NCTU Pattern Recognition, Homework 4

Deadline: May 25, 23:59

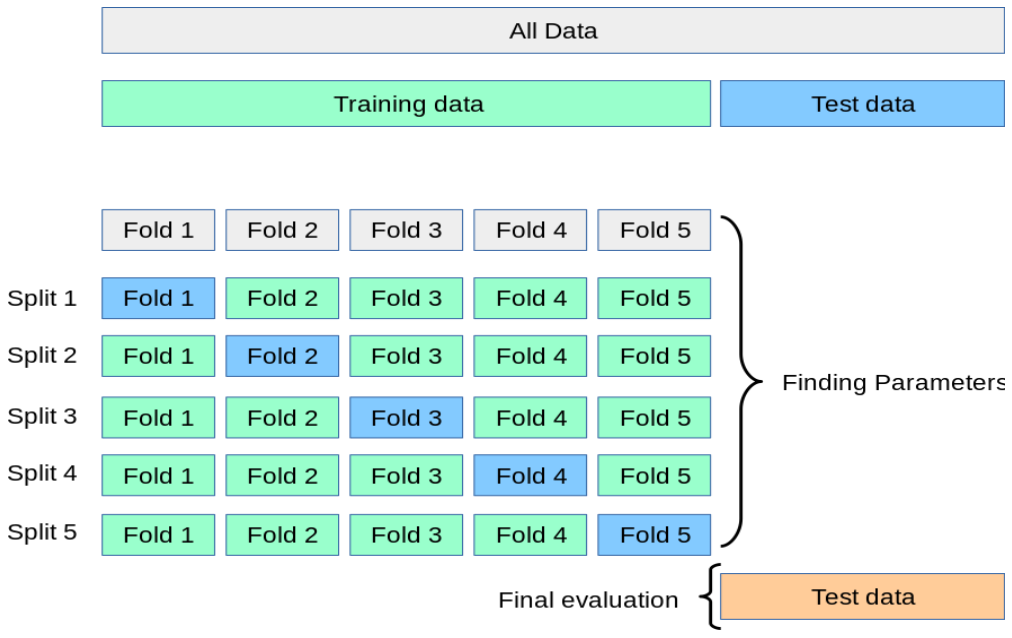
Part. 1, Coding (50%):

In this coding assignment, you need to implement the cross-validation and grid search using only NumPy, then train the SVM model from scikit-learn on the provided dataset and test the performance with testing data. Find the sample code and data on the GitHub page https://github.com/NCTU-VRDL/CS_AT0828/tree/main/HW4

Please note that only NumPy can be used to implement cross-validation and grid search. You will get no points by simply calling sklearn.model_selection.GridSearchCV.

1. (10%) K-fold data partition: Implement the K-fold cross-validation function. Your function should take K as an argument and return a list of lists (*len(list) should equal to K*), which contains K elements. Each element is a list containing two parts, the first part contains the index of all training folds (index_x_train, index_y_train), e.g., Fold 2 to Fold 5 in split 1. The second part contains the index of the validation fold, e.g., Fold 1 in split 1 (index_x_val, index_y_val)

   Note: You need to handle if the sample size is not divisible by K. Using the strategy from sklearn. The first `n_samples % n_splits` folds have size `n_samples // n_splits + 1`, other folds have size `n_samples // n_splits`, where `n_samples` is the number of samples, `n_splits` is K, `%` stands for modulus, `//` stands for integer division. See this post for more details
   Note: Each of the samples should be used **exactly once** as the validation data
   Note: Please **shuffle** your data before partition



K = 10 (from the github provided by TAs)
I shuffle the data when calling the cross_validation function, so the results may be different every execution.
I show the result with the data less than given to make the result easy to read.

- **If the sample size is divisible by K with less data**
  I used the first 20 data given to show when sample size is divisible by K and K is equal to 10.

```
Split: 1,
Training index: [ 0  1  2  3  5  6  7  8  9 10 11 13 14 15 16 17 18 19], Training length: 18
Validation index: [ 4 12], Validation length: 2
Split: 2,
Training index: [ 1  2  3  4  5  7  8  9 10 11 12 13 14 15 16 17 18 19], Training length: 18
Validation index: [0 6], Validation length: 2
Split: 3,
Training index: [ 0  1  2  3  4  6  7  8  9 10 11 12 14 15 16 17 18 19], Training length: 18
Validation index: [ 5 13], Validation length: 2
Split: 4,
Training index: [ 0  1  2  3  4  5  6  7  8 10 12 13 14 15 16 17 18 19], Training length: 18
Validation index: [ 9 11], Validation length: 2
Split: 5,
Training index: [ 0  1  3  4  5  6  7  9 10 11 12 13 14 15 16 17 18 19], Training length: 18
Validation index: [2 8], Validation length: 2
Split: 6,
Training index: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 16 17 18 19], Training length: 18
Validation index: [14 15], Validation length: 2
Split: 7,
Training index: [ 0  1  2  3  4  5  6  7  8  9 11 12 13 14 15 16 17 18], Training length: 18
Validation index: [10 19], Validation length: 2
Split: 8,
Training index: [ 0  1  2  4  5  6  7  8  9 10 11 12 13 14 15 16 17 19], Training length: 18
Validation index: [ 3 18], Validation length: 2
Split: 9,
Training index: [ 0  1  2  3  4  5  6  8  9 10 11 12 13 14 15 17 18 19], Training length: 18
Validation index: [ 7 16], Validation length: 2
Split: 10,
Training index: [ 0  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 18 19], Training length: 18
Validation index: [ 1 17], Validation length: 2
```

