fall2022_hw1

October 9, 2022

1 CS171-EE142 - Fall 2022 - Homework 1

2 Due: Thursday, October 13, 2022 @ 11:59pm

2.0.1 Maximum points: 75 pts

2.1 Submit your solution to Gradescope:

1. Submit a single PDF to **HW1**
2. Submit your jupyter notebook to **HW1-code**

See the additional submission instructions at the end of this notebook

2.2 Enter your information below:

2.2.1 Your Name (submitter): Yash Aggarwal

2.2.2 Your student ID (submitter): 862333037

By submitting this notebook, I assert that the work below is my own work, completed for this course. Except where explicitly cited, none of the portions of this notebook are duplicated from anyone else's work or my own previous work.

2.3 Academic Integrity

Each assignment should be done individually. You may discuss general approaches with other students in the class, and ask questions to the TAs, but you must only submit work that is yours . If you receive help by any external sources (other than the TA and the instructor), you must properly credit those sources, and if the help is significant, the appropriate grade reduction will be applied. If you fail to do so, the instructor and the TAs are obligated to take the appropriate actions outlined at http://conduct.ucr.edu/policies/academicintegrity.html . Please read carefully the UCR academic integrity policies included in the link.

3 Overview

In this assignment you will explore some basic computations on data and build a nearest neighbor classifier.

For this assignment we will use the functionality of Pandas (https://pandas.pydata.org/), Matplotlib (https://matplotlib.org/), and Numpy (http://www.numpy.org/). You may also find Seaborn (https://seaborn.pydata.org/) useful for some data visualization.

If you are asked to **implement** a particular functionality, you should **not** use an existing implementation from the libraries above (or some other library that you may find). When in doubt, please ask.

Before you start, make sure you have installed all those packages in your local Jupyter instance

3.1 Read *all* cells carefully and answer all parts (both text and missing code)

You will complete all the code marked `TODO` and answer descriptive/derivation questions

[1]:

```
# Standard library imports.
import random as rand
from collections import Counter

# Related third party imports.
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

# Local application/library specific imports.
# import here if you write .py script
```

4 Getting real data

In this assignment we are going to use the penguins dataset.

More info

This dataset can also be easily imported from seaborn https://github.com/mwaskom/seaborn-data

[2]:

```
data = sns.load_dataset("penguins") # load penguins dataset from seaborn
data = data.dropna() # drop samples with missing values (NaN)
data
```

```
[2]:     species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
     0    Adelie  Torgersen            39.1           18.7              181.0
     1    Adelie  Torgersen            39.5           17.4              186.0
     2    Adelie  Torgersen            40.3           18.0              195.0
     4    Adelie  Torgersen            36.7           19.3              193.0
     5    Adelie  Torgersen            39.3           20.6              190.0
     ..      …          …               …              …                  …
     338  Gentoo     Biscoe            47.2           13.7              214.0
     340  Gentoo     Biscoe            46.8           14.3              215.0
     341  Gentoo     Biscoe            50.4           15.7              222.0
     342  Gentoo     Biscoe            45.2           14.8              212.0
     343  Gentoo     Biscoe            49.9           16.1              213.0

          body_mass_g     sex
     0         3750.0    Male
     1         3800.0  Female
     2         3250.0  Female
     4         3450.0  Female
     5         3650.0    Male
     ..           …      …
     338       4925.0  Female
     340       4850.0  Female
     341       5750.0    Male
     342       5200.0  Female
     343       5400.0    Male

     [333 rows x 7 columns]
```

This data has 344 samples and we keep the 333 samples without missing values

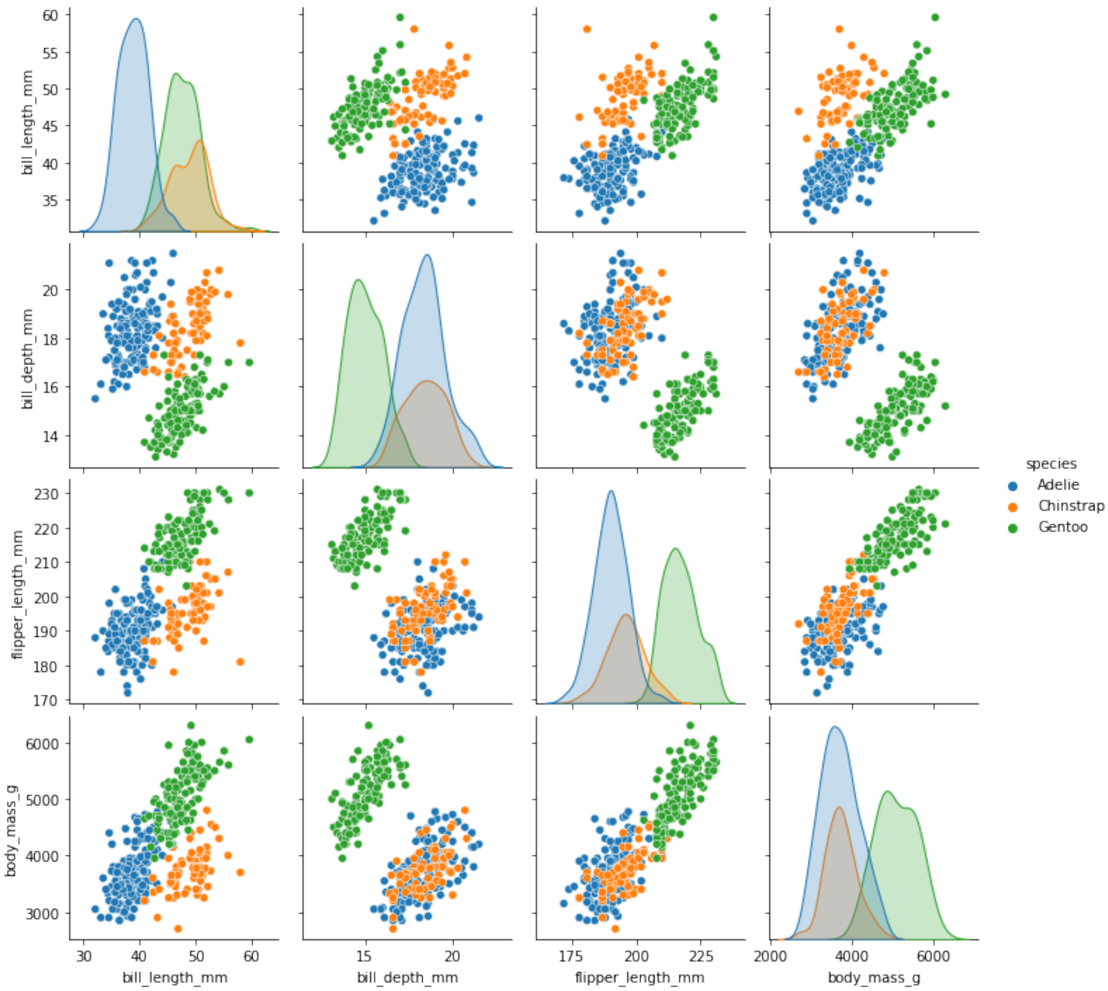
We will treat `'species'` as the label for each data sample and remaining 6 entries as the features `['island', 'bill_length_mm','bill_depth_mm','flipper_length_mm', 'body_mass_g', 'sex']`.

