |
| 결과 |  |
| Support Vector Classifier |  |
| Kernel SVM |  |
| Decision Tree |  |
| Random Forest |  |
| Algorithm cheet-sheet |  |
| Classifier Accuracy |  |
| Classifier Log Loss |  |
| accuracy |  |
| Accuracy comparison |  |

# Plot\_cluster\_comparison

|  |  |
| --- | --- |
| 제목 | Comparing different clustering algorithms on toy datasets |
| 결과 |  |
| kmeans cluster |  |
| AffinityPropagation |  |
| MeanShift |  |
| SpecturalClustering |  |
| Agglomaerative Clustering |  |
| DBSCAN |  |
| HDBSCAN |  |
|  |  |

# Plot\_cluser\_iris

# Plot\_coin\_segmentation

# Plot\_coin\_ward\_segmentation

# Plot\_color\_quantization

# Plot\_column\_transformer

# Plot\_column\_transformer\_mixed\_types

# Plot\_compare\_calibration

|  |  |
| --- | --- |
| 제목 | Comparison of Calibration of Classifiers |
| 결과 | - LogisticRegression이 logloss를 직접 최적화 하므로 가장 잘 보정된 예측을 반환하는 것을 볼 수 잇다. |
| 목적 | l[ogistic regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression), [Gaussian Naive Bayes](https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes), [Random Forest Classifier,](https://scikit-learn.org/stable/modules/ensemble.html#forest) [Linear SVM](https://scikit-learn.org/stable/modules/svm.html#svm-classification) 의 4가지 다른 모델의 보정을 비교하기 위해 사용 |

# Plot\_compare\_cross\_decomposition

# Plot\_compare\_krr

# Plot\_compare\_methods

# Plot\_compare\_reduction

# Plot\_concentration\_prior

# Plot\_cost\_complexity\_pruning

|  |  |
| --- | --- |
| 제목 | Post pruning decision trees with cost complexity pruning |
| 결과 | - 최대 effective alpha 값은 노드가 하나뿐인 tree이므로 제거된다.    - alpha가 증가함에 따라 노드의 수와 트리 깊이가 감소한다.    - alpha가 증가함에 따라 더 많은 tree가 가지치기되어 더 잘 일반화 되는 Decision Tree가 생성된다. Alpha=0.015로 설정되면 test정확도가 최대가 된다. |
| 목적 | Cost complexity(비용 복잡성)을 고려한 Decision Tree 가지치기 방법을 설명하기 위해 사용한다. |

# Plot\_covariance\_estimation

|  |  |
| --- | --- |
| 제목 | Shrinkage covariance estimation: LedoitWolf vs OAS and max-likelihood |
| 결과 | - Legoit-Wolf estimate는 최적에 가깝고 OAS estimate는 조금 더 떨어져있다. 그러나 이 두 방식 모두 교차검증보다 비용이 적게들고 성능 또한 좋다. |
| 목적 | Covariance estimation의 일반적인 접근은 maximum likelihood estimator이다. 그러나 편향되지 않아 공분산으로 수렴해버린다. 이를 방지하기 위해 사용한다. |

# Plot\_cv\_diabetes

# Plot\_cv\_digits

# Plot\_cv\_indices

|  |  |
| --- | --- |
| 제목 | Visualizing cross-validation behavior in scikit-learn |
| 결과 | - 먼저 데이터의 구조를 이해해야하기 때문에 데이터를 시각화 한다.    - cross-validation 동작을 시각화 한다.  - 그러나 KFold cross-validation은 데이터의 class나 group을 고려하지 않는다.  - StratifiedKFold, GroupKFlod, StratifiedGroupKFold를 사용하여 변경할 수 있다.    - StratifiedKFold : 각 클래스에 대한 샘플의 백분율을 유지한다.    - GroupKFlod : 동일한 그룹이 두 개의 다른 Fold에 나타나지 않도록 한다.    - StratifiedGroupKFold : 계층화된 Fold를 반환하려고 시도하는 동안의 제약조건을 유지한다.  - Legoit-Wolf estimate는 최적에 가깝고 OAS estimate는 조금 더 떨어져있다. 그러나 이 두 방식 모두 교차검증보다 비용이 적게들고 성능 또한 좋다. |
| 목적 | 올바른 cross-validation object를 선택하는 것은 중요하다. 모델 과적합을 피하고 데이터를 적절하게 분할하기위해 시각화를 통한 정보를 얻기 위해 사용한다. |

# Plot\_cv\_predict

# Plot\_cyclical\_feature\_engineering

# Plot\_dbscan

|  |  |
| --- | --- |
| 제목 | Demo of DBSCAN clustering algorithm |
| 결과 | - Cluster 간의 거리가 짧고 데이터가 세밀하게 몰려있는데도 분류가 잘 된 것을 볼 수 있다. |
| 목적 | 데이터가 세밀하게 몰려 있어서 밀도가 높은 부분을 군집화하기 위해서 사용한다. |

# Plot\_det

|  |  |
| --- | --- |
| 제목 | Detection error tradeoff(DET) curve |
| 결과 | - 좌측의 ROC곡선은 유사한 반면에 DET곡선은 차이가 명확히 보인다. |
| 목적 | ROC 곡선은 플로팅하는 데 사용되는 선형 척도로 인해 다른 분류기는 일반적으로 그래프의 왼쪽 상단 모서리에서만 다르며 플롯의 많은 부분에서 유사하게 나타난다. 반면에 DET 곡선은 정상적인 편차 척도에서 직선을 나타내기 때문에 전체적으로 구별되는 보습을 보이기 때문에 사용한다. |

# Plot\_dict\_face\_patches

# Plot\_digits\_agglomeration

# Plot\_digits\_classification

|  |  |
| --- | --- |
| 제목 | Recognizing hand-written digits |
| 결과 | - 숫자 dataset은 8x8 픽셀로 구성되어있다. Target값은 숫자 이미지의 상단에 있다.    - 분류기를 적용하여 숫자 이미지를 예측한 결과를 이미지 상단에 표시하였다.    - 실제값과 예측값 사이의 Confusion Matrix이다. |
| 목적 | 0에서 9까지 손으로 쓴 숫자의 이미지를 인식하기 위해 사용 |

# Plot\_digits\_classification\_exercise

# Plot\_digits\_denoising

|  |  |
| --- | --- |
| 제목 | Image denoising using kernel PCA |
| KPCA | Kernel Principal component analysis  - kernel을 사용하여 non-linear dimensionality reduction을 수행하는 방법 |
| KPCA의  응용 | Denoising, compression, structure3d prediction |
| 배경 | - 많은 머신러닝 알고리즘은 입력 데이터가 선형적으로 구분 가능하다고 가정함  - 실정에는 비선형적인 문제들이 많이 발생 |
|  |  |

# Plot\_digits\_kde\_sampling

# Plot\_digits\_last\_image

# Plot\_digits\_linkage

|  |  |
| --- | --- |
| 제목 | Various Agglomerative Clustering on a 2D embedding of digits |
| 결과 | - 대부분의 숫자를 포함하는 매우 큰 클러스터, 대부분의 숫자가 0인 중간 크기의 클러스터, 변두리 주변 노이즈 포인트에서 가져온 다른 클러스터로 훨씬 더 pathologic하다.      - 다른 옵션들은 더 고르게 분포된 클러스터로 이어지므로 dataset의 re-random sampling에 덜 민감할 수 있다. |
| 목적 | 숫자 데이터의 2D embedding에서 Agglomerative Clustering을 위한 다양한 연결 옵션을 직관적으로 보여주기 위해 사용 |

