Untitled

April 26, 2021

1 Introduction to probabilities

2 Probability mass function

1. let's take fair coin with two sides head h_h, and tails h_t, in the experiment of flipping the coin k times, the probability (probability mass function) of getting heads is $\frac{k_h}{k}$, and probability of getting tail is $\frac{k_t}{k}$, and the sum of all possible events is $\frac{k_h}{k} + \frac{k_t}{k} = 1$, note that $k_h + k_t = 1$.
2. in this experiment the possible outcomes $\Omega = head, tail$, and probability of an event A, $p(A) = \frac{|A|}{|\Omega|}$, the sample space in the experiment has size of 2, for event A being head, the frequency of getting head in single coin flip is just 1, then $p(A) = \frac{1}{2}$.
3. let's take another example, When we throw a die, the obvious choice of the sample space is $\Omega = 1, 2, 3, 4, 5, 6$, and the probability mass function should be given by p(i) = 1/6, $i = 1, \dots, 6$. The probability of the event $\{2, 4, 6\}$ that the outcome is even is now easily seen to be $p(\{2, 4, 6\}) = p(2) + p(4) + p(6) = \frac{1}{2}$.

3 Combinatorics

1. starting with a set, ordered, or unordered (the difference matters in calculation), for example 1,2,3, the list of ordered subsets of size two are (1,2), (2,1), (1,3), (3,1), (2,3), (3,2); the list of unordered subsets of size two are 1,2, 1,3, and 2,3.
2. given any set with length n, the number of possible sequences of length k is n^k, in the previous example all possible sequences of length 2 is $3^2 = 6$
3. the number of ordered subsets of k elements from the same set is: $n \times (n-1) \times \cdots \times (n-k+1)$, in the previous set that is 3×2.
4. the number of subsets of k elements from a set with n elements is n choose k, denoted by $\binom{n}{k} = \frac{n \times (n-1) \times \cdots \times (n-k+1)}{k!} = \frac{n!}{k!(n-k)!}$
5. Example(with replacement) Consider an urn with eight balls,numbered $1, \dots, 8$. We draw three balls with replacement, that is, after drawing a ball, we note its number and put it back into the urn, so that it may be drawn a second or even a third time. The sample space Ω of this experiment consists of all sequences of length three, with the symbols $1, \dots, 8$. the sample space Ω has $8^3 = 512$ elements. we can conclude for any sequence of length 3 for instance (4, 4, 8) has probability 1/512 to occur.

6. Example(without replacement) Consider the same urn, with the same balls. We now draw three balls without replacement, that is, a chosen ball is not put back in the urn. We note the numbers of the chosen balls, in order. The sample space Ω' corresponding to this event is the set consisting of all sequences of length three with the symbols $1, \ldots, 8$ where each symbol can appear at most once. The number of elements in Ω' is the number of ordered subsets of size three. this is equal to $8 \times 7 \times 6 = 336$.

7. Example(unordered subset) Consider the same urn once more, this time choosing three balls simultaneously, so that the order is irrelevant. This experiment corresponds to the sample space Ω'' which consists of all subsets of size three of a set with eight elements. then Ω'' of size 8 with choosing 3 with order $\frac{8X7X6}{3!} = 56$ elements. the probability to select the set 3, 7, 1 is now 1/56. Note that this is six times the probability of the event in the previous example. The reason for this is that the set 3, 7, 1 can appear in 3! = 6 different orderings.

4 condition probability

1. for two events A, B in sample space Ω, suppose $P(B) > 0$:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{1}$$

2. Example1, suppose we throw a die, what is the conditional probability of seeing a 3, conditioned ont he event that the outcome is at most 4? denoting the event of seeing 3 by E_3, and the event of the outcome is at most 4 by $E_{\leq 4}$, in the obvious sample space, that $P(E_3) = \frac{1}{6}$, $P(E_{\leq 4}) = \frac{2}{3}$, and $P(E_3 \cap E_{leq4}) = P(E_3) = \frac{1}{6}$, hence:

$$P(E_3|E_{\leq 4}) = \frac{P(E_3 \cap E_{\leq 4})}{P(E_{\leq})} = \frac{1/6}{2/3} = \frac{1}{4} \tag{2}$$