- **If the sample size is not divisible by K with less data**
  I used the first 23 data of given to show when sample size is not divisible by K and K is equal to 10.

```
Split: 1,
Training index: [ 0  1  2  5  6  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22], Training length: 20
Validation index: [3 4 7], Validation length: 3
Split: 2,
Training index: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 20 21], Training length: 20
Validation index: [18 19 22], Validation length: 3
Split: 3,
Training index: [ 0  1  2  3  4  5  6  7  8  9 11 14 15 16 17 18 19 20 21 22], Training length: 20
Validation index: [10 12 13], Validation length: 3
Split: 4,
Training index: [ 0  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 18 19 20 21 22], Training length: 21
Validation index: [ 1 17], Validation length: 2
Split: 5,
Training index: [ 0  1  2  3  4  5  7  8  9 10 11 12 13 14 15 16 17 18 19 20 22], Training length: 21
Validation index: [ 6 21], Validation length: 2
Split: 6,
Training index: [ 0  1  3  4  5  6  7  8  9 10 12 13 14 15 16 17 18 19 20 21 22], Training length: 21
Validation index: [ 2 11], Validation length: 2
Split: 7,
Training index: [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 17 18 19 20 21 22], Training length: 21
Validation index: [ 0 16], Validation length: 2
Split: 8,
Training index: [ 0  1  2  3  4  6  7  9 10 11 12 13 14 15 16 17 18 19 20 21 22], Training length: 21
Validation index: [5 8], Validation length: 2
Split: 9,
Training index: [ 0  1  2  3  4  5  6  7  8 10 11 12 13 14 15 16 17 18 19 21 22], Training length: 21
Validation index: [ 9 20], Validation length: 2
Split: 10,
Training index: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 16 17 18 19 20 21 22], Training length: 21
Validation index: [14 15], Validation length: 2
```