# Plot\_digits\_pipe

# Plot\_discretization

|  |  |
| --- | --- |
| 제목 | Using KBinsDiscretizer to discretize continuous features |
| 결과 | - descretization 전 linear regression 모델은 구축이 빠르고 해석이 비교적 간단하지만 선형관계만 모델링 할 수 있고, Decision tree 모델은 훨씬 복잡한 모델링이 가능하다. descretization 후 linear regression과 decision tree 모델이 정확히 동일한 예측을 수행한 것을 볼 수 있다. |
| 목적 | 연속 데이터에서 선형 모델의 성능을 더 향상 시키는 방법 중 하나가 이산화(binning)이기 때문에 사용한다. |

# Plot\_discretization\_classification

|  |  |
| --- | --- |
| 제목 | Feature discretization |
| 결과 | - 세 번째 행은 선형으로 분리가 가능하고 나머지는 불가능하다.  - 그래프마다 우측 하단에 있는 숫자는 분류 정확도를 보여준다.  - descretization은 선형으로 분리할 수 없는 데이터에 대해서는 성능을 향상 시키지만 선형으로 분리 가능한 데이터에 대해서는 성능을 감소 시킨다. |
| 목적 | Discretization이 선형으로 분리할 수 있는 데이터와 분리할 수 없는 데이터에 대한 성능을 비교하기 위해 사용한다. |

# Plot\_descretization\_strategies

# Plot\_display\_object\_visualization

|  |  |
| --- | --- |
| 제목 | Visualizations with Display Objects |
| 결과 | - Confusion Matrix    -ROC Curve    - 정밀도, 재현율 곡선 |
| 목적 | 모델의 예측이 이미 계산되었거나 계산 비용이 많이들 때 사용한다. |

# Plot\_document\_classification\_20newsgroups

|  |  |
| --- | --- |
| 제목 | Classification of text documents using sparse features |
| 결과 | - 각 분류기의 정확도, 학습시간, 테스트 시간을 나타냅니다. |
| 목적 | Bag-of-words 접근 방식을 사용하여 주제별로 문서를 분류하기 위해 사용 |

# Plot\_document\_clustering

# Plot\_elastic\_net\_precomputed\_gram\_matrix\_with\_weighted\_samples

# Plot\_ensemble\_oob

|  |  |
| --- | --- |
| 제목 | OOB Errors for Random Forests |
| 결과 |  |
| 목적 | OOB(out-of-bag) error를 통해 randomforestclassifier가 훈련되는 동안 적합성을 검증하기 위해 사용한다. |

# Plot\_f\_test\_vs\_mi

|  |  |
| --- | --- |
| 제목 | Comparison of F-test and mutual information |
| 결과 | - 0에서 1사이에 균일하게 분포된 x1, x2, x3에 대해 F-test와 mutal information  - F-test는 선형 종속성만 포착하므로 x1을 가장 구별되는 기능으로 평가한다.  - mutual information은 변수간의 종속성을 포착할 수 있어 x2를 가장 구별되는 기능으로 평가한다.  - x3는 두 방법 모두 관련없는 것으로 올바르게 평가한다. |
| 목적 | F-test와 mutual information의 차이점을 확인하기 위해 사용 |

# Plot\_face\_compress

# Plot\_face\_recognition

|  |  |
| --- | --- |
| 제목 | Faces recognition example using eigenfaces and SVMs |
| 결과 | - 모델의 품질에 대한 confusion matrix    - test dataset 일부에 대한 예측 결과 |
| 목적 | SVM 알고리즘을 이용해 사람의 얼굴을 인식하기 위해 사용 |

# Plot\_faces\_decomposition

# Plot\_feature\_agglomeration\_vs\_univariate\_selection

# Plot\_feature\_selection

|  |  |
| --- | --- |
| 제목 | Univariate Feature Selection |
| 결과 | - 전체 feature set 에서 첫 번째 4개만 중요한 것을 볼 수 있다. 이 feature들이 univariate feature selection에서 가장 높은 점수를 갖음을 볼 수 있다. |
| 목적 | SVM은 각 feature에 중요한 feature에 큰 가중치를 할당하지만 중요하지 않은 feature도 많이 선택한다. SVM 모델을 적용하기 전에 univariate feature selection을 적용하면 중요한 특성에 기인한 SVM의 가중치가 증가하므로 성능이 향상되기 때문에 사용 |

# Plot\_feature\_selection\_pipeline

# Plot\_feature\_transformation

|  |  |
| --- | --- |
| 제목 | Feature transformations with ensembles of trees |
| 결과 | - 각 모델별로 roc curv를 통해서 성능을 파악할 수 있다. |
| 목적 | Tree ensemble의 모델별로 각각 다른 roc curv를 그리기 위해 사용 |

# Plot\_feature\_union

# Plot\_forest\_importances

|  |  |
| --- | --- |
| 제목 | Feature importances with a forest of trees |
| 결과 | - 파란색 막대는 tree간의 변동성과 forest 기능의 중요도를 나타낸다.   - mean decrease in impurity(평균 불순물 감소) 를 기반으로 한 중요도  - feature permutation(기능순열) 에 기반한 중요도 - 두가지 방법 모두 동일한 feature가 가장 중요한 것으로 평가한다. 상대적 중요성을 다르지만 mean decrease in impurity 중요도는 feature permutation 중요도 보다 기능을 완전히 생략할 가능석이 적다. |
| 목적 | Artificial classification 작업에서 feature의 중요성을 평가하기 위해 사용한다. |

# Plot\_forest\_importances\_faces

|  |  |
| --- | --- |
| 제목 | Pixel importances with a parallel forest of trees |
| 결과 | - 픽셀의 불순물 기반 중요도(MDI)는 파악할 수 있지만 얼굴의 형태를 알아 볼 수 없는 문제점이 있다. |
| 목적 | 이미지 분류 작업에서 픽셀의 불순물 기반 중요도(MDI)를 평가하기 위해 사용한다. |

# Plot\_forest\_iris

|  |  |
| --- | --- |
| 제목 | Plot the decision surfaces of ensembles of trees on the iris dataset |
| 결과 | - 첫 번째 행 : 분류기가 꽃받침 너비와 꽃받침 길이 특성만 사용함  - 두 번째 행 : 꽃잎 길이와 꽃받침 길이만 사용  - 세 번째 행 : 꽃잎 너비와 꽃잎 길이만 사용 |
| 목적 | iris dataset의 특징에 대한 decision surface을 시각화하여 모델별로 비교하기 위해 사용 |

# Plot\_gmm

# Plot\_gmm\_covariances

# Plot\_gmm\_pdf

# Plot\_gmm\_selection

# Plot\_gmm\_sin

# Plot\_gpc

# Plot\_gpc\_iris

# Plot\_gpc\_isoprobability

# Plot\_gpc\_xor

# Plot\_gpr\_co2

# Plot\_gpr\_noisy

# Plot\_gpr\_noisy\_targets

# Plot\_gpr\_on\_structured\_data

# Plot\_gpr\_prior\_psterior

# Plot\_gradient\_boosting\_categorical

|  |  |
| --- | --- |
| 제목 | Categorical Feature Support in Gradient Boosting |
| 결과 | - one-hot-encoded data가 있는 모델의 학습 시간이 가장 느린 것을 알 수 있다. One-hot-encoding은 각 범주의 feature당 하나의 추가기능을 생성하므로 예상된 결과다. |
| 목적 | Categorical feature에 대한 다양한 인코딩 방법을 사용하여  HistGradientBoostingRegressor의 학습 시간과 예측 성능을 비교하기 위해 사용 |

# Plot\_gradient\_boosting\_early\_stopping

|  |  |
| --- | --- |
| 제목 | Early stopping of Gradient Boosting |
| 결과 | - early stopping을 적용했을 때에도 비슷한 성능을 나타낸다.    - 반면 학습 시간은 상당히 줄어든 것을 확인할 수 있다. |
| 목적 | Early stopping이란 모델의 score가 더이상 향상되지 않을 때 학습을 중지시키는 것을 말한다. 이는 훈련 시간, 메모리 사용량 및 예측 대기 시간을 줄이기 위해 사용한다. |