3. Example2: suppose we have a population of people, suppose in addition that the probability that an individual has a certain disease is 1/100. There is a test for this disease, and this test is 90% acccurate, in the sense that the probability that a sick person is tested positive is 09, and that a healthy person is tested positive with probability 0.1. One particular individual is tested positive. Perhaps this individual is inclined to think the following: 'I have been tested positive by a test which is accurate 90

4. let A be the event that this individual has the disease, and let B be the event that the test is positive. The individual is interested in the conditional probability $P(A|B)$. for a sick person, the test is positive with probability 0.9, Hence $P(B|A) = 0.9$, and $P(B|A^c) = 0.1$, and that $P(A) = 0.01$.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^c)P(A^c)} \tag{3}$$

$$= \frac{0.9 * 0.01}{0.9 * 0.01 + 0.1 * 0.99} = 0.09 \tag{4}$$

5 bayes rule

1. we can generalize the previous rule, let $B_1, B_2, \ldots, B_3$ be a partition of Ω such that $P(B_i) > 0$ for all i, and let A be any event with $P(A) > 0$, then for all i,

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_{j=1}^{n} P(A|B_j)P(B_j)} \tag{5}$$

5.1 Motiviational example, are you ready?

6 Naive Bayes

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix

%matplotlib inline
sns.set(rc={'figure.figsize': [10, 10]}, font_scale=1.2)
```

[2]:
```
df = pd.read_csv('sentiment.csv',sep='\t',names=['Liked','Tweet'])
```

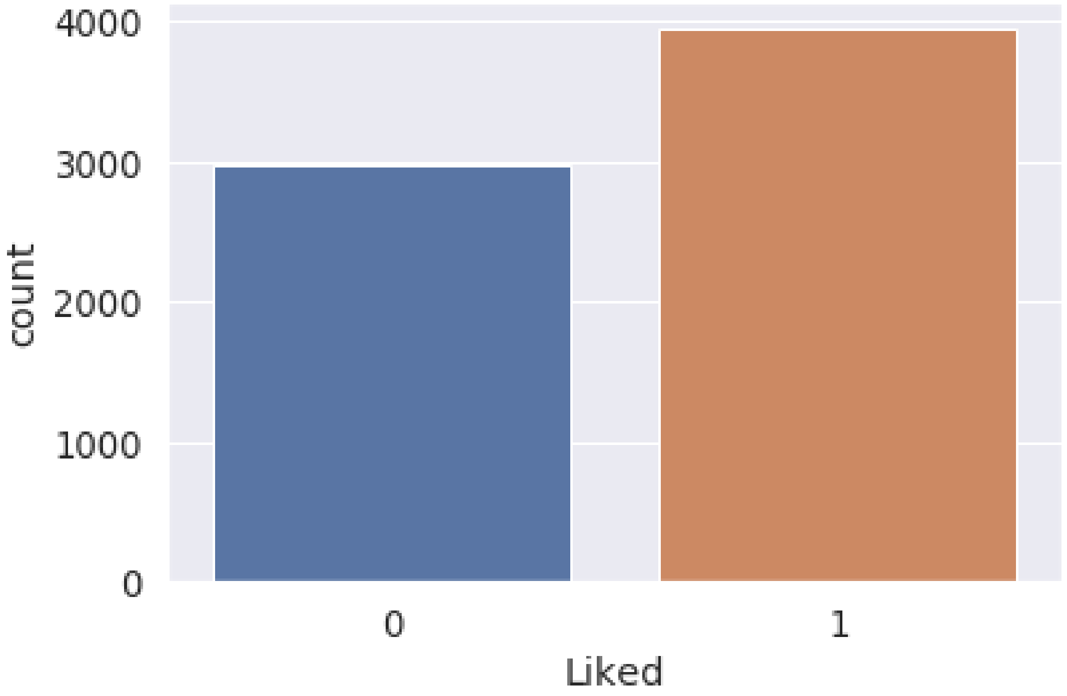
[3]:
```
df
```

[3]:
```
      Liked                                              Tweet
0         1             The Da Vinci Code book is just awesome.
1         1  this was the first clive cussler i've ever rea…
2         1                    i liked the Da Vinci Code a lot.
3         1                    i liked the Da Vinci Code a lot.
4         1  I liked the Da Vinci Code but it ultimatly did…
…       …                                              …
6913      0                      Brokeback Mountain was boring.
6914      0          So Brokeback Mountain was really depressing.
6915      0  As I sit here, watching the MTV Movie Awards, …
6916      0    Ok brokeback mountain is such a horrible movie.
6917      0   Oh, and Brokeback Mountain was a terrible movie.