The `'species'` label has three possible values `['Adelie', 'Chinstrap', 'Gentoo']`.

Note *that the arrangement of data samples is different from the notation we used in the class. In the class, data samples are given as column vectors. Here data samples are given as row vectors.* ***`data`*** *has 333 sample vectors, each of length 6 and stored as rows of* ***`data`****.*

In this homework, we will only focus on the 4 **numerical features**. We can visualize pair-wise relations among 4 numerical features `['bill_length_mm','bill_depth_mm','flipper_length_mm', 'body_mass_g']` in the data using scatterplot of all pairs of features and color the points by class label.

```
[3]: fig = sns.pairplot(data, hue="species")
```



4.1 Question 1: Basic data analysis [25 pts]

4.2 Question 1a: Counting and simple statistics [5]

1. Calculate and print the min, max, and mean values of each of the four features in the entire dataset [1]
2. Count and print the number of samples that belong to each of the three categories [1]
3. Calculate and print the mean of each feature for each label as a 3 × 4 table [3]

*Note: For **mean** values, keep 1 digit after the decimal point when printing*

[4]:

```
# TODO: complete the code blocks below

print("\n1. print min, max, mean values of 4 features:\n")
features = ['bill_length_mm','bill_depth_mm','flipper_length_mm', 'body_mass_g']
```

```
for feature in features:
    # TODO
    print(f"(min, max, mean) of {feature}: ({min(data[feature])},␣
 ↪{max(data[feature])}, {round(data[feature].mean(),1)})")
    pass

print("\n2. print number of samples per category:\n")

cnt = data['species'].value_counts()
categories = ['Adelie', 'Chinstrap', 'Gentoo']
for category in categories:
    # TODO
    print(f"number of samples in {category}: {cnt[category]}")
    pass


print("\n3. print a 3x4 table of mean value of feature for each label:\n")
# TODO: create a table (as a list) with mean values
# inbuilt method # data.groupby(by='species').mean()
# sum of all the features in respective positon in 2-D list

cnt = data['species'].value_counts()
table = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
for index, row in data.iterrows():
    r = 0
    c = 0
    if row['species'] == 'Adelie':
        r = 0
    elif row['species'] == 'Chinstrap':
        r = 1
    else:
        r = 2

    table[r][0] +=  row['bill_length_mm']
    table[r][1] +=  row['bill_depth_mm']
    table[r][2] +=  row['flipper_length_mm']
    table[r][3] +=  row['body_mass_g']

# Divide the matrix with respective count to get average.
for r in range(len(table)):
    for c in range(len(table[r])):
        if r == 0:
            table[r][c] /= cnt['Adelie']
        elif r ==1:
            table[r][c] /= cnt['Chinstrap']
        else:
            table[r][c] /= cnt['Gentoo']
```

```
pd.set_option("display.precision", 1)
panda_table = pd.DataFrame(data = table, index = categories, columns = features)
panda_table
```

```
1. print min, max, mean values of 4 features:

(min, max, mean) of bill_length_mm: (32.1, 59.6, 44.0)
(min, max, mean) of bill_depth_mm: (13.1, 21.5, 17.2)
(min, max, mean) of flipper_length_mm: (172.0, 231.0, 201.0)
(min, max, mean) of body_mass_g: (2700.0, 6300.0, 4207.1)

2. print number of samples per category:

number of samples in Adelie: 146
number of samples in Chinstrap: 68
number of samples in Gentoo: 119

3. print a 3x4 table of mean value of feature for each label:
```

[4]:

```
           bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g
Adelie               38.8           18.3              190.1       3706.2
Chinstrap            48.8           18.4              195.8       3733.1
Gentoo               47.6           15.0              217.2       5092.4
```

4.2.1 Question 1b: Implement the ℓ_p distance function [10]

1. Write code that implements the Lp distance function between two data points as we saw it in class [7] \ ℓ_p distance between two vectors $\mathbf{x}_i, \mathbf{x}_j$ in $\mathbb{R}^d$ can be written as

$$\|\mathbf{x}_i - \mathbf{x}_j\|_p = \left(\sum_{k=1}^{d} |\mathbf{x}_i(k) - \mathbf{x}_j(k)|^p\right)^{1/p}.$$

   You should implement this function yourself. Do not use any built-in function to compute distance or norm.

2. Verify that it is correct by comparing it for p=1 and p=2 against an existing implementation in Numpy for the two selected data points below. Note that the difference of the distances may not be exactly 0 due to numerical precision issues. [3]

[5]:

```
# TODO: complete the function below
def distance(x,y,p=2):
    """calculates Lp distance between point x and point y
    Args:
        x (np.ndarray): datapoint x
        y (np.ndarray): datapoint y
        p (int): order of Lp norm
    """
    tot_sum = 0
    for i in range(len(x)):
        diff = x[i] - y[i]
        diff = abs(diff)
        diff = pow(diff,p)
        tot_sum += diff

    tot_sum = pow(tot_sum,1/p)
    return tot_sum



# TODO: pick 2 samples from the dataset (only keep the numerical features)

data2 = data[['bill_length_mm','bill_depth_mm','flipper_length_mm',␣
 ↪'body_mass_g']]

import random
num1 = random.randrange(0,333,5)
num2 = random.randrange(0,333,5)
data_point1 = data2.iloc[num1]
data_point2 = data2.iloc[num2]
p = 2

# calculate the distance and compare with numpy built-in function
dist = distance(data_point1, data_point2, p)
print(f"my distance: {dist}")

# verify using numpy built-in function
dist_np = np.linalg.norm(data_point1 - data_point2, p)
print(f"np distance: {dist_np}")
```

```
my distance: 1200.7065461635495
np distance: 1200.7065461635495
```

4.2.2 Question 1c: Compute the distance matrix between all data points [10]

1. Compute an $N \times N$ distance matrix between all data points (where $N = 333$ is the number of data points) [3]
2. Plot the above matrix and include a colorbar. Add title and x/y labels to the plot.[3]
3. What is the minimum number of distance computations that you can do in order to populate every value of this matrix? (note: it is OK if in the first two questions you do all the N^2 computations) [2]