# Plot\_gradient\_boosting\_oob

|  |  |
| --- | --- |
| 제목 | Gradient Boosting Out-of-Bag estimates |
| 결과 | - 처음 100회 iteration에서 test loss보다 더 나은 추정치를 제공한다. |
| 목적 | OOB(out-of-bag) estimates는 boosting iteration의 최적의 수를 추정하기 위해 사용한다. |

# Plot\_gradient\_boosting\_quantile

|  |  |
| --- | --- |
| 제목 | Prediction Intervals for Gradient Boosting Regression |
| 결과 | - gradient boosting regression을 이용한 예측에 대한 신뢰구간을 보여준다. |
| 목적 | 예측 신뢰구간을 만들기 위해 사용한다. |

# Plot\_gradient\_boosting\_regression

|  |  |
| --- | --- |
| 제목 | Gradient Boosting Regression |
| 결과 | - train, test에 대한 편차를 나타낸다.    - 불순물 기반 및 순열 방법에 feature 중요도를 나타낸다. |
| 목적 | Gradient boosting regression은 회귀 및 분류 문제에서 사용할 수 있는 앙상블 기법이다. |

# Plot\_gradient\_boosting\_regularization

|  |  |
| --- | --- |
| 제목 | Gradient Boosting regularization |
| 결과 | - 축소를 통한 정규화(learning rate<1.0), shrinkage와 확률적 기울기  boosting(subsample<1.0) 모두 성능을 향상 시킨다. |
| 목적 | Regularization(정규화)는 모델의 성능을 향상시키기 위해서 사용한다. |

# Plot\_grid\_search\_digits

|  |  |
| --- | --- |
| 제목 | Parameter estimation using grid search with cross-validation |
| 목적 | GridSearchCV 개체를 사용하여 수행되는 cross-validation을 통해 분류기를 최적화 시키기 위해 사용한다. |

# Plot\_grid\_search\_refit\_callable

# Plot\_grid\_search\_stats

|  |  |
| --- | --- |
| 제목 | Statistical comparison of models using grid search |
| 결과 | - Fold에 따라 모델의 성능이 크게 좌우된다. 2\_poly(2차 다항식)의 경우 성능이 가장 안좋은 것을 볼 수 있다.    - 가장 성능이 좋은 RBF모델과 두 번째인 Linear모델의 성능 차이를 계산하기 위한 사후 분포 곡선이다.  - RBF모델이 Linear모델보다 더 나을 확률을 사후 분포 곡선의 0에서 무한대까지의 면적을 통해 계산할 수 있다.(Linear모델이 RBF모델보다 나을 확률은 -무한대 부터 0까지의 면적으로 계산할 수 있다.) |
| 목적 | GridSeearchCV를 사용하여 훈련되고 평가된 모델의 성능을 통계적으로 비교하기 위해 사용 |

# Plot\_hashing\_vs\_dict\_vectorizer

# Plot\_huber\_vs\_ridge

# Plot\_ica\_blind\_source\_separation

# Plot\_ica\_vs\_pca

|  |  |
| --- | --- |
| 제목 | FastICA on 2D point clouds |
| 결과 | - ICA는 비 가우시안성이 높은 투영에 해당하는 특징 공간에서 방향을 찾는 알고리즘이다. 이러한 방향은 원래 특징 공간에서 직교할 필요는 없지만 모든 방향이 동일한 분산에 해당하는 백색 특징 공간에서는 직교한다.  반면에 PCA는 최대 분산을 설명하는 방향에 해당하는 원시 피쳐 공간에서 직교 방향을 찾는다.    - 좌측 상단 : 자유도가 낮은 2개의 비 가우시안 프로세스를 사용하여 독립 소스 시뮬레이션을 한다.  - 우측 상단 : 혼합하여 observation을 생성한다. PCA로 식별된 방향은 주확색 벡터로 표시된다.  - 좌측 하단 : PCA공간의 신호를 나타낸다.  - 우측 하단 : ICA를 실행하는 것은 이 공간에서 가장 큰 가우스가 아닌 방향을 식별하기 위해 회전을 찾는 것이다.  여기에서 우리는 자유도가 낮은 2개의 학생 T인 고도로 비 가우시안 프로세스를 사용하여 독립 소스를 시뮬레이션합니다(왼쪽 위 그림). 우리는 그것들을 혼합하여 관찰을 생성합니다(오른쪽 상단 그림). 이 원시 관측 공간에서 PCA로 식별된 방향은 주황색 벡터로 표시됩니다. PCA 벡터(왼쪽 하단)에 해당하는 분산으로 미백한 후 PCA 공간의 신호를 나타냅니다. ICA를 실행하는 것은 이 공간에서 가장 큰 가우스가 아닌 방향(오른쪽 아래)을 식별하기 위해 회전을 찾는 것에 해당합니다. |
| 목적 | 독립 구성요소 분성(ICA)와 주성분 분석(PCA)을 사용한 결과를 feature space에서 시각화 하기 위해 사용한다. |

# Plot\_inductive\_clustering

|  |  |
| --- | --- |
| 제목 | Inductive Clustering |
| 결과 |  |
| 목적 | Clustering은 data가 많을 경우 비용이 많이들 수 있다. Clustering 알고리즘은 귀납적이지 않아 매번 계산을 다시해야하기 때문이다. 이를 개선하고자 clustering을 사용하여 분류기로 귀납적 모델을 학습하여 비용을 절감할 수 있기 때문에 사용한다. |

# Plot\_iris\_dataset

|  |  |
| --- | --- |
| 제목 | The Iris Dateset |
| 결과 | - 이 데이터 세트는 150x4 numpy.ndarray에 저장된 3가지 다른 유형의 붓꽃(Setosa, Versicolour 및 Virginica) 꽃잎과 꽃받침 길이로 구성되어있다. |

# Plot\_iris\_dic

|  |  |
| --- | --- |
| 제목 | Plot the decision surface of decision trees trained on the iris dataset |
| 결과 | - 모든 feature pair에 대해 학습된 decision tree 함수를 표시한다.    - 모든 feature에 대해 함께 학습된 Decision Tree의 구조를 표시한다. |
| 목적 | Iris dataset을 decision tree를 통해 학습하고 decision surface를 plotting하기 위해 사용한다. |

# Plot\_iris\_logistic

|  |  |
| --- | --- |
| 제목 | Logistic Regression 3-class Classifier |
| 결과 | - 두 차원(꽃받침 길이 및 너비)에 대한 logistic regression 분류기의 경계를 나타낸다. |

# Plot\_isolation\_forest

|  |  |
| --- | --- |
| 제목 | IsolationForest example |
| 결과 | - 빨간색 점을 이상현상으로 탐지한 것을 볼 수 있다. |
| 목적 | 이상 현상을 탐지하기 위해 사용 |

# Plot\_isotonic\_regression

|  |  |
| --- | --- |
| 제목 | Isotonic Regression |
| 결과 | - Linear regression보다 Isotonic regression이 더 적합한 것을 볼 수 있다. |
| 목적 | Isotonic Regression은 단조성 외에 목표 함수에 대한 모양을 가정하지 않는 장점이 있기 때문에 사용한다. |

# Plot\_iterative\_imputer\_variants\_comparison

# Plot\_johnson\_lindenstrauss\_bound

|  |  |
| --- | --- |
| 제목 | The Johnson-Lindenstrauss bound for embedding with random projections |
| 결과 |  |
| 목적 |  |