- **The result of given data**

```
Split: 1,
Training index: [  0   1   2   3   4   5   6   7   8  10  11  12  13  14  15  16  17  18
  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  40
  41  42  43  44  45  46  47  48  49  51  52  53  54  55  56  57  58  59
  60  61  62  63  66  67  68  69  70  71  72  73  74  75  76  77  78  79
  80  82  83  84  86  87  89  90  92  93  94  95  96  98  99 100 101 102
 103 104 105 106 108 109 110 111 112 113 114 115 116 117 118 119 120 121
 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 138 139 140
 141 142 143 144 145 146 147 148 149 150 151 152 155 156 157 158 159 160
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 192 193 194 195 196 198 200
 201 202 203 205 206 207 208 209 210 211 212 213 214 216 217 218 219 220
 222 223 224 225 226 227 228 229 231 232 233 234 235 236 237 238 239 240
 241 242 244 245 246 247 248 249 250 251 252 253 254 256 257 258 259 260
 261 262 263 264 265 267 268 269 270 271 272 273 275 276 277 278 279 280
 281 282 283 284 285 286 287 288 289 290 291 292 293 295 296 297 298 299
 301 304 305 306 307 308 309 310 311 314 315 316 317 318 319 320 322 324
 325 326 327 328 329 330 331 332 334 335 336 338 339 340 341 342 343 344
 345 346 347 348 349 350 351 352 353 355 356 357 358 359 360 361 362 363
 364 366 367 368 369 370 371 372 373 374 375 376 377 378 380 381 382 383
 384 385 386 387 388 389 391 392 393 394 395 396 397 398 399 400 401 402
 403 404 405 406 407 408 409 410 411 412 413 415 416 417 418 419 421 422
 423 424 425 426 427 428 429 430 431 432 433 434 435 436 438 439 440 441
 442 443 444 445 446 448 449 450 451 452 453 454 455 456 457 458 460 461
 462 464 465 466 467 468 469 470 471 472 473 474 476 477 478 479 480 481
 482 483 484 485 486 487 488 489 490 491 492 493 495 496 498 499 500 501
 502 503 504 505 506 507 508 511 512 513 515 516 517 518 519 520 521 522
 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540
 541 542 543 544 545 546 547 548 549], Training length: 495
Validation index: [  9  19  37  38  39  50  64  65  81  85  88  91  97 107 137 153 154 161
 191 197 199 204 215 221 230 243 255 266 274 294 300 302 303 312 313 321
 323 333 337 354 365 379 390 414 420 437 447 459 463 475 494 497 509 510
 514], Validation length: 55
```

```
Split: 2,
Training index: [  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  35  36
  37  38  39  40  41  42  43  44  46  47  48  49  50  51  52  53  54  55
  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73
  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91
  92  94  96  97  98 100 101 103 104 106 107 108 109 110 111 112 113 114
 115 116 117 118 120 121 122 123 124 125 126 127 129 130 131 132 133 134
 135 136 137 138 139 141 142 143 144 145 146 147 148 149 150 151 152 153
 154 155 156 157 158 159 160 161 163 164 165 166 167 168 169 170 171 172
 173 174 175 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
 192 193 194 195 196 197 198 199 200 202 204 205 206 207 208 209 210 211
 212 213 214 215 216 217 218 219 221 222 223 226 227 228 229 230 231 232
 233 234 235 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
 252 253 254 255 256 257 258 260 261 262 263 265 266 267 268 269 270 271
 272 273 274 275 277 278 279 280 281 282 283 284 285 286 287 289 290 291
 292 293 294 295 296 297 299 300 301 302 303 305 306 307 309 310 311 312
 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330
 331 332 333 334 336 337 338 339 341 342 344 345 346 347 348 349 350 351
 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369
 370 371 372 374 375 376 377 378 379 380 381 383 384 385 386 387 389 390
 391 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409
 412 413 414 415 416 417 419 420 421 422 425 426 427 428 429 430 431 432
 434 435 436 437 438 439 440 441 442 443 444 446 447 448 449 450 451 452
 454 455 457 459 460 462 463 464 465 467 468 469 470 471 472 473 474 475
 477 478 480 481 482 483 484 485 486 487 489 490 491 492 493 494 495 496
 497 498 499 500 501 502 504 505 506 507 508 509 510 511 512 513 514 515
 516 517 518 519 520 523 525 526 529 530 531 532 533 534 535 536 537 538
 539 540 541 543 545 546 547 548 549], Training length: 495
Validation index: [ 34  45  93  95  99 102 105 119 128 140 162 176 201 203 220 224 225 236
 259 264 276 288 298 304 308 335 340 343 373 382 388 392 410 411 418 423
 424 433 445 453 456 458 461 466 476 479 488 503 521 522 524 527 528 542
 544], Validation length: 55
```

```
Split: 3,
Training index: [  0   1   2   3   4   5   6   7   8   9  10  12  13  14  15  16  17  18
  19  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
  38  39  40  41  43  44  45  46  47  48  49  50  51  52  53  54  55  56
  57  58  59  60  61  62  63  64  65  66  67  69  71  72  73  75  76  77
  78  80  81  82  83  84  85  86  87  88  89  91  92  93  94  95  96  97
  98  99 100 101 102 103 104 105 107 108 109 110 111 112 114 117 118 119
 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137
 138 140 141 142 144 145 146 147 149 150 151 152 153 154 155 156 157 158
 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176
 177 179 180 181 183 184 185 186 188 189 190 191 192 193 194 195 196 197
 198 199 200 201 202 203 204 206 207 209 210 211 212 213 214 215 216 217
 218 219 220 221 222 223 224 225 227 228 229 230 231 232 236 237 238 239
 241 242 243 244 245 246 247 248 249 250 251 252 253 255 256 257 258 259
 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277
 278 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296
 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315
 316 317 318 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334
 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352
 353 354 355 357 358 359 360 362 363 365 366 367 368 369 370 371 372 373
 374 375 377 378 379 381 382 383 385 386 388 389 390 391 392 394 395 396
 398 399 400 401 402 404 405 406 407 409 410 411 413 414 415 416 417 418
 419 420 422 423 424 426 427 428 429 430 431 432 433 434 435 436 437 438
 439 440 441 442 443 444 445 446 447 448 449 450 452 453 454 455 456 457
 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475
 476 477 478 479 481 482 483 484 485 486 487 488 490 491 492 494 496 497
 498 499 500 501 502 503 504 505 506 507 508 509 510 511 513 514 515 516
 517 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 535 536
 538 539 540 542 543 544 545 548 549], Training length: 495
Validation index: [ 11  20  42  68  70  74  79  90 106 113 115 116 139 143 148 178 182 187
 205 208 226 233 234 235 240 254 279 297 319 356 361 364 376 380 384 387
 393 397 403 408 412 421 425 451 480 489 493 495 512 518 534 537 541 546
 547], Validation length: 55
```

```