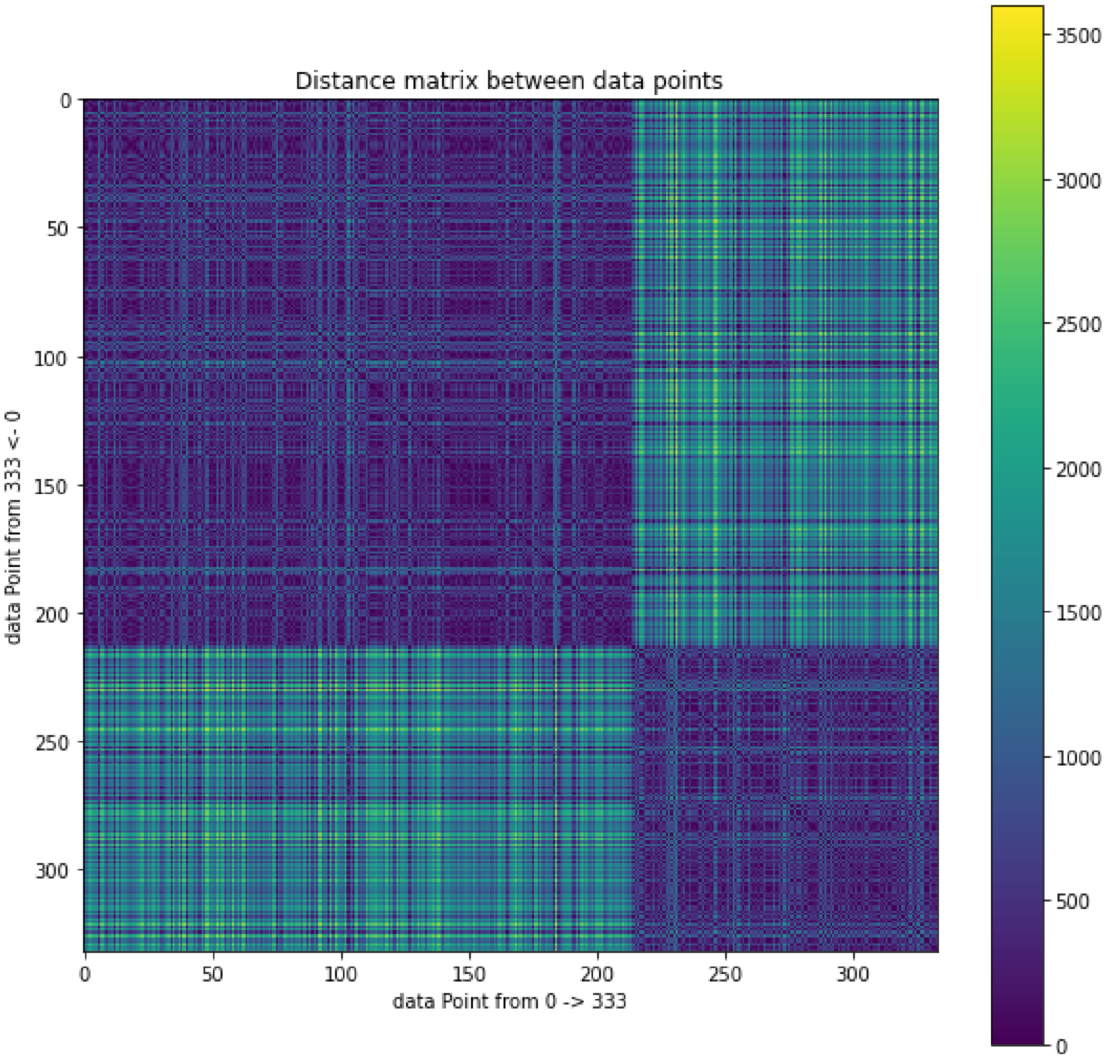
4. Note that the data points in your dataset are sorted by class. What do you observe in the distance matrix? [2]

[6]:

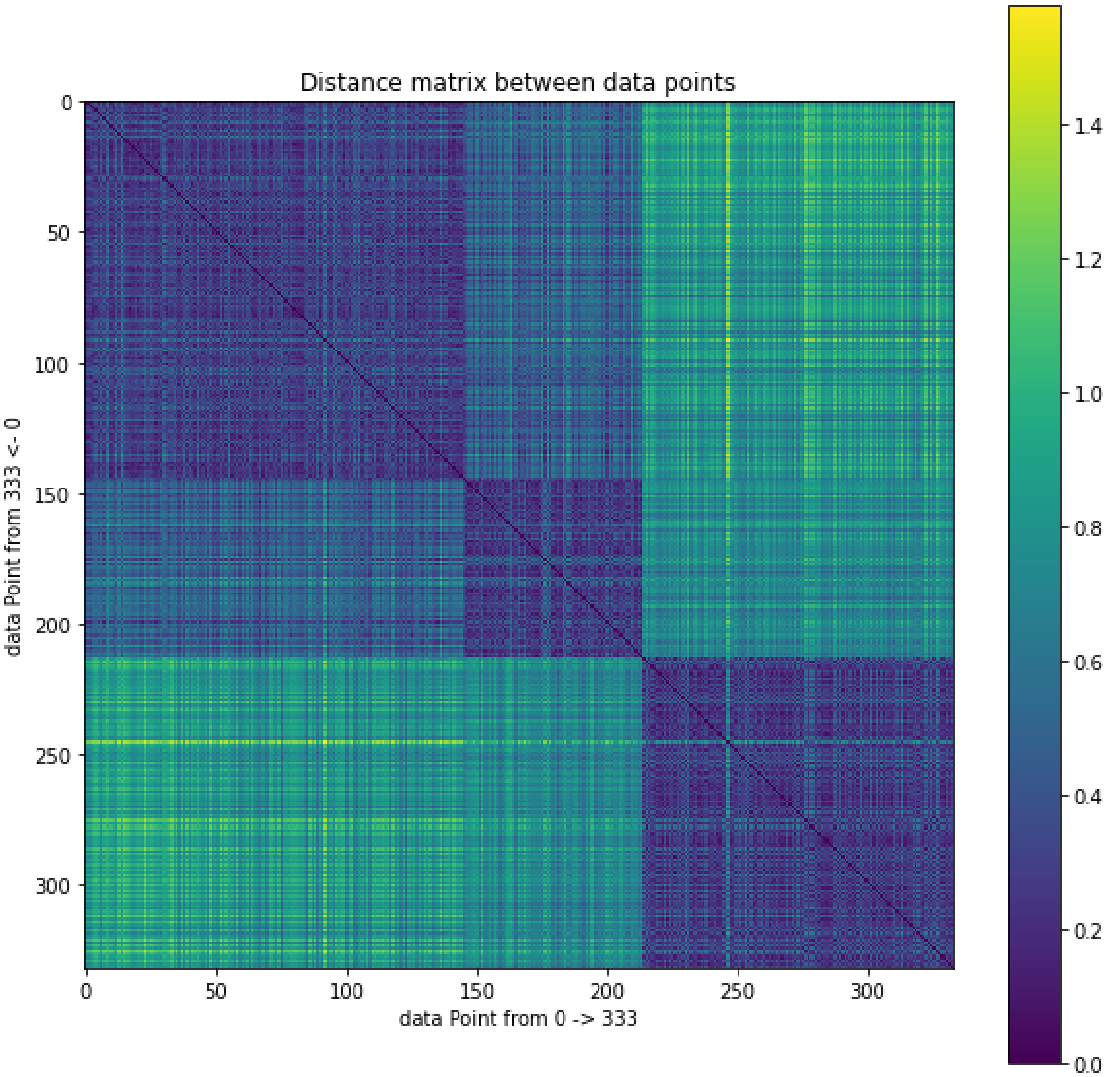
```python
# TODO: complete the function below

print("\n1. Compute distance matrix\n")

# create a subset of data such that it only has numerical values
data3 = data.drop(axis = 1, labels = ['species','island','sex'])

dist_matrix = np.zeros((333, 333))
for i in range(0,333):
    for j in range (0, 333):
        dist_matrix[i][j] = distance(data3.iloc[i], data3.iloc[j], 2)

print (dist_matrix)

print("\n2. Plot matrix with a colorbar.\n")
plt.figure(figsize = (10,10))
plt.imshow(dist_matrix, interpolation='nearest')
plt.title('Distance matrix between data points')
plt.xlabel('data Point from 0 -> 333')
plt.ylabel('data Point from 333 <- 0')
plt.colorbar()
plt.show()


# normalize the dataset

data3_scaled = data3.copy()
col_names_list = data3.columns.values.tolist()

for column in col_names_list:
    col_max = data3_scaled[column].max()
    col_min = data3_scaled[column].min()
    data3_scaled[column] = (data3_scaled[column] - col_min) / (col_max - col_min)


dist_matrix_scaled = np.zeros((333, 333))
for i in range(0,333):
    for j in range (0, 333):
        dist_matrix_scaled[i][j] = distance(data3_scaled.iloc[i], data3_scaled.iloc[j], 2)

# print (dist_matrix)
```

```python
print("\n2. Plot matrix with a colorbar (normalized data).\n")
plt.figure(figsize = (10,10))
plt.imshow(dist_matrix_scaled, interpolation='nearest')
plt.title('Distance matrix between data points')
plt.xlabel('data Point from 0 -> 333')
plt.ylabel('data Point from 333 <- 0')
plt.colorbar()
plt.show()
```

```
1. Compute distance matrix

[[   0.           50.26778292  500.19789084 … 2000.45437089
  1450.34941307 1650.34766034]
 [  50.26778292    0.          550.0745404  … 1950.36347894
  1400.25542313 1600.26211915]
 [ 500.19789084  550.0745404     0.         … 2500.16725441
  1950.08288285 2150.09761871]
 …
 [2000.45437089 1950.36347894 2500.16725441 …    0.
   550.11621499  350.11628068]
 [1450.34941307 1400.25542313 1950.08288285 …  550.11621499
     0.          200.06194041]
 [1650.34766034 1600.26211915 2150.09761871 …  350.11628068
   200.06194041    0.        ]]

2. Plot matrix with a colorbar.
```