# Plot\_kde\_1d

|  |  |
| --- | --- |
| 제목 | Simple 1D Kernel Density Estimation |
| 결과 | - histogram은 단위 블록이 일반 그리드의 각 점 위에 쌓이는 방식이다. 그러나 상단 두 패널에서는 이러한 블록에 대한 그리드를 선택하면 밀도 분포의 기본 모양에 대해 매우 다양한 아이디어가 나올 수 있다.  - 각 블록이 나타내는 점의 중앙에 위치하면 좌측하단 패널에 표시된 추정치를 얻을 수 있다. 이것은 ‘Top hat’ 커널을 사용한 커널 밀도 추정이다. 이 아이디어는 다른 커널 모양으로 일반화 될 수 있다.  - 우측 하단의 패널은 동일한 분포에 대한 가우스 커널 밀도 추정치를 보여준다.    - kernelDensity 추정기를 통해 BallTree 또는 KDTree 구조를 사용하여 구현한 커널 밀도 추정을 보여준다.    - 1차원에서 100개 샘플의 분포에 대한 커널 밀도 추정치를 비교한다. 더 높은 파원으로도 확장 가능하다. |
| 목적 | KernelDensity 클래스를 사용하여 1차원에서 커널 밀도 추정의 원칙을 확인하기 위해 사용 |

# Plot\_kernel\_approximation

|  |  |
| --- | --- |
| 제목 |  |
| 결과 |  |
| 목적 |  |

# Plot\_kmeans\_assumptions

|  |  |
| --- | --- |
| 제목 | Demonstration of k-means assumptions |
| 결과 | - 처음 세개의 plot에서 결과적으로 바람직하지 않은 cluster가 생성된다. |
| 목적 | Kmeans clustering의 직관적이지 않고 예상치 못한 cluster를 생성하는 상황을 설명하기 위해 사용한다. |

# Plot\_kmeans\_digits

|  |  |
| --- | --- |
| 제목 | A demo of K-Means clustering on the handwritten digits data |
| 결과 | - PCA(주성분 분석)을 통한 초기화에 대한 결과를 보여준다. |
| 목적 | K-Means clustering의 다양한 초기화 전략을 비교하기 위해 사용한다. |

# Plot\_kmeans\_plusplus

|  |  |
| --- | --- |
| 제목 | An example of K-Means++ initialization |
| 결과 |  |
| 목적 | K-Means의 초기화 용도로 사용되는 K-Means++의 출력결과를 시각화 하기 위해 사용한다. |

# Plot\_kmeans\_silhouette\_analysis

|  |  |
| --- | --- |
| 제목 | Selecting the number of clusters with silhouette analysis on KMeans clustering |
| 결과 | - 1에 가까운 실루엣 계수(이 값을 참조)는 샘플이 인접 클러스터에서 멀리 떨어져 있음을 나타낸다. 값 0은 샘플이 두 개의 인접 클러스터 사이의 결정 경계에 있거나 매우 가깝다는 것을 나타내고 음수 값은 해당 샘플이 잘못된 클러스터에 할당되었을 수 있음을 나타낸다.  이 예에서 실루엣 분석은 n\_clusters에 대한 최적 값을 선택하는 데 사용된다. 실루엣 플롯은 3, 5 및 6의 n\_clusters 값이 평균 실루엣 점수 미만인 클러스터의 존재와 실루엣 플롯 크기의 넓은 변동으로 인해 주어진 데이터에 대해 잘못된 선택임을 보여준다. 실루엣 분석은 2와 4 사이를 결정할 때 더 모호하다.  또한 실루엣 플롯의 두께에서 클러스터 크기를 시각화할 수 있다. n\_clusters가 2일 때 클러스터 0에 대한 실루엣 플롯은 3개의 하위 클러스터를 하나의 큰 클러스터로 그룹화하기 때문에 크기가 더 크다. 그러나 n\_clusters가 4와 같을 때 모든 플롯은 거의 비슷한 두께를 가지므로 오른쪽의 레이블이 지정된 산점도에서 확인할 수 있는 것과 같이 크기가 비슷하다. |
| 목적 | silhouette analysis는 cluster 사이의 separation distance를 영구하기 위해 사용한다. |

# Plot\_kmeans\_stability\_low\_dim\_dense

|  |  |
| --- | --- |
| 제목 | Empirical evaluation of the impact of k-means initialization |
| 결과 | - 초기화 수를 제어하는 n\_init 매개변수의 값을 증가시키기 위해 모델과 init 메소드의 각 조합에 대해 도달한 최상의 initialization을 보여준다.    - 두 번째 플롯은 init="random" 및 n\_init=1을 사용하는 MiniBatchKMeans 추정기의 단일 실행을 보여준다. 이 실행은 추정된 중심이 ground truth 클러스터 사이에 끼어 나쁜 수렴(로컬 최적)으로 이어진다. |
| 목적 | Initialization의 상대 표준편차로 측정된 알고리즘의 성능을 향상시키는 k-means 초기화 전략 능력을 평가하기 위해 사용한다. |

# Plot\_label\_propagation\_digits

|  |  |
| --- | --- |
| 제목 | Label Propagation digits: Demonstrating performance |
| 결과 | 이미지 삽입 중...  - confusion matrix    - 가장 불확실한 상위 10개의 예측 이미지 이다. |
| 목적 | 매우 적은 양의 데이터를 확산시켜 학습하는 semisupervised learning을 위해 사용한다. |

# Plot\_label\_propagation\_digits\_active\_learning

|  |  |
| --- | --- |
| 제목 | Label Propagation digits active learning |
| 결과 | - 10개의 label만 이용해 학습하고 가장 불확실한 5개의 예측 이미지를 출력한다. 다음으로 label을 5개씩 늘려가며 학습한다. |
| 목적 | Label propagation을 이용한 학습을 위해 사용한다. |

# Plot\_label\_propagation\_structure

|  |  |
| --- | --- |
| 제목 | Label Propagation learning a complex structure |
| 결과 | - 먼저 두 개의 동심원이 있는 dataset을 생성한다. Label은 0(외부 원), 1(내부 원), -1(알수없음)이다.    - raw data(원시 데이터)    - 각 샘플에 label이 잘 연결된 것을 볼 수 있다. |
| 목적 | ‘manifold learning’(다양성 학습)을 위해 사용 |

# Plot\_lasso\_and\_elasticnet

# Plot\_lasso\_coordinate\_descent\_path

# Plot\_lasso\_dense\_vs\_sparse\_data

# Plot\_lasso\_lars

# Plot\_lasso\_lars\_ic

# Plot\_lasso\_model\_selection

# Plot\_Ida

# Plot\_Ida\_qda

# Plot\_learning\_curve

|  |  |
| --- | --- |
| 제목 | Plotting Learning Curves |
| 결과 | - 1열 : naive bayes를 이용한 학습  - 2열 : RBF 커널이 있는 SVM을 이용한 학습  - 1행 : 학습 정확도  - 2행 : 모델이 다양한 크기의 train dataset으로 학습하는데 필요한 시간  - 3행 : 학습에 필요한 시간 |
| 목적 | 학습 결과를 시각화하기 위해 사용한다. |

# Plot\_linear\_model\_coefficient\_interpretation

|  |  |
| --- | --- |
| 제목 | Common pitfalls in the interpretation of coefficients of linear models |
| 결과 | - Data set의 구조를 봤을때는 experience와 같은 변수가 직관적으로 더 영향력이 있어보이지만 위의 표에서 가장 중요한 변수는 union으로 보인다.    - coefficient들을 정규화 하기 위해 표준편차를 곱한 결과    - 정규화 결과 coefficient를 해석해는데 발생하는 오류가 사라졌음을 알 수 있다. |
| 목적 | Linear model의 coefficient를 해석하는데 발생하는 문제점을 찾기 위해 사용 |