Split: 4,
Training index: [  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  44  45  47  48  50  53  54  56  57  58  60  61
  62  63  64  65  66  67  68  69  70  71  72  73  74  76  77  78  79  80
  81  82  83  84  85  86  87  88  89  90  91  93  94  95  96  97  98  99
 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 119
 120 121 123 124 125 126 127 128 131 132 133 134 135 136 137 139 140 141
 142 143 144 145 146 147 148 149 151 152 153 154 155 156 157 159 160 161
 162 163 164 165 169 170 171 172 173 174 175 176 177 178 180 181 182 184
 185 186 187 188 189 190 191 192 193 194 195 197 198 199 201 202 203 204
 205 206 207 208 209 210 211 212 214 215 216 217 218 219 220 221 222 223
 224 225 226 227 228 229 230 231 232 233 234 235 236 237 239 240 241 242
 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 259 260 261
 262 263 264 265 266 267 268 269 270 272 273 274 275 276 277 278 279 280
 281 282 283 284 286 287 288 289 290 291 292 293 294 296 297 298 299 300
 301 302 303 304 305 306 307 308 309 310 311 312 313 315 317 318 319 320
 321 322 323 324 325 326 329 330 331 332 333 334 335 336 337 338 340 341
 342 343 344 345 346 347 348 350 351 352 353 354 355 356 357 358 359 360
 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
 379 380 382 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398
 399 400 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417
 418 419 420 421 422 423 424 425 427 428 429 430 431 432 433 434 435 436
 437 438 439 440 441 442 444 445 446 447 448 449 451 452 453 454 455 456
 457 458 459 460 461 462 463 464 465 466 467 468 469 470 472 474 475 476
 477 478 479 480 481 482 483 485 486 488 489 490 491 493 494 495 496 497
 498 500 503 504 505 506 507 509 510 512 514 515 516 517 518 519 520 521
 522 523 524 525 526 527 528 529 530 531 532 533 534 536 537 538 539 540
 541 542 543 544 545 546 547 548 549], Training length: 495
Validation index: [  0  43  46  49  51  52  55  59  75  92 100 118 122 129 130 138 150 158
 166 167 168 179 183 196 200 213 238 258 271 285 295 314 316 327 328 339
 349 381 383 401 426 443 450 471 473 484 487 492 499 501 502 508 511 513
 535], Validation length: 55
```

```
Split: 5,
Training index: [  0   1   3   5   6   7   8   9  10  11  12  13  15  16  17  18  19  20
  22  23  25  26  27  28  29  30  31  33  34  35  36  37  38  39  40  41
  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59
  60  61  63  64  65  66  67  68  70  71  72  73  74  75  76  77  78  79
  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97
  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115
 116 117 118 119 120 122 123 125 126 127 128 129 130 131 132 133 135 136
 137 138 139 140 142 143 144 146 147 148 149 150 151 152 153 154 155 156
 158 159 160 161 162 163 164 165 166 167 168 169 171 172 173 174 176 177
 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 194 195 196
 197 199 200 201 202 203 204 205 206 208 209 210 211 213 214 215 216 217
 219 220 221 224 225 226 227 228 230 231 232 233 234 235 236 237 238 239
 240 242 243 244 245 246 247 248 249 251 252 253 254 255 256 258 259 260
 261 262 263 264 265 266 267 268 269 270 271 273 274 275 276 278 279 281
 282 284 285 286 287 288 289 290 291 292 293 294 295 297 298 299 300 302
 303 304 305 306 308 309 310 311 312 313 314 316 317 318 319 320 321 322
 323 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
 343 344 346 347 348 349 350 351 352 353 354 355 356 358 359 360 361 362
 363 364 365 366 367 368 369 370 372 373 374 375 376 377 378 379 380 381
 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399
 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418
 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
 437 438 439 440 441 442 443 444 445 446 447 448 450 451 452 453 454 456
 457 458 459 460 461 463 466 467 468 469 470 471 473 474 475 476 479 480
 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 499
 500 501 502 503 505 506 507 508 509 510 511 512 513 514 515 516 517 518
 520 521 522 523 524 526 527 528 529 530 531 532 533 534 535 536 537 539
 540 541 542 543 544 546 547 548 549], Training length: 495
Validation index: [  2   4  14  21  24  32  62  69 121 124 134 141 145 157 170 175 193 198
 207 212 218 222 223 229 241 250 257 272 277 280 283 296 301 307 315 324
 325 345 357 371 400 449 455 462 464 465 472 477 478 498 504 519 525 538
 545], Validation length: 55
```

```
Split: 6,
Training index: [  0   2   3   4   5   6   8   9  10  12  13  14  15  16  17  18  19  20
  21  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39
  40  41  44  45  46  47  48  49  50  51  52  53  55  56  57  58  59  60
  61  63  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
  81  82  83  84  85  86  88  89  90  91  92  93  94  95  96  97  98  99
 100 101 102 103 104 105 108 109 110 111 113 114 115 116 117 118 120 121
 122 123 124 126 127 128 129 130 132 134 135 137 139 140 141 142 143 144
 145 146 147 148 149 150 151 152 154 155 156 157 158 159 160 161 162 164
 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182
 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
 201 202 203 205 206 207 208 209 210 211 212 213 214 215 217 218 219 222
 223 224 225 226 228 229 230 231 232 233 234 235 236 237 238 239 240 241
 242 244 245 247 248 249 250 251 253 254 255 256 257 258 259 260 261 262
 263 264 265 266 267 268 270 272 273 274 275 277 278 279 280 281 282 283
 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301
 302 303 304 305 306 307 308 309 310 311 312 313 314 316 317 318 319 321
 322 323 324 325 326 327 328 329 330 331 332 333 334 335 338 339 340 341
 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359
 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 376 377 378
 379 380 381 382 383 384 385 387 388 389 390 391 392 393 394 395 397 398
 399 400 401 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417
 418 419 420 421 422 423 424 425 426 427 428 429 430 431 433 434 435 436
 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 455
 456 457 461 462 463 465 466 468 469 470 471 472 473 474 475 476 477 478
 480 481 482 483 484 485 486 487 489 490 491 493 494 495 496 497 498 499
 500 501 503 504 505 506 507 508 509 510 512 513 514 515 516 517 519 520
 521 522 523 524 525 526 527 528 529 530 531 532 534 535 536 537 538 539
 540 542 543 544 545 546 547 548 549], Training length: 495
Validation index: [  1   7  11  22  42  43  54  62  64  87 106 107 112 119 125 131 133 136
 138 153 163 204 216 220 221 227 243 246 252 269 271 276 315 320 336 337
 375 386 396 402 432 454 458 459 460 464 467 479 488 492 502 511 518 533
 541], Validation length: 55
```

```
Split: 7,