2. Plot matrix with a colorbar (normalized data).



Write your answer here:

3. As the distance of a point from itself is 0, we can simply put the values in diagonal as 0 instead of computing the distance. This will reduce the calculations by or N. As the matrix will be symmetrical about diagonal, i.e. the distance from i,j is same as j,i we can reduce the calculations by half. So the maximum calculations will be (N*N - N / 2) or N(N-1)/2

4. As the dataset is sorted by species, we can see in the distance matrix that in some clusters the distance is close to 0. This is because the characterstics of different datapoints of same species are similar resulting in smaller differences. Also, this distance increases as we move towards other species.

4.3 Question 2: K-Nearest Neighbors Classifier [50 pts]

The K-Nearest Neighbors Classifier is one of the most popular instance-based (and in general) classification models. In this question, we will implement our own version and test in different scenarios.

4.3.1 Question 2a: Implement the K-NN Classifier [20]

For the implementation, your function should have the format:

```
def knnclassify(test_data,training_data, training_labels, K=1):
```

where `test_data` contains test data points, `training_data` contains training data points, `training_labels` holds the training labels, and `K` is the number of neighbors.

The output of this function should be `pred_labels` which contains the predicted label for each test data point (it should, therefore, have the same number of rows as `test_data`).

[7]:

```
# prepare datasets, convert to numpy array, X - data values, Y - labels
X = data[['bill_length_mm','bill_depth_mm','flipper_length_mm', 'body_mass_g']].
 ↪values
all_labels = data['species'].values
unique_labels = np.unique(all_labels)
new_labels = np.zeros(len(all_labels))
for i in range(0,len(unique_labels)):
  new_labels[all_labels == unique_labels[i]] = i
Y = new_labels
```

[8]:

```
# TODO: complete the function below
def knnclassify(test_data, training_data, training_labels, K=1):
    """KNN classifier
    Args:
    test_data (numpy.ndarray): Test data points.
    training_data (numpy.ndarray): Training data points.
    training_labels (numpy.ndarray): Training labels.
    K (int): The number of neighbors.

    Returns:
    pred_labels: contains the predicted label for each test data point, have␣
 ↪the same number of rows as 'test_data'
    """

    # use the existing distance function to compute distances.
    pred_labels = np.zeros(test_data.shape[0], dtype=int)
    for idxi, i in enumerate(test_data):
        # compute distance array for all train points
        distance_array = []
        for idxj,j in enumerate(training_data):
            distance_array.insert( 0, [distance(i, j, 2),␣
↪training_labels[idxj]] )

        # sort the distance array
        distance_array.sort(key = lambda x:x[0])

        # get the first k values
        distance_array = distance_array[:K]

        # get count of all values
        counts = {}

        for i in distance_array:
            if i[1] in counts:
                counts[i[1]] += 1
            else:
                counts[i[1]] = 1

        # get the label that has the max counts
        max_key = max(counts, key = counts.get)
        pred_labels[idxi] = max_key

    # return the final array
    return pred_labels
```

4.3.2 Question 2b: Measuring performance [20]

In this question you will have to evaluate the average performance of your classifier for different values of K. In particular, K will range in $\{1, \cdots, 10\}$. We are going to measure the performance using classification accuracy. For computing the accuracy, you may use

`accuracy = sum(test_labels == pred_labels)/len(test_labels)`

where `test_labels` are the actual class labels and `pred_labels` are the predicted labels

In order to get a proper estimate for the accuracy for every K, we need to run multiple iterations where for each iteration we get a different randomized split of our data into train and test. In this question, we are going to run 100 iterations for every K, and for every random splitting, you may use:

`(training_data, test_data, training_labels, test_labels) = train_test_split(X, Y, test_size=0.3`

where the train/test ratio is 70/30.

After computing the accuracy for every K for every iteration, you will have 100 accuracies per K. The best way to store those accuracies is in a matrix that has as many rows as values for K and 100 columns, each one for each iteration.

Compute average accuracy as a function of K. Because we have a randomized process, we also need to compute how certain/uncertain our estimation for the accuracy per K is. For that reason, we also need to compute the standard deviation of the accuracy for every K. Having computed both average accuracy and standard deviation, make a figure that shows the average accuracy as a function of K with each point of the figure being surrounded by an error-bar encoding the standard deviation. You may find

`plt.errorbar()`

useful for this plot.