# Plot\_linearsvc\_support\_vectors

|  |  |
| --- | --- |
| 제목 | Plot the support vectors in LinearSVC |
| 결과 |  |
| 목적 | SVC(LIBSVM 기반)와 달리 LinearSVC(LIBLINEAR 기반)는 지원 벡터를 제공하지 않는다. 그래서 LinearSVC에서 지원 벡터를 얻는 방법을 보여주기 위해 사용한다. |

# Plot\_linkage\_comparison

|  |  |
| --- | --- |
| 제목 | Comparing different hierarchical linkage methods on toy datasets |
| 결과 | - single linkage : 학습이 빠르고 구형이 아닌 데이터에서 잘 수행될 수 있지만 noise가 있는 경우 성능이 저하된다.  - average linkage & complete linkage : 완전하게 분리된 data에 대해서는 잘 수행되지만 그렇지 않은경우 성능이 저하된다.  - ward linkage : noise가 있는 data에 가장 효과적인 방법이다. |
| 목적 | 2D인 data set에 대한 계층적 clustering을 위한 다양한 연결 방법의 특성을 보여주기 위해 사용한다. |

# Plot\_lIe\_digits

# Plot\_lot\_novelty\_detection

# Plot\_lof\_outlier\_detection

|  |  |
| --- | --- |
| 제목 | Outliner detection with Local Outlier Factor(LOF) |
| 결과 |  |
| LOF의 문제 의식 | Density가 상이한 클러스터들이 있을 때 이상치를 일괄적으로 찾아내는 방법은 없다.    위의 그림에서 o1은 걸러내기 쉽지만 o2는 걸러내기 어렵다. |
| LOF 개념 | 상대적인 dense를 고려하여 outliner 결정 |
|  |  |

# Plot\_logistic

|  |  |
| --- | --- |
| 제목 | Logistic function |
| 결과 | - logistic regression과 linear regression의 예측결과를 시각화한 plot이다. |
| 목적 | 범주형 데이터를 대상으로 독립변수의 선형 결합으로 종속변수를 설명하기 위해 사용한다. |

# Plot\_logistic\_l1\_l2\_sparsity

|  |  |
| --- | --- |
| 제목 | L1 Penalty and Sparsity in Logistic Regression |
| 결과 |  |
| 목적 | Elasticnet을 사용할때 l1, l2 penalty가 c값에 따라 어떻게 변하는지 확인하기 위해 사용 |

# Plot\_logistic\_multinomial

|  |  |
| --- | --- |
| 제목 | Plot multinomial and One-vs-Rest Logistic Regression |
| 결과 | - 다항식    - 일대일 |
| 목적 | 다항식 및 일대일 logistic regression의 결과를 시각화 하기 위해 사용 |

# Plot\_logistic\_path

|  |  |
| --- | --- |
| 제목 | Regularization path of L1- Logistic Regression |
| 결과 | - 강력한 정규화(plot의 좌측)가 적용된 경우 모든 coefficients는 정확히 0이고 정규화가 점진적으로 느슨해 지면 coefficients는 증가한다. |
| 목적 | 이진 분류 문제에 l1-logistic regression을 적용하기 위해 사용 |

# Plot\_lw\_vs\_oas

|  |  |
| --- | --- |
| 제목 | Ledoit-Wolf vs OAS estimation |
| 결과 |  |
| 목적 | 공분산 최대가능도 추정치를 shrinkage를 사용하여 정규화하기 위해 사용한다. |

# Plot\_mahalanobis\_sphere

# Plot\_manifold\_sphere

|  |  |
| --- | --- |
| 제목 | Manifold Learning methods on a severed sphere |
| 결과 | - manifold learning에 대한 직관을 얻기 위해 차원 축소를 사용한다.  - 2행 2열의 MDS는 다른 manifold learning 알고리즘과 달리 거리가 원래의 고차원 공간에서 거리를 잘 표현하는 데이터의 저차원 표현을 찾는 것이다.(평평한 지도를 나타내는ㄴ 문제와 상당히 유사하다.) |
| 목적 | 구형 dataset에 대한 다양한 manifold learning을 적용하기 위해 사용 |

# Plot\_map\_data\_to\_normal

# Plot\_mds

# Plot\_mean\_shift

|  |  |
| --- | --- |
| 제목 | A demo of the mean-shift clustering algorithm |
| 결과 |  |
| 목적 | Mean-shift(이동평균)을 이용한 clustering을 위해 사용 |

# Plot\_mini\_batch\_kmeans

|  |  |
| --- | --- |
| 제목 | Comparison of the K-Means and MiniBatchKMeans clustering algorithms |
| 결과 | - minibatch를 사용한 결과 학습 시간은 빠르고 결과는 약간 다르다. |
| 목적 | Minibatch를 이용한 kmeans 와 이용하지 않은 결과를 비교하기 위해 사용한다. |

# Plot\_missing\_values

|  |  |
| --- | --- |
| 제목 | Imputing missing values before building an estimator |
| 결과 | - 다양한 시도에 따른 결과 |
| 목적 | 결측치(missing value)를 대체하는 방법을 설명하기 위해 사용 |

# Plot\_mlp\_alpha

|  |  |
| --- | --- |
| 제목 | Varying regularization in Multi-layer Perceptron |
| 결과 |  |
| 목적 | Mlp의 다양한 정규화를 시각화하기 위해 사용한다. |

# Plot\_mlp\_training\_curves

|  |  |
| --- | --- |
| 제목 | Compare Stochastic learning strategies for MLPClassifier |
| 결과 |  |
| 목적 | SGD, Adam을 포함한 다양한 학습방법에 대한 손실곡선을 시각화 하기 위해 사용 |

# Plot\_mnist\_filters

# Plot\_model\_complexity\_influence

|  |  |
| --- | --- |
| 제목 | Model Complexity Influence |
| 결과 | - 더 복잡한(표현력 있는) 모델은 더 많은 훈련 시간이 필요하다.  - 더 복잡한 모델은 예측 오류를 항상 줄이는 것은 아니다. |
| 목적 | 모델의 복잡성이 예측 정확도와 계산 성능에 어떻게 영향을 미치는지 확인하기 위해 사용한다. |

# Plot\_monotonic\_constraints

# Plot\_multi\_metric\_evaluation

# Plot\_multi\_task\_lasso\_support

# Plot\_nca\_classification

|  |  |
| --- | --- |
| 제목 | Comparing Nearest Neighbors with and without Neighborhood Components Analysis |
| 결과 |  |
| 목적 | NCA(neighborhood components analysis) 사용여부에 따른 KNN 비교하기 위해 사용한다. |

# Plot\_nca\_dim\_reduction

|  |  |
| --- | --- |
| 제목 | Dimensionality Reduction with Neighborhood Components Analysis |
| 결과 | - 주성분 분석(PCA)은 데이터의 가장 큰 변동을 설명하는 속성의 조합(주성분 또는 특징 공간의 방향)을 식별한다.    - LDA(Linear Discriminant Analysis)는 클래스 간의 가장 큰 차이를 설명하는 속성을 식별한다.    - NCA(Neighborhood Components Analysis)는 확률적 최근접이웃 알고리즘이 최상의 정확도를 제공하도록 특성 공간을 찾으려고 한다.  - NCA는 차원의 큰 축소에도 불구하고 시각적으로 의미 있는 데이터의 클러스터링을 시행함을 알 수 있습니다. |
| 목적 | Digits dataset에 적용된 다양한 차원 축소 방법을 비교하기 위해 사용한다. |

# Plot\_nca\_illustration

|  |  |
| --- | --- |
| 제목 | Neighborhood Components Analysis Illustration |
| 결과 |  |
| 목적 | NCA(Neighborhood Components Analysis)을 이용한 분석결과를 시각화 하기 위해 사용한다. |