Training index: [  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  18  19  20  21  22  23  25  26  27  28  30  32  33  34  35  36  37  39
  41  42  43  44  45  48  49  51  53  54  55  56  57  58  59  61  62  63
  64  65  66  67  68  69  70  71  72  73  74  75  76  78  79  80  81  82
  83  84  86  87  88  90  92  93  94  95  96  97  98 100 102 103 104 105
 106 107 108 109 110 112 113 114 115 116 117 118 119 120 121 122 123 124
 125 126 127 128 129 130 131 132 133 134 135 136 138 140 141 142 143 144
 145 146 148 149 151 152 153 154 155 156 157 159 162 163 164 165 166 167
 168 169 170 172 173 174 175 176 177 178 179 180 181 184 185 187 188 190
 191 192 193 194 195 196 197 199 200 201 203 204 205 206 207 208 209 210
 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228
 229 231 232 233 234 235 236 238 240 241 242 243 244 245 246 247 248 249
 250 251 252 253 254 255 257 258 259 260 262 265 266 267 268 269 270 271
 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289
 290 291 292 293 294 295 296 297 298 299 300 301 302 303 305 306 307 308
 309 310 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327
 328 329 330 331 332 333 335 336 337 338 339 340 341 342 343 344 345 347
 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366
 367 368 369 370 371 372 373 375 376 377 378 379 380 381 382 383 384 385
 386 387 388 389 390 391 392 393 395 396 397 398 399 400 401 402 403 404
 405 406 408 409 413 414 415 416 417 418 419 420 421 422 423 424 425 426
 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444
 445 446 447 448 449 450 452 453 454 455 456 457 458 459 460 461 462 463
 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481
 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499
 500 501 502 503 504 505 507 508 509 510 511 512 513 514 515 516 517 518
 519 520 521 522 523 525 526 527 528 529 530 531 532 533 534 535 537 538
 539 541 543 544 545 546 547 548 549], Training length: 495
Validation index: [ 24  29  31  38  40  46  47  50  52  60  77  85  89  91  99 101 111 137
 139 147 150 158 160 161 171 182 183 186 189 198 202 230 237 239 256 261
 263 264 304 311 334 346 348 374 394 407 410 411 412 451 506 524 536 540
 542], Validation length: 55
```

```
Split: 8,
Training index: [  0   1   3   4   5   6   7   8   9  10  11  12  13  16  17  18  19  20
  22  23  24  25  26  27  28  29  31  32  33  35  36  37  38  39  40  41
  42  43  44  45  46  47  48  49  50  51  52  53  54  56  58  59  60  61
  62  63  64  65  67  68  69  71  73  74  75  76  77  78  79  80  82  84
  85  86  87  88  89  91  94  95  96  97  98  99 100 101 102 103 104 105
 106 107 108 109 110 111 112 113 115 116 117 118 119 120 121 122 123 124
 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142
 144 145 146 147 148 149 150 152 153 154 155 156 157 158 159 160 161 162
 163 164 165 166 167 168 169 170 171 172 173 174 175 176 178 179 180 182
 183 184 186 187 188 189 190 191 192 193 195 196 197 198 199 200 202 203
 204 205 206 207 208 209 210 211 212 213 215 216 217 218 219 220 221 222
 223 225 226 227 228 229 230 231 232 233 234 237 238 239 240 241 242 243
 244 245 246 247 248 249 250 251 252 253 254 256 257 258 259 260 261 262
 263 264 265 266 267 268 269 270 271 272 274 275 276 277 278 279 280 282
 284 285 286 287 288 289 290 291 292 293 294 296 297 298 299 300 301 302
 304 305 307 308 310 311 312 313 314 315 316 317 318 319 320 321 322 324
 326 327 328 329 330 331 332 333 334 335 336 337 338 339 341 342 343 344
 345 346 347 348 349 350 351 352 353 354 355 356 357 358 360 361 362 363
 364 365 366 367 368 369 370 371 372 373 374 375 376 377 379 381 383 384
 385 386 387 388 389 391 392 393 394 395 396 398 399 400 401 402 403 404
 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 421 422 423
 424 425 426 427 429 430 432 433 434 435 436 437 438 439 440 442 443 444
 445 446 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463
 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481
 482 483 485 486 487 488 490 491 492 493 494 495 496 498 499 500 501 502
 503 504 505 506 507 509 510 511 512 513 514 515 516 517 518 519 520 521
 522 523 524 525 526 527 528 529 530 531 532 533 534 536 537 538 539 540
 541 542 543 544 545 546 547 548 549], Training length: 495
Validation index: [  2  14  15  21  30  34  55  57  66  70  72  81  83  90  92  93 114 143
 151 177 181 185 194 201 214 224 235 236 255 273 281 283 295 303 306 309
 323 325 340 359 378 380 382 390 397 420 428 431 441 447 484 489 497 508
 535], Validation length: 55
```

```
Split: 9,
Training index: [  1   2   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
  38  39  40  41  42  43  44  45  46  47  48  50  51  52  53  54  55  56
  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  76
  77  78  79  80  81  82  83  84  85  87  88  89  90  91  92  93  94  95
  96  97  98  99 101 102 104 105 106 107 108 109 110 111 112 114 115 116
 117 118 119 121 122 123 125 126 127 128 129 130 131 132 133 134 135 136
 137 138 139 140 141 142 143 145 146 147 148 149 150 151 152 153 154 156
 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 175
 176 177 178 179 181 182 183 184 185 186 187 188 189 190 191 193 194 195
 196 197 198 199 200 201 202 203 204 205 206 209 210 211 212 213 214 215
 216 217 218 219 220 221 222 224 225 226 227 228 229 230 231 232 234 235
 236 237 238 239 240 241 242 243 244 246 247 249 251 252 253 254 255 256
 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274
 275 276 277 278 279 280 281 282 283 284 285 286 289 290 292 293 294 295
 297 299 300 301 302 303 304 305 306 307 308 309 311 312 313 314 315 316
 317 318 319 320 321 322 323 324 325 327 328 329 330 331 333 334 336 337
 339 340 341 342 344 345 346 347 348 349 350 352 353 354 355 356 357 358
 359 360 361 362 363 364 366 367 368 369 370 371 372 373 374 375 376 377
 378 379 380 381 382 383 384 385 386 387 389 390 391 392 393 394 395 396
 397 398 400 401 402 403 406 407 408 409 410 411 412 413 414 415 416 417
 418 420 421 422 423 424 425 426 427 428 430 431 432 433 434 435 436 437
 438 440 441 443 444 445 446 447 451 452 453 454 455 456 457 458 459 460
 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478
 479 480 481 482 483 484 485 486 487 488 489 490 492 493 494 496 497 498
 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 515 517 518
 519 520 523 524 525 526 527 528 529 530 531 533 534 535 536 537 539 540
 541 542 543 544 545 546 547 548 549], Training length: 495
Validation index: [  0   3  49  74  75  86 100 103 113 120 124 144 155 174 180 192 207 208
 223 233 245 248 250 287 288 291 296 298 310 326 332 335 338 343 351 365
 388 399 404 405 419 429 439 442 448 449 450 491 495 514 516 521 522 532
 538], Validation length: 55
```