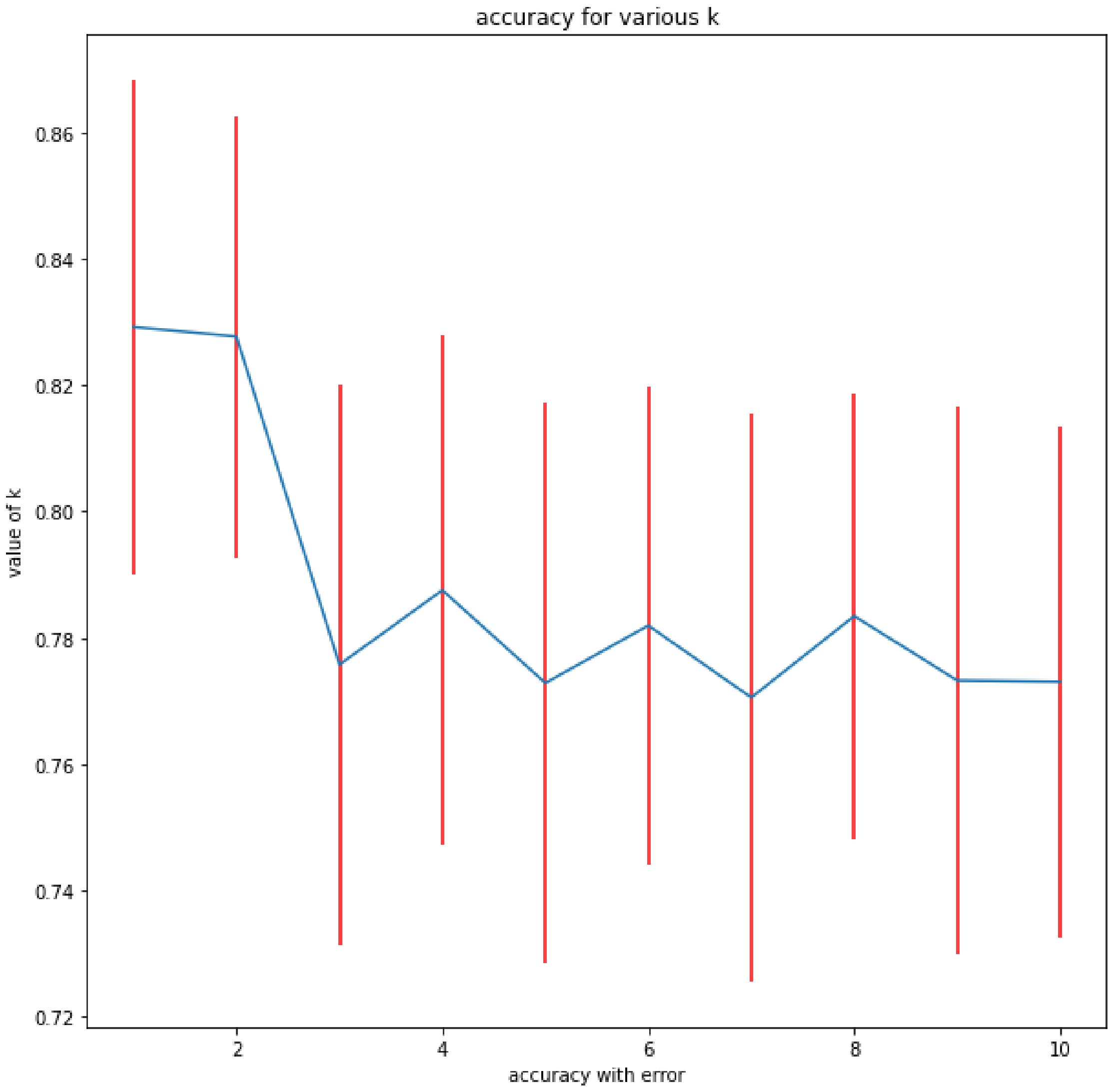
[9]:

```
# TODO: write your code here
# You should be able to get above 95% accuracies

k_mean_list = []
k_std_list = []
k_range_list = [l for l in range(1,11)]

output_matrix = np.zeros((len(k_range_list),100))

for k in k_range_list:
    print ('For k => ', k)
    for epoch in range(output_matrix.shape[1]):
        if epoch %50 == 0:
            print ('For epoch => ', epoch)
        (training_data, test_data, training_labels, test_labels) =␣
 ↪train_test_split(X, Y, test_size=0.3)
        pred_labels = knnclassify(test_data, training_data, training_labels,␣
 ↪K=k)
        accuracy = sum(test_labels == pred_labels)/len(test_labels)
        output_matrix[k-1][epoch] = accuracy
    k_mean_list.append(output_matrix[k-1].mean())
    k_std_list.append(output_matrix[k-1].std())
```

```
For k =>  1
For epoch =>  0
For epoch =>  50
For k =>  2
For epoch =>  0
For epoch =>  50
For k =>  3
For epoch =>  0
For epoch =>  50
For k =>  4
For epoch =>  0
For epoch =>  50
For k =>  5
For epoch =>  0
For epoch =>  50
For k =>  6
For epoch =>  0
For epoch =>  50
For k =>  7
For epoch =>  0
For epoch =>  50
For k =>  8
For epoch =>  0
For epoch =>  50
For k =>  9
For epoch =>  0
For epoch =>  50
For k =>  10
For epoch =>  0
For epoch =>  50
```

[10]:

```
plt.figure(figsize = (10,10))
plt.title('accuracy for various k')
plt.ylabel('value of k')
plt.xlabel('accuracy with error')
plt.errorbar(k_range_list, k_mean_list, yerr = k_std_list, ecolor = 'red')
# plt.savefig('10k.png')
plt.show()
```



Note about data normalization

You will notice that if you compute distances using the raw data, the KNN accuracy will be in 77-83% range.

You can improve the accuracy of the KNN algorithm to above 95% by performing data normalization as the pre-processing step where you can first convert every feature value to the range [0,1] and then perform KNN algorithm. This makes sense in this problem because the four features have a very different range. If you just compute the distance without normalization, then features with large values influence the results more.

