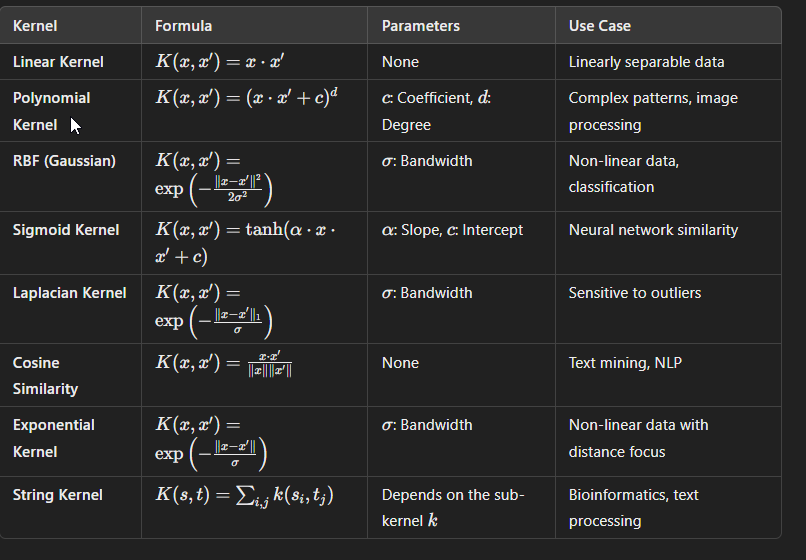






Kernel Functions



Coding

|  |  |
| --- | --- |
|  | BLANK BOX SOLUTION (CANDIDATE CODE):  # Show an example entry for 'sectionmodels' to identify field names  db['sectionmodels'].find\_one()  # Define a class named Sectionmodels class Sectionmodels(Document):  index = IntField()  value = IntField()  description = StringField()  name = StringField()  questions = ListField(StringField())  question = StringField()  v = IntField(db\_field='\_\_v') |
|  | One solve is to use the following code: import numpy as np  data = np.genfromtxt("data/counts.tsv") counts = data[:,1] median = np.median(counts) print(median) |
|  | One way to determine the correct answer is to use the following loop and review the scatter plots that were created.  for legend, column in columns\_to\_plot.items():  plt.scatter(range(len(this\_drive)), this\_drive[:,column])  plt.title(legend)  plt.show() |
|  | Here is one solution:  print(sorted\_list[:30])  Find that 'film' is ranked 19. |
|  | Here is one solution:  figure, axes = plt.subplots(16,8, figsize=(8,8), dpi=300) for i, ax in enumerate(axes.flatten()):   ax.imshow(x\_train[i], cmap="gray")  ax.set\_axis\_off()  ax.set\_title(y\_train[i])  plt.tight\_layout() plt.show()  Find the answer in the output matrix. |
|  | Here are a couple of solutions:  Solution 1:  quiz\_roll = [] flatlist = [] total = 0 i=0 avg = 0.0  for key, subdict in students.items():  for k, v in subdict.items():  if k == 'Quiz1' or k == 'Quiz2' or k == 'Quiz3':  temp = students[key][k]  quiz\_roll.append(temp)  i += 1  flatlist=[element for sublist in quiz\_roll for element in sublist] total = sum(flatlist) avg = round(total / i, 2)  print(total,i,avg)  Solution 2  tab = 0 i=0 avg = 0.0  for key, subdict in students.items():  for k, v in subdict.items():  if k == 'Quiz1' or k == 'Quiz2' or k == 'Quiz3':  temp = students[key][k]  tab += int(temp[0])  i += 1  avg = round(tab/ i, 2)  print(tab,i,avg) |
|  | One way to solve this is:  filetype\_mean = {} filetype\_std = {} for filetype in files:  print(filetype)  for filename in files[filetype]:  (mean,std) = get\_file\_stats(f'../data/giac\_images/{filetype}/{filename}')  print(f'\t{filename}:\t{mean}\t{std}')  Note: Mean is displayed first followed by Standard Deviation (std) in the output. This is as per the order used in the initialization code |
|  |  |
|  |  |
|  |  |
|  |  |