# Plot\_nearest\_centroid

|  |  |
| --- | --- |
| 제목 | Nearest Centroid Classification |
| 결과 |  |
| 목적 | 파라미터 변화에 따른 Nearest Centroid classification 결과를 시각화 하기 위해 사용한다. |

# Plot\_nested\_cross\_validation\_iris

|  |  |
| --- | --- |
| 제목 | Nested versus non-nested cross-validation |
| 결과 | - |
| 목적 | 중첩되지 않은 cross-validation과 중첩된 cross-validation을 비교하기 위해 사용한다. |

# Plot\_nnls

|  |  |
| --- | --- |
| 제목 | Non-negative least squares |
| 결과 | - OLS와 NNLS 간의 회귀계수를 비교하면 높은 상관관계가 있음을 볼 수 있다. |
| 목적 | 회귀 계수에 대한 양의 제약조건이 있는 선형모델과 고전적인 선형 모델을 비교하기 위해 사용한다. |

# Plot\_ols

# Plot\_ols\_3d

# Plot\_ols\_ridge\_variance

# Plot\_omp

# Plot\_oneclass

# Plot\_optics

|  |  |
| --- | --- |
| 제목 | Demo of OPTICS clustering algorithm |
| 결과 | - cluster를 확장 할수록 군집화가 잘 되는 것을 볼 수 있다. |
| 목적 | High density(고밀도) 코어 샘플을 찾아 cluster를 확장하기 위해 사용한다. |

# Plot\_out\_of\_core\_classification

# Plot\_outlier\_detection\_wine

|  |  |
| --- | --- |
| 제목 | Outlier detection on a real data set |
| 결과 | - 최소 공분산 결정자(Minimum Covariance Determinant)를 이용한 이상값 감지    - 데이터 분포에 집중해 최소 공분산 결정자를 이용한 이상값 감지 |
| 목적 | 이상값을 감지하기 위해 사용한다. |

# Plot\_particial\_dependence

# Plot\_particial\_dependence\_visualization\_api

# Plot\_pca\_3d

|  |  |
| --- | --- |
| 제목 | Principal components analysis (PCA) |
| 결과 |  |
| 목적 | 3차원의 데이터에 대한 주성분 분석을 위해 사용한다. |

# Plot\_pca\_iris

|  |  |
| --- | --- |
| 제목 | PCA example with Iris Data-set |
| 결과 |  |
| 목적 | Iris데이터에 대한 주성분 분석을 위해 사용한다. |

# Plot\_pca\_vs\_fa\_model\_selection

|  |  |
| --- | --- |
| 제목 | Model selection with Probabilistic PCA and Factor Analysis (FA) |
| 결과 | - 등분산 noise일 때 FA, PCA 모두 낮은 순위 부분 공간의 크기를 잘 복구 한다. 이경우 PCA의 likelihood가 더 높다.    - 이분산 noise가 존재할 때 PCA는 순위를 과대평가한다. |
| 목적 | 등분산 또는 이분산 noise로 손상된 데이터를 교차검증을 통해 PCA와 FAㅡㄹ 비교하기 위해 사용한다. |

# Plot\_pca\_vs\_Ida

# Plot\_pcr\_vs\_pls

|  |  |
| --- | --- |
| 제목 | Principal Component Regression vs Partial Least Squares Regression |
| 결과 | - PLS가 PCR보다 성능이 더 우수한 것을 확인할 수 있다. |
| 목적 | PCR(주성분 회귀)과 PLS(부분 최소 제곱 회귀)를 비교하기 위해 사용한다. |

# Plot\_permutation\_importance

|  |  |
| --- | --- |
| 제목 | Permutation Importance vs Random Forest Feature Importance (MDI) |
| 결과 | - 예측하지 않는 특성에 대해서도 중요도가 높을 수 있다(과적합)    - 불순물 기반 특성 중요도의 문제를 완화할 수 있는 순열 중요도를 사용한 결과 |
| 목적 | Permutation Importance(순열 중요도)와 imupurity based feature importance(불순물 기반 특성 중요도)를 비교하기 위해 사용한다. |

# Plot\_permuation\_tests\_for\_classification

# Plot\_pipeline\_display

# Plot\_poisson\_regression\_non\_normal\_loss

|  |  |
| --- | --- |
| 제목 | Poisson regression and non-normal loss |
| 결과 | - ridge는 분산이 일정한 반면에 poissonregressor 및 HistgradientBoostingRegressor에 사용된 poisson분포의 분산은 기대값에 비례한다. |
| 목적 | 일반적인 최소 제곱 오차가 적용된 선형 모델과 푸아송 손실이 적용된 비선형 GBRT 모델을 비교하기 위해 사용한다. |

# Plot\_polynomial\_interpolation

|  |  |
| --- | --- |
| 제목 | Polynomial and Spline interpolation |
| 결과 |  |

# Plot\_precision\_recall

|  |  |
| --- | --- |
| 제목 | Precision-Recall |
| 결과 | - 정밀도  - 재현율    - 정밀도-재현율 곡선 |
| 목적 | 모델의 품질 평가 지표인 Precision(정밀도), Recall(재현율)을 설명하기 위해 사용 |

# Plot\_quantile\_regression

# Plot\_random\_dataset

|  |  |
| --- | --- |
| 제목 | Plot randomly generated classification dataset |
| 결과 | - 처음 4개의 플롯은 다양한 수의 정보 기능, 클래스당 클러스터 및 클래스와 함께 make\_classification을 사용한다. 마지막 2개의 플롯은 make\_blobs 및  make\_gaussian\_quantiles를 사용합니다. |
| 목적 | 무작위로 생성된 여러 분류 data set을 시각화하기 위해 사용한다. |

# Plot\_random\_forest\_embedding

# Plot\_random\_forest\_regression\_multioutput

# Plot\_random\_multilabel\_dataset

# Plot\_randomized\_search

# Plot\_ransac

|  |  |
| --- | --- |
| 제목 | Robust linear model estimation using RANSAC |
| 결과 | - 노란색 점으로 표현된 데이터 때문에 linear regressor 모델은 영향을 받았고 RANSAC regressor 모델은 잘 적합된 것을 볼 수 있다. |
| 목적 | RANSAC알고리즘을 사용해 결함이 있는 데이터에 선형모델을 견고하게 fitting하기 위해 사용한다. |

# Plot\_rbf\_parameters

# Plot\_rbm\_logistic\_classification

# Plot\_regression

|  |  |
| --- | --- |
| 제목 | Nearest Neighbors regression |
| 최근접 회귀 | - 주변의 가장 가까운 k 개의 샘플을 통해 값을 예측하는 방식  - 예를 들어 k개 샘플의 평균을 이용 |
| 결과 |  |

# Plot\_release\_highlights\_0\_22\_0

|  |  |
| --- | --- |
| 제목 | Release Highlights for scikit-learn 0.22 |
| 결과 | - 새로운 plotting API추가(같은 그림에 다른 플롯 추가 가능)    - 모든 적합 추정량에 대해 각 기능의 중요도를 추정하는 데 사용가능 |

# Plot\_release\_highlights\_0\_23\_0

|  |  |
| --- | --- |
| 제목 | Release Highlights for scikit-learn 0.23 |
| 결과 | - 비정규 손실 함수가 있는 일반화 선형 모델 사용가능  - PossonRegressor, GammaRegressor, TweedieRegressor 추가    - histogram기반 gradient boosting classifier 개선(자동 early stop기능 추가) |

# Plot\_release\_highlights\_0\_24\_0

|  |  |
| --- | --- |
| 제목 | Release Highlights for scikit-learn 0.24 |
| 결과 | - 연속 반감기를 매개변수의 공간을 탐색하고 최적의 조합 식별 가능    - HistGradientBoosting classifier 범주형 기능에 대한 분석 가능    - 새로운 종류의 부분의존도플롯(ICE) 사용가능 |