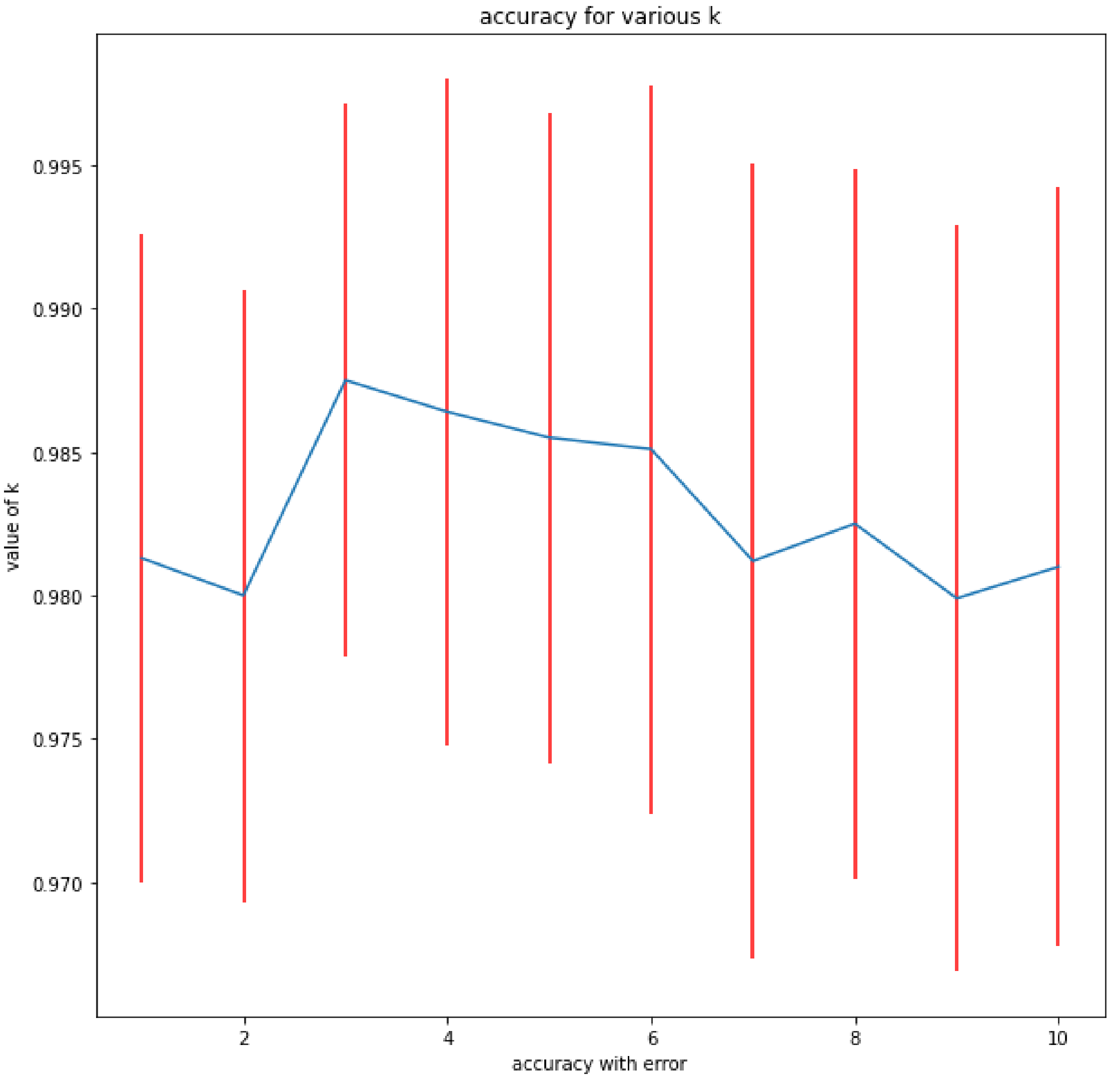
You can read more about such normalization procedures here.

[11]:
```
# Working with normalized dataset
```

```
data4 = data.copy()
data4_scaled = data4.copy()
col_names_list = data4.columns.values.tolist()

col_names_list.remove('species')
col_names_list.remove('sex')
col_names_list.remove('island')

for column in col_names_list:
    col_max = data4_scaled[column].max()
    col_min = data4_scaled[column].min()
    data4_scaled[column] = (data4_scaled[column] - col_min) / (col_max -␣
 ↪col_min)

# prepare datasets, convert to numpy array, X - data values, Y - labels
X = data4_scaled[['bill_length_mm','bill_depth_mm','flipper_length_mm',␣
 ↪'body_mass_g']].values
all_labels = data['species'].values
unique_labels = np.unique(all_labels)
new_labels = np.zeros(len(all_labels))
for i in range(0,len(unique_labels)):
  new_labels[all_labels == unique_labels[i]] = i
Y = new_labels

# TODO: write your code here
# You should be able to get above 95% accuracies

k_mean_list = []
k_std_list = []
k_range_list = [l for l in range(1,11)]

output_matrix = np.zeros((len(k_range_list),100))

for k in k_range_list:
    print ('For k => ', k)
    for epoch in range(output_matrix.shape[1]):
        if epoch %50 == 0:
            print ('For epoch => ', epoch)
        (training_data, test_data, training_labels, test_labels) =␣
 ↪train_test_split(X, Y, test_size=0.3)
        pred_labels = knnclassify(test_data, training_data, training_labels,␣
 ↪K=k)
        accuracy = sum(test_labels == pred_labels)/len(test_labels)
        output_matrix[k-1][epoch] = accuracy
    k_mean_list.append(output_matrix[k-1].mean())
    k_std_list.append(output_matrix[k-1].std())
```

```python
plt.figure(figsize = (10,10))
plt.title('accuracy for various k')
plt.ylabel('value of k')
plt.xlabel('accuracy with error')
plt.errorbar(k_range_list, k_mean_list, yerr = k_std_list, ecolor = 'red')
# plt.savefig('10k.png')
plt.show()
```

```
For k =>  1
For epoch =>  0
For epoch =>  50
For k =>  2
For epoch =>  0
For epoch =>  50
For k =>  3
For epoch =>  0
For epoch =>  50
For k =>  4
For epoch =>  0
For epoch =>  50
For k =>  5
For epoch =>  0
For epoch =>  50
For k =>  6
For epoch =>  0
For epoch =>  50
For k =>  7
For epoch =>  0
For epoch =>  50
For k =>  8
For epoch =>  0
For epoch =>  50
For k =>  9
For epoch =>  0
For epoch =>  50
For k =>  10
For epoch =>  0
For epoch =>  50
```



4.3.3 Question 2c: Feature Selection [10]

For the following questions, you should perform KNN classification using only 2 features from `['bill_length_mm','bill_depth_mm','flipper_length_mm', 'body_mass_g']` per sample.

You may want to refer to pairplot above to decide which features are best

Answer the following questions

1. Which two features did you choose to get the **best** performance for KNN algorithm?
2. Which two features did you choose to get the **worst** performance for KNN algorithm?
3. Why? (Justify your answer)

Write your answer here:

1. Using 'bill_length_mm' and 'bill_depth_mm' or 'bill_length_mm' and 'flipper_length_mm' a good accuracy of upto 95% can be achieved with k = 3
2. Using 'flipper_length_mm' and 'body_mass_g' a very bad accuracy of upto 71% can be achieved with k = 3
3. This phenomenon can also be seen in the pairplots that the clusters corresponding to all the species are seperate in case of good accuracy parameters like 'bill_length_mm' and 'bill_depth_mm' and overlapping in case of worse accuracy. Also as the magnitude of values of 'body_mass_g' is much greater than other values so it tends to influence the distance calculations more thus resulting in wrong predictions. Normalization of values can help in this case and it was noticed that if the values were normalized, the accuracy increases to 98%.