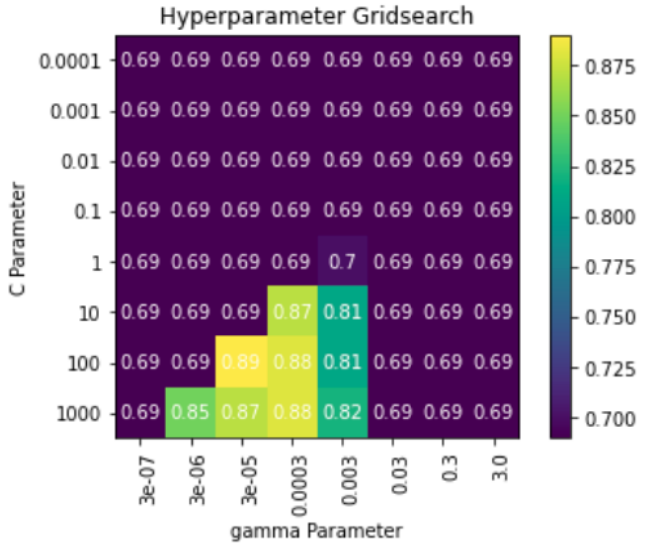
```
Split: 10,
Training index: [  0   1   2   3   4   5   6   7   8   9  10  11  12  14  15  19  20  21
  22  23  24  25  26  27  29  30  31  32  33  34  35  36  37  38  39  40
  41  42  43  44  46  47  48  49  50  52  53  54  55  56  57  58  59  60
  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78
  79  81  82  83  84  85  86  87  89  90  91  92  93  94  95  97  98  99
 100 101 102 103 104 106 107 108 109 110 111 112 113 114 115 116 117 118
 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 135 136 137
 138 139 140 141 143 144 145 146 147 148 150 151 152 153 154 155 158 160
 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178
 179 180 181 182 183 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199 201 202 203 204 205 206 207 208 209 210 211 212 214 215 216 218
 219 220 221 222 223 224 227 228 229 230 231 232 233 234 235 236 237 238
 239 240 241 242 243 245 246 247 248 250 252 253 254 255 256 257 258 260
 261 262 263 264 265 266 268 269 270 271 272 273 274 275 276 277 279 280
 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298
 299 300 302 303 304 305 306 307 309 310 311 312 313 314 315 316 317 318
 319 320 321 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337
 338 339 340 341 342 343 344 345 346 348 349 350 351 352 353 354 355 356
 357 359 360 361 362 363 364 365 366 367 368 369 370 371 373 374 375 376
 377 378 380 381 382 383 384 386 387 388 389 390 391 392 393 394 395 396
 397 398 399 401 402 403 404 405 407 408 409 410 411 412 413 416 417 418
 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
 437 438 439 440 441 442 443 445 446 447 448 449 450 451 452 454 455 456
 457 458 459 460 462 464 465 466 467 468 469 470 471 474 475 476 478 479
 481 484 485 486 487 488 489 491 492 493 494 495 496 497 498 499 500 502
 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520
 521 522 523 524 525 526 527 529 530 531 532 533 534 535 536 537 538 539
 540 541 542 543 544 545 546 548 549], Training length: 495
Validation index: [ 13  16  17  18  28  45  51  80  88  96 105 134 142 149 156 157 159 184
 200 213 217 225 226 244 249 251 259 267 278 301 308 322 347 358 372 379
 385 400 406 414 415 444 453 461 463 472 473 477 480 482 483 490 501 528
 547], Validation length: 55
```