# Plot\_release\_highlights\_1\_0\_0

|  |  |
| --- | --- |
| 제목 | Release Highlights for scikit-learn 1.0 |
| 결과 | - SplineTransformer 사용가능    - QuantileRegressor(분위수 회귀) 사용가능    - SGD-One-Class-SVM 사용 가능 |

# Plot\_rfe\_digits

# Plot\_rfe\_with\_cross\_validation

# Plot\_rfe\_digits

# Plot\_rfe\_with\_cross\_validation

# Plot\_ridge\_coeffs

|  |  |
| --- | --- |
| 제목 | Plot Ridge coefficients as a function of the L2 regularization |
| 결과 | - 좌측 : alpha(정규화)가 0에 가까워짐에 따라 ridge regression에 의해 발견된 계수는 무작위로 sampling된 벡터 w를 향해 안정화 된다. Alpha가 클경우 계수가 더 작아 더 간단하고 편향된 solution이 된다.  - 우측 : ridge regression 모델에서 찾은 계수와 w사이의 평균제곱오차를 나타낸다. 덜 정규화된 모델은 정확한 계수를 찾고 더 강하게 정규화된 모델은 오류를 증가 시킨다. |
| 목적 | Ridge regression으로 solution을 찾는 방법과 정규화가 계수와 해당 값에 미치는 영향을 보여주기 위해 사용한다. |

# Plot\_ridge\_path

|  |  |
| --- | --- |
| 제목 | Plot Ridge coefficients as a function of the regularization |
| 결과 | - 각 색상은 계수 벡터의 다른 특성을 나타내며 이는 정규화 매개변수의 함수로 표시된다.  - alpha가 매우 크면 정규화 효과가 손실함수를 지배하고 계수는 0이 되는 경향이 있다. path의 끝에서 alpha는 0을 향하고 solution은 보통 최소 제곱을 향하는 경향이 있으므로 계수는 큰 진동을 나타낸다. 실제로 둘 사이의 균형이 유지되는 방식으로 alpha를 조정할 필요가 있습니다. |
| 목적 | 추정기 계수에서 Collinearity(공선성)의 효과를 보여주기 위해 사용한다. |

# Plot\_robust\_fit

# Plot\_roblust\_vs\_empirical\_covariance

# Plot\_roc

|  |  |
| --- | --- |
| 제목 | Receiver Operating Characteristic (ROC) |
| 결과 | - y축 : 참 양성률  - x축 : 거짓 양성률  - plot의 좌측 상단은 ‘이상적인’ 지점을 의미한다.(거짓긍정률 = 0, 참긍정률=1 이기때문)  - ‘가파름’은 거짓 긍정 비율을 최소화 하면서 참 긍정 비율 최대화 하는 것이 이상적이기 때문에 중요하다. |
| 목적 | 모델의 품질 평가 지표인 ROC 곡선을 설명하기 위해 사용 |

# Plot\_roc\_crossval

|  |  |
| --- | --- |
| 제목 | Receiver Operating Characteristic (ROC) with cross validation |
| 결과 | - 모델의 출력이 train 데이터의 변경에 의해 어떻게 영향을 받는지, 그리고 K-Fold 교차 검증에 의해 생성된 분할이 서로 얼마나 다른지를 대략적으로 보여준다. |
| 목적 | Cross-validation을 사용하여 모델의 품질을 평가하기 위한 ROC 곡선을 설명하기 위해 사용한다. |

# Plot\_roc\_curve\_visualization\_api

|  |  |
| --- | --- |
| 제목 | ROC Curve with Visualization API |
| 결과 | - SVC에 대해 계산된 ROC곡선을 단일 호출로 plotting    - randomforest classifier를 학습하고 SVC ROC 곡선과 비교하기위해 ROC 곡선 plotting |
| 목적 | ROC 곡선을 비교하여 시각화하는 API를 설명하기 위해 사용한다. |

# Plot\_scalable\_poly\_kernels

# Plot\_segmentation\_toy

# Plot\_select\_from\_model\_diabetes

|  |  |
| --- | --- |
| 제목 | Model-based and sequential feature selection |
| 결과 | - 계수의 feature 중요도  - SelectFromModel: ['s1' 's5'] / Done in 0.001s  - sequential selection: ['bmi' 's5'] / Done in 0.348s  - SFS(SequentialFeatureSelection), SFM(SelectFromModel) 모두 동일한 집단을 선택 했으나 SFM은 ‘s1’계수를 사용하 속도가 빠르다. |
| 목적 | SelectFromModel(특징 중요도 기반)과 SequentialFeatureSelection(greedy한 접근 방식에 의존)을 비교하기 위해 사용한다. |

# Plot\_self\_training\_varying\_threshold

# Plot\_semi\_supervised\_newsgroups

# Plot\_semi\_supervised\_versus\_svm\_iris

# Plot\_separating\_hyperplane

# Plot\_separating\_hyperplane\_unbalanced

# Plot\_sgd\_comparison

|  |  |
| --- | --- |
| 제목 | Comparing various online solvers |
| 결과 |  |
| 목적 | 다양한 online solver의 오차를 비교하기 위해 사용한다. |

# Plot\_sgd\_early\_stopping

|  |  |
| --- | --- |
| 제목 | Early stopping of Stochastic Gradient Descent |
| 결과 | - early stopping 없이 구축된 모델과 거의 동일한 정확도를 보이고 학습 시간을 크게 단축할 수 있다. |
| 목적 | SGD(확률적 경사 하강법)의 손실 함수가 더이상 감소하지 않을 때 학습을 중단하기 위해 사용한다. |

# Plot\_sgd\_iris

|  |  |
| --- | --- |
| 제목 | Plot multi-class SGD on the iris dataset |
| 결과 | - multi-class classifier에 대한 분류는 평면에 색으로 표신된다.  - one-versus-all(OVA) classifier에 대한 분류는 dashed line으로 표시된다. |
| 목적 | Iris dataset에 대한 decision surface와 dashed line을 표시하기 위해 사용한다. |

# Plot\_sgd\_loss\_functions

|  |  |
| --- | --- |
| 제목 | SGD: convex loss functions |
| 결과 |  |
| 목적 | SGDClassifier에서 지원하는 다양한 손실함수를 비교하기 위해 사용한다. |

# Plot\_sdf\_penalities

# Plot\_sgd\_separating\_hyperplane

|  |  |
| --- | --- |
| 제목 | SGD: Maximum margin separating hyperplane |
| 결과 | - SGD를 사용하여 훈련된 SVM classifier로 2개 class로 분리 가능한 dataset을 분리하는 최대 margin을 나타낸다. |

# Plot\_sgd\_weighted\_samples

|  |  |
| --- | --- |
| 제목 | SGD: Weighted samples |
| 결과 | - 점의 크기는 가중치에 비례한다. |
| 목적 | SGD로 예측한 가중치를 나타내기 위해 사용한다. |

# Plot\_sgdocsvm\_vs\_ocsvm

# Plot\_sparse\_coding

# Plot\_sparse\_cov

# Plot\_sparse\_logistic\_regression\_20newsgroups

# Plot\_sparse\_logistic\_regression\_mnist

# Plot\_species\_distribution\_modeling

# Plot\_species\_kde

# Plot\_spectral\_biclustering

# Plot\_spectral\_coclustering

# Plot\_stack\_predictors

# Plot\_succesive\_halving\_heatmap

# Plot\_succesive\_halving\_iterations

# Plot\_svm\_anova

|  |  |
| --- | --- |
| 제목 | SVM-Anova: SVM with univariate feature selection |
| 결과 | - 약 10%의 feature를 선택할 때 모델이 최고의 성능을 달성한다는 것을 알 수 있다. |
| 목적 | 성능향상을 위해 SVC(support vector classifier)를 실행하기 전에 일변량 feature 선택을 수행하기 위해 사용한다. |