4.4 Submission instructions

1. Download this Colab to ipynb, and convert it to PDF. Follow similar steps as here but convert to PDF.

- Download your .ipynb file. You can do it using only Google Colab. `File -> Download -> Download .ipynb`
- Reupload it so Colab can see it. Click on the `Files` icon on the far left to expand the side bar. You can directly drag the downloaded .ipynb file to the area. Or click `Upload to session storage` icon and then select & upload your .ipynb file.
- Conversion using %%shell. `!sudo apt-get update !sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-generic-recommended !jupyter nbconvert --log-level CRITICAL --to pdf name_of_hw.ipynb`
- Your PDF file is ready. Click 3 dots and `Download`.

2. Upload the PDF to Gradescope, select the correct pdf pages for each question. **Important!**
3. Upload the ipynb file to Gradescope

[12]:
```
!sudo apt-get update
!sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-generic-recommended
!jupyter nbconvert --log-level CRITICAL --to pdf fall2022_hw1.ipynb # make sure the ipynb name is correct
```

```
Hit:1 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Hit:2 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Get:3 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
[3,626 B]
Hit:4 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Hit:5 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Hit:6 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:7 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Ign:9 https://developer.download.nvidia.com/compute/machine-
```

```
learning/repos/ubuntu1804/x86_64  InRelease
Get:10
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
InRelease [1,581 B]
Hit:11 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  Release
Get:12
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
Packages [950 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages
[2,992 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages
[2,324 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages
[3,424 kB]
Get:18 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages
[1,546 kB]
Fetched 11.5 MB in 2s (4,919 kB/s)
Reading package lists… Done
Reading package lists… Done
Building dependency tree
Reading state information… Done
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-latex-base texlive-latex-extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa
Suggested packages:
  fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
  | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
  ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
  | pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
  python-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
  | libtcltk-ruby texlive-pictures-doc vprerex
The following NEW packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-generic-recommended
  texlive-latex-base texlive-latex-extra texlive-latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa
0 upgraded, 47 newly installed, 0 to remove and 12 not upgraded.
Need to get 146 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback
all 1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data all
0.4.8-2 [1,479 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common all 6.09
[33.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern all
2.004.5-3 [4,551 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono all
20171026-2 [75.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre all
20160520-1 [8,761 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common all
11 [6,066 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1
amd64 1.20.2-0ubuntu3.1 [108 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2
amd64 2.2.7-1ubuntu2.9 [18.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35 amd64
0.35-13 [15.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0 amd64
0.13-6 [55.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common
all 9.26~dfsg+0-0ubuntu0.18.04.17 [5,092 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9 amd64
9.26~dfsg+0-0ubuntu0.18.04.17 [2,267 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjs-jquery all
3.2.1-1 [152 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libkpathsea6
amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpotrace0 amd64
1.14-2 [17.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libptexenc1
amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
```

```
Get:19 http://archive.ubuntu.com/ubuntu bionic/main amd64 rubygems-integration
all 1.11 [4,994 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5 amd64
2.5.1-1ubuntu1.12 [48.6 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby amd64 1:2.5.1
[5,712 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 rake all
12.3.1-1ubuntu0.1 [44.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-did-you-mean all
1.2.0-2 [9,700 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-minitest all
5.10.3-1 [38.6 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-power-assert all
0.3.0-1 [7,952 B]
Get:27 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-test-unit all
3.2.5-1 [61.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libruby2.5
amd64 2.5.1-1ubuntu1.12 [3,073 kB]
Get:29 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsynctex1
amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexlua52
amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
Get:31 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexluajit2
amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
Get:32 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libzzip-0-13
amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
Get:33 http://archive.ubuntu.com/ubuntu bionic/main amd64 lmodern all 2.004.5-3
[9,631 kB]
Get:34 http://archive.ubuntu.com/ubuntu bionic/main amd64 preview-latex-style
all 11.91-1ubuntu1 [185 kB]
Get:35 http://archive.ubuntu.com/ubuntu bionic/main amd64 t1utils amd64 1.41-2
[56.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tex-gyre all
20160520-1 [4,998 kB]
Get:37 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 texlive-
binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]
Get:38 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-base all
2017.20180305-1 [18.7 MB]
Get:39 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-fonts-
recommended all 2017.20180305-1 [5,262 kB]
Get:40 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-plain-
generic all 2017.20180305-2 [23.6 MB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-generic-
recommended all 2017.20180305-1 [15.9 kB]
Get:42 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-base all
2017.20180305-1 [951 kB]
```

```
Get:43 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-
recommended all 2017.20180305-1 [14.9 MB]
Get:44 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-pictures
all 2017.20180305-1 [4,026 kB]
Get:45 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-latex-
extra all 2017.20180305-2 [10.6 MB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tipa all 2:1.3-20
[2,978 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 6s (23.0 MB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 76,
<> line 47.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 123934 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb …
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2_all.deb …
Unpacking fonts-lato (2.0-2) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.8-2_all.deb …
Unpacking poppler-data (0.4.8-2) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.09_all.deb …
Unpacking tex-common (6.09) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/04-fonts-lmodern_2.004.5-3_all.deb …
Unpacking fonts-lmodern (2.004.5-3) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/05-fonts-noto-mono_20171026-2_all.deb …
Unpacking fonts-noto-mono (20171026-2) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/06-fonts-texgyre_20160520-1_all.deb …
Unpacking fonts-texgyre (20160520-1) …
Selecting previously unselected package javascript-common.
Preparing to unpack …/07-javascript-common_11_all.deb …
Unpacking javascript-common (11) …
Selecting previously unselected package libcupsfilters1:amd64.
Preparing to unpack …/08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb …
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
```

```
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack …/09-libcupsimage2_2.2.7-1ubuntu2.9_amd64.deb …
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.9) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/10-libijs-0.35_0.35-13_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-13) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/11-libjbig2dec0_0.13-6_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.13-6) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.17_all.deb
…
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.17) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.17_amd64.deb …
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.17) …
Selecting previously unselected package libjs-jquery.