2. (20%) Grid Search & Cross-validation: using sklearn.svm.SVC to train a classifier on the provided train set and conduct the grid search of "C" and "gamma," "kernel'='rbf' to find the best hyperparameters by cross-validation. Print the best hyperparameters you found.

   Note: We suggest using K=5

```
best_C = 100, best_gamma = 3e-05
```

3. (10%) Plot the grid search results of your SVM. The x and y represent “gamma” and “C” hyperparameters, respectively. And the color represents the average score of validation folds.



4. (10%) Train your SVM model by the best hyperparameters you found from question 2 on the whole training data and evaluate the performance on the test set.

Accuracy	Your scores
acc > 0.9	10points
0.85 <= acc <= 0.9	5 points
acc < 0.85	0 points

```
Accuracy score:  0.9166666666666666
```

Part. 2, Questions (50%):

1. (10%) Given a valid kernel $k_1(x, x')$, prove that the following proposed functions are or are not valid kernels.
   a. $k(x, x') = (k_1(x, x'))^2 + (k_1(x, x') + 1)^2$
   b. $k(x, x') = (k_1(x, x'))^2 + \exp(\|x\|^2) * \exp(\|x'\|^2)$

1. (a) $\because k(x,x') = g(k_1(x,x'))$ is valid kernel, where $g(\cdot)$ is polynomial with nonnegative coefficients

$\therefore (k_1(x,x'))^2$ is valid kernel

$\because$ according to teaching material, $k(x,x') = (x^T x' + c)^M$ is valid kernel

$\therefore (k_1(x,x')+1)^2$ is valid kernel

$\because k(x,x') = k_1(x,x') + k_2(x,x')$ is valid kernel

$\therefore k(x,x') = (k_1(x,x'))^2 + (k_1(x,x')+1)^2$ is valid kernel #

(b) $\exp(\|x\|^2) \times \exp(\|x'\|^2) = \exp(\|x\|^2 + \|x'\|^2) = \exp(\|x-x'\|^2 + 2\|x\|\|x'\|)$

$\because k(x,x') = \sum_{r=0}^{\infty} \phi_r(x)\phi_r(x') = \sum_{r=0}^{\infty} \frac{x^r}{\sqrt{r!}} \frac{(x')^r}{\sqrt{r!}} = \sum_{r=0}^{\infty} \frac{(xx')^r}{r!} = e^{xx'}$ is valid kernel

$\therefore \exp(2\|x\|\|x'\|)$ is valid kernel

$\because \exp(\|x-x'\|^2)$

$\because$ according to teaching material from CMU, $\exp(r\|x-x'\|^2)$ for some $r>0$ is not valid kernel

$\therefore k(x,x') = (k_1(x,x'))^2 + \exp(\|x\|^2) \times \exp(\|x'\|^2)$ is not valid kernel #

2. (10%) Show that the kernel matrix $\mathbf{K} = [k(\mathbf{x}_n, \mathbf{x}_m)]_{nm}$ should be positive semidefinite is the necessary and sufficient condition for $k(\mathbf{x}, \mathbf{x}')$ to be a valid kernel.