# Plot\_svm\_kernels

|  |  |
| --- | --- |
| 제목 | SVM-Kernels |
| 결과 | - kernel : linear    - kernel : poly    - kernel : rbf  -polynomial과 RBF는 데이터가 선형으로 분리되지 않을 때 유용하다. |
| 목적 | 유형에 따른 SVM-Kernel의 차이를 보기위해 사용한다. |

# Plot\_svm\_margin

|  |  |
| --- | --- |
| 제목 | SVM Margins Example |
| 결과 | - C = 1 : C값이 크면 기본적으로 모델의 데이터 분포에 대한 믿음이 별로 없고 분리선에 가까운 점만 고려한다.    - C = 0.5 : C값이 작으면 더 많은(모든 관측값)이 포함되므로 해당 영역의 모든 데이터를 사용하여 margin을 계산할 수 있다. |
| 목적 | 매개변수 C가 분리선에 미치는 영향을 파악하기 위해 사용한다. |

# Plot\_svm\_nonlinear

|  |  |
| --- | --- |
| 제목 | Non-linear SVM |
| 결과 | - XOR에 대한 예측을 보여준다. |
| 목적 | Non-linear SVM은 RBF 커널과 함께 비선형 SVC를 사용하여 이진 분류를 수행하기 위해 사용된다. |

# Plot\_svm\_regression

|  |  |
| --- | --- |
| 제목 | Support Vector Regression (SVR) using linear and non-linear kernels |
| 결과 | - 비선형 커널(RBF)을 사용한 경우 과적합이 의심될 수 있고 선형 커널(Linear)은 추세만 보여주므로 정확도가 낮을 수 있다. |
| 목적 | 선형 및 비선형 커널을 사용한 SVR을 비교하기 위해 사용한다. |

# Plot\_svm\_scale\_c

|  |  |
| --- | --- |
| 제목 | Scaling the regularization parameter for SVCs |
| 결과 | - L1 : sample의 수로 정규화 할때 성능이 좋은 것을 볼 수 있다.    - L2 : 정규화 되지 않은 경우 성능이 좋은 것을 볼 수 있다. |
| 목적 | classification을 위해 SVM을 사용할 때 정규화 매개변수(C)를 확장하는 효과를 L1, L2별로 비교하기 위해 사용한다. |

# Plot\_svm\_tie\_breaking

# Plot\_swissroll

# Plot\_t\_sne\_perlexity

# Plot\_theilsen

# Plot\_tomography\_l1\_reconstruction

# Plot\_topics\_extraction\_with\_nmf\_Ida

# Plot\_train\_error\_vs\_test\_error

|  |  |
| --- | --- |
| 제목 | Train error vs Test error |
| 결과 | - 정규화가 증가함에 따라 train에서의 성능은 감소하지만 test에서의 성능은 정규화 매개변수의 값 범위 내에서 최적이다. |
| 목적 | 알려지지 않은 data(test data)에 대한 모델의 성능이 train data에 대한 성능과 어떻게 차이가 있는지 알기 위해 사용한다. |

# Plot\_transformed\_target

|  |  |
| --- | --- |
| 제목 | Effect of transforming the targets in regression model |
| 결과 | - 좌측 : 원본 / 우측 : 변형된 target    - target 변형 후 R2(결정계수) 가 상승한 것을 볼 수 있다. |
| 목적 | 선형 회귀 모델을 학습하기 전에 target을 변환하는 것의 이점을 설명하기 위해 사용한다. |

# Plot\_tree\_regression

|  |  |
| --- | --- |
| 제목 | Decision Tree Regression |
| 결과 | - tree의 최대 깊이(max\_depth)가 너무 높게 설정되면(초록색 선) 과적합이 발생하는 것을 볼 수 있다. |
| 목적 | Decision Tree 모델을 이용한 예측(Regression)을 하기 위해 사용한다. |

# Plot\_tree\_regression\_multioutput

|  |  |
| --- | --- |
| 제목 | Multi-output Decision Tree Regression |
| 결과 | - tree의 최대 깊이(max\_depth)가 너무 높게 설정되면(초록색 선) 과적합이 발생하는 것을 볼 수 있다. |
| 목적 | Decision Tree 모델을 사용하여 다중출력 회귀(Multi-output Regression)을 하기 위해 사용한다. |

# Plot\_tweedie\_regression\_insurance\_claims

# Plot\_underfitting\_overfitting

|  |  |
| --- | --- |
| 제목 | Underfitting vs. Overfitting |
| 결과 | - Degree(차수) = 1 : 선형 함수가 적합하지 않다는 것을 알 수 있다 이것을 과소적합(underfitting) 이라고 한다.  - Degree = 4 : 실제 함수에 거의 완벽하게 근사한다.  - Degree = 15 : train data에 과적합(overfitting)되어 적합하지 않다는 것을 볼 수 있다. |
| 목적 | 과소적합과 과적합의 예시를 보여주기 위해 사용된다. |

# Plot\_unveil\_tree\_structure

|  |  |
| --- | --- |
| 제목 | Understanding the decision tree structure |
| 결과 | - children\_left[i]: 노드 i의 왼쪽 자식 ID 또는 리프 노드인 경우 -1  - children\_right[i]: 노드 i의 오른쪽 자식 ID 또는 리프 노드인 경우 -1  - feature[i]: 노드 i를 분할하는 데 사용되는 기능  - threshold[i]: 노드 i의 임계값  - n\_node\_samples[i]: 노드 i에 도달하는 훈련 샘플의 수  - impurity[i]: 노드 i의 불순물    - Decision Tree의 예시 |
| 목적 | Decision tree의 구조를 파악하기 위해 사용한다. |

# Plot\_validation\_curve

|  |  |
| --- | --- |
| 제목 | Plotting Validation Curves |
| 결과 | - gamma 값이 너무 작으면 training score와 validation score 모두 낮아 과소적합이 발생하고 너무 높으면 과적합되어 training score는 좋지만 validation score는 좋지 않음을 알 수 있다. |
| 목적 | 매개변수(gamma)에 따른 training score와 validation score를 비교하기 위해 사용한다. |

# Plot\_varimax\_fa

|  |  |
| --- | --- |
| 제목 | Factor Analysis (with rotation) to visualize patterns |
| 결과 | - column간 상관관계를 보여준다.  - sepal length, petal length and petal width가 높은 상관 관계가 있는 것을 볼 수 있다.    - 가중치의 제곱 분산을 최대화하여 찾은 varimax 회전은 두 번째 구성 요소가 sepal 너비에만 양의 하중을 가하는 구조를 나타낸다. |
| 목적 | Pattern 시각화를 위한 factor(요인) 분석을 위해 사용한다. |

# Plot\_voting\_decision\_regions

# Plot\_voting\_regressor

# Plot\_ward\_structured\_vs\_unstructued

# Plot\_weighted\_samples

# Scene\_classification

# Svm\_gui

# Tray\_food\_segmentation

# Wikipedia\_principal\_eigenvector

|  |  |
| --- | --- |
| 0 | 위키피디아에서 영향력이 큰 단어를 추출하는 프로그램 |
| 1 | Eigenvector centrality:  네트워크에서 특정 노드의 영향력을 측정함 |
| 2 | principal eigenvector  절대값이 가장 큰 eigen vector |
| 3 | DBpedia:  위키백과 프로젝트에서 만든 정보로부터 구조화된 내용을 추출하기 위한 프로젝트이다. |
| 4 | 결과 |