Preparing to unpack …/14-libjs-jquery_3.2.1-1_all.deb …
Unpacking libjs-jquery (3.2.1-1) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libpotrace0.
Preparing to unpack …/16-libpotrace0_1.14-2_amd64.deb …
Unpacking libpotrace0 (1.14-2) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/18-rubygems-integration_1.11_all.deb …
Unpacking rubygems-integration (1.11) …
Selecting previously unselected package ruby2.5.
Preparing to unpack …/19-ruby2.5_2.5.1-1ubuntu1.12_amd64.deb …
Unpacking ruby2.5 (2.5.1-1ubuntu1.12) …
Selecting previously unselected package ruby.
Preparing to unpack …/20-ruby_1%3a2.5.1_amd64.deb …
Unpacking ruby (1:2.5.1) …
Selecting previously unselected package rake.
Preparing to unpack …/21-rake_12.3.1-1ubuntu0.1_all.deb …
Unpacking rake (12.3.1-1ubuntu0.1) …
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack …/22-ruby-did-you-mean_1.2.0-2_all.deb …
Unpacking ruby-did-you-mean (1.2.0-2) …
Selecting previously unselected package ruby-minitest.
Preparing to unpack …/23-ruby-minitest_5.10.3-1_all.deb …
Unpacking ruby-minitest (5.10.3-1) …
```

```
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/24-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-power-assert.
Preparing to unpack …/25-ruby-power-assert_0.3.0-1_all.deb …
Unpacking ruby-power-assert (0.3.0-1) …
Selecting previously unselected package ruby-test-unit.
Preparing to unpack …/26-ruby-test-unit_3.2.5-1_all.deb …
Unpacking ruby-test-unit (3.2.5-1) …
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack …/27-libruby2.5_2.5.1-1ubuntu1.12_amd64.deb …
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.12) …
Selecting previously unselected package libsynctex1:amd64.
Preparing to unpack …/28-libsynctex1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack …/29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/31-libzzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Selecting previously unselected package lmodern.
Preparing to unpack …/32-lmodern_2.004.5-3_all.deb …
Unpacking lmodern (2.004.5-3) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/33-preview-latex-style_11.91-1ubuntu1_all.deb …
Unpacking preview-latex-style (11.91-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/34-t1utils_1.41-2_amd64.deb …
Unpacking t1utils (1.41-2) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/35-tex-gyre_20160520-1_all.deb …
Unpacking tex-gyre (20160520-1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/37-texlive-base_2017.20180305-1_all.deb …
Unpacking texlive-base (2017.20180305-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/38-texlive-fonts-recommended_2017.20180305-1_all.deb …
```

```
Unpacking texlive-fonts-recommended (2017.20180305-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/39-texlive-plain-generic_2017.20180305-2_all.deb …
Unpacking texlive-plain-generic (2017.20180305-2) …
Selecting previously unselected package texlive-generic-recommended.
Preparing to unpack …/40-texlive-generic-recommended_2017.20180305-1_all.deb
…
Unpacking texlive-generic-recommended (2017.20180305-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/41-texlive-latex-base_2017.20180305-1_all.deb …
Unpacking texlive-latex-base (2017.20180305-1) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/42-texlive-latex-recommended_2017.20180305-1_all.deb …
Unpacking texlive-latex-recommended (2017.20180305-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/43-texlive-pictures_2017.20180305-1_all.deb …
Unpacking texlive-pictures (2017.20180305-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/44-texlive-latex-extra_2017.20180305-2_all.deb …
Unpacking texlive-latex-extra (2017.20180305-2) …
Selecting previously unselected package tipa.
Preparing to unpack …/45-tipa_2%3a1.3-20_all.deb …
Unpacking tipa (2:1.3-20) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/46-texlive-xetex_2017.20180305-1_all.deb …
Unpacking texlive-xetex (2017.20180305-1) …
Setting up libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.17) …
Setting up libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libjs-jquery (3.2.1-1) …
Setting up libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1) …
Setting up libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up tex-common (6.09) …
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line
76.)
debconf: falling back to frontend: Readline
update-language: texlive-base not installed and configured, doing nothing!
Setting up poppler-data (0.4.8-2) …
Setting up tex-gyre (20160520-1) …
Setting up preview-latex-style (11.91-1ubuntu1) …
Setting up fonts-texgyre (20160520-1) …
Setting up fonts-noto-mono (20171026-2) …
Setting up fonts-lato (2.0-2) …
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.9) …
Setting up libjbig2dec0:amd64 (0.13-6) …
Setting up ruby-did-you-mean (1.2.0-2) …
Setting up t1utils (1.41-2) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up libijs-0.35:amd64 (0.35-13) …
Setting up rubygems-integration (1.11) …
Setting up libpotrace0 (1.14-2) …
Setting up javascript-common (11) …
Setting up ruby-minitest (5.10.3-1) …
Setting up libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.17) …
Setting up libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-lmodern (2.004.5-3) …
Setting up ruby-power-assert (0.3.0-1) …
Setting up texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up texlive-base (2017.20180305-1) …
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4:
/var/lib/texmf/tex/generic/config/pdftexconfig.tex
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line
76.)
debconf: falling back to frontend: Readline
Setting up texlive-fonts-recommended (2017.20180305-1) …
Setting up texlive-plain-generic (2017.20180305-2) …
Setting up texlive-generic-recommended (2017.20180305-1) …
Setting up texlive-latex-base (2017.20180305-1) …
Setting up lmodern (2.004.5-3) …
Setting up texlive-latex-recommended (2017.20180305-1) …
Setting up texlive-pictures (2017.20180305-1) …
Setting up tipa (2:1.3-20) …
Regenerating '/var/lib/texmf/fmtutil.cnf-DEBIAN'… done.
Regenerating '/var/lib/texmf/fmtutil.cnf-TEXLIVEDIST'… done.
update-fmtutil has updated the following file(s):
        /var/lib/texmf/fmtutil.cnf-DEBIAN
        /var/lib/texmf/fmtutil.cnf-TEXLIVEDIST
If you want to activate the changes in the above file(s),
you should run fmtutil-sys or fmtutil.
Setting up texlive-latex-extra (2017.20180305-2) …
Setting up texlive-xetex (2017.20180305-1) …
Setting up ruby2.5 (2.5.1-1ubuntu1.12) …
Setting up ruby (1:2.5.1) …
Setting up ruby-test-unit (3.2.5-1) …
Setting up rake (12.3.1-1ubuntu0.1) …
Setting up libruby2.5:amd64 (2.5.1-1ubuntu1.12) …
Processing triggers for mime-support (3.60ubuntu1) …
Processing triggers for libc-bin (2.27-3ubuntu1.6) …
Processing triggers for man-db (2.8.3-2ubuntu0.1) …
Processing triggers for fontconfig (2.12.6-0ubuntu2) …
Processing triggers for tex-common (6.09) …
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line
76.)
debconf: falling back to frontend: Readline
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time… done.
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
```

```
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and
include the error message in the cell output (the default behaviour is to abort
conversion). This flag is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=
--ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True
--TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True
--TemplateExporter.exclude_input=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
```

```
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
'pdf', 'python', 'rst', 'script', 'slides']
            or a dotted object name that represents the import path for an
            `Exporter` class
    Default: 'html'
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template file to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_file]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                  to output to the directory of each notebook.
To recover
                                  previous default behaviour (outputting to the
current
                                  working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a
copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-
html-slideshow)
```

```
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb

            which will convert mynotebook.ipynb to the default format (probably
HTML).

            You can specify the export format with `--to`.
            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', 'pdf', 'python', 'rst', 'script', 'slides'].

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX
includes
            'base', 'article' and 'report'.  HTML includes 'basic' and 'full'.
You
            can specify the flavor of the format used.

            > jupyter nbconvert --to html --template basic mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:
```

```
            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file, containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```

[12]:

```

```