2、 $K(x_1, x_2)$ 能寫成 $\phi(x_1)^T \phi(x_2)$ 的充要條件：① $k(x_1, x_2) = k(x_2, x_1)$ (交換性)

② $\forall c_i, x_i \ (i = 1 \sim N)$, (半正定性)

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(x_i, x_j) \geq 0$$

implying that its eigenvalues are non-negative

for any chosen continuous kernel function we can prove via closure properties its positive semidefinites on the entire input space and then apply the Mercer theorem. The kernel evaluated at any set of test points yields a positive semidefinite Gram matrix.

1. closure under sum
2. closure under product
3. closure under tensor product
4. closure under concatenation of function

3. (10%) Consider the dual formulation of the least-squares linear regression problem given on page 6 in the ppt of Kernel Methods. Show that the solution for the components $\mathbf{a_n}$ of the vector $\mathbf{a}$ can be expressed as a linear combination of the elements of the vector $\boldsymbol{\varphi}(\mathbf{x_n})$. Denoting these coefficients by the vector $\mathbf{w}$, show that the dual of the dual formulation is given by the original representation in terms of the parameter vector $\mathbf{w}$.

3. define the Gram matrix $K = \Phi\Phi^T$, which is an $N \times N$ symmetric matrix with elements

$K_{nm} = \phi(x_n)^T\phi(x_m) = k(x_n, x_m)$

then sum-of-square error function can be written as

$J(a) = \frac{1}{2}a^TKKa - a^TKt + \frac{1}{2}t^Tt + \frac{\lambda}{2}a^TKa$, where $t = (t_1, \cdots, t_N)^T$

then decompose $a = a_{\parallel} + a_{\perp}$ where $a_{\parallel}^T a_{\perp} = 0$ and $Ka_{\perp} = 0$

thus the value of $a_{\perp}$ is not determined by $J(a)$. We can remove the ambiguity by setting $a_{\perp} = 0$, or equivalently by adding a regularizer term $\frac{\epsilon}{2}a_{\perp}^Ta_{\perp}$ to $J(a)$ where ϵ is a small positive constant

Then $a = a_{\parallel}$ where $a_{\parallel}$ lies in the span of $K = \Phi\Phi^T$ and hence can be written as a linear combination of the columns of Φ, so that in vector notation $a = \Phi u$

substituting $a = \Phi u$ into $J(a)$, we obtain $J(u) = \frac{1}{2}(\Phi\Phi^T\Phi u - t)^T(\Phi\Phi^T\Phi u - t) + \frac{\lambda}{2}u^T\Phi^T\Phi\Phi^T\Phi u$

since the matrix $\Phi^T\Phi$ has full rank, we can define an equivalent parametrization given by

$w = \Phi^T\Phi u$

and substituting this into $J(u)$, we recover the original regularized error function $J(w)$

$J(w) = \frac{1}{2}\sum_{n=1}^{N}\{w^T\phi(x_n) - t_n\}^2 + \frac{\lambda}{2}w^Tw$

4. (10%) Prove that the Gaussian kernel defined by (eq 1) is valid and show the function **φ(x)**, where x ∈ R^1.

(eq1) $$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2/2\sigma^2\right) = \boldsymbol{\phi}(x)^{\mathrm{T}}\boldsymbol{\phi}(x')$$

4. expanding the square $\|x-x'\|^2 = x^Tx + (x')^Tx' - 2x^Tx'$

to give $k(x,x') = \exp(-x^Tx/2\sigma^2)\exp(x^Tx'/\sigma^2)\exp(-(x')^Tx'/2\sigma^2)$

and then making use of $\begin{cases} k(x,x') = f(x)\,k_1(x,x')\,f(x') \\ k(x,x') = \exp(k_1(x,x')) \end{cases}$, together with the validity of the linear kernel $k(x,x') = x^Tx'$, then the Gaussian kernel is valid

$\phi(x)^T = \exp\left((-x^Tx + 2x^T)/2\sigma^2\right)$

$\phi(x') = \exp\left((-(x')^Tx' + 2x')/2\sigma^2\right)$

5. (10%) Consider the optimization problem

minimize $(x - 2)^2$

subject to $(x+3)(x-1) \leqq 2$

State the dual problem.

5.

$(x+3)(x-1) \leq 2 \Rightarrow x^2 + 2x - 5 \leq 0$

$\Rightarrow \left[x-(-1+\sqrt{6})\right]\left[x-(-1-\sqrt{6})\right] \leq 0$

$L(x,\lambda) = (1+\lambda)x^2 - 6\lambda x + (1-5\lambda)$

the dual problem is $\begin{cases} \text{maximize} & -9\lambda^2/(1+\lambda) + 1 - 5\lambda \\ \text{subject to} & \lambda \geq 0 \end{cases}$