[6918 rows x 2 columns]
```

[4]:
```
sns.countplot(x='Liked', data=df)
```

[4]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f260ee4c340>
```



```
[5]: x = df["Tweet"]
     y = df["Liked"]
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
      ↪random_state=22)
     tfidv = TfidfVectorizer(stop_words='english')
     tfidv.fit(x_train)
     x_train = tfidv.transform(x_train)
     x_test = tfidv.transform(x_test)
```

```
[6]: pd.DataFrame(x_train.toarray(), columns=tfidv.get_feature_names())
```

```
[6]:       007   10  10pm   12   17  1984  1st  200  2007  286  …  yes  \
     0     0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     1     0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     2     0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     3     0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     4     0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     …    …   …    …   …    …    …   …   …    …   …   …   …
     5529  0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     5530  0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     5531  0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     5532  0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
     5533  0.0  0.0   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0.0  …  0.0
```

```
      yesterday  yip  young  younger  yuck  yuh  zach  zen    µª
0           0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
1           0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
2           0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
3           0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
4           0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
…         …  …    …      …   …  …   …  …  …
5529        0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
5530        0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
5531        0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
5532        0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0
5533        0.0  0.0    0.0      0.0   0.0  0.0   0.0  0.0   0.0

[5534 rows x 1619 columns]
```

[7]:

```
model = MultinomialNB()
model.fit(x_train, y_train)
```

[7]:

```
MultinomialNB()
```

[8]:

```
# test the model on the test_set
y_pred = model.predict(x_test)
#calculate the accuracy
accuracy_score(y_test, y_pred)
#print confusion_matrix
confusion_matrix(y_test, y_pred)
```

[8]:

```
array([[580,  18],
       [  7, 779]])
```

[9]:

```
model.predict(tfidv.transform(['I love The Matrix Movie']))
```

[9]:

```
array([1])
```

7 calculus

1. given a function $f(x)$, how to calculate the rate of change at specific point of time?
2. the rate of change $= \frac{\Delta y}{\Delta x}$
3. but first how to interpret the rate of change? it's how var the function f changes per x, for example for f being the function of distance, and x is the time the rate of change is the average speed.
4. what if we are only interested of the speed at specific infinitesimal point? that is when the derivative is useful.

5. the derivative is the infinitesimal rate of change, or in other words, the rate of change of function f(x) when $\Delta x \rightarrow 0$

6. to generalize this rule the derivative of f(x)

$$\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x))}{\Delta x} \tag{6}$$

8 derivatives of most common function

- $$\frac{d}{dx} x^a = a x^{a-1} \tag{7}$$

- $$\frac{d}{dx} e^x = e^x \tag{8}$$

- $$\frac{d}{dx} a^x = a^x ln(a) \mid a > 0 \tag{9}$$

- $$\frac{d}{dx} ln(x) = \frac{1}{x} \mid x > 0 \tag{10}$$

- $$\frac{d}{dx} \log_a(x) = \frac{1}{x ln(a)} \mid x, a > 0 \tag{11}$$

- $$\frac{d}{dx} sin(x) = cos(x) \tag{12}$$

- $$\frac{d}{dx} cos(x) = -sin(x) \tag{13}$$

- $$\frac{d}{dx} tan(x) = sec^2(x) \tag{14}$$

9 higher derivatives

1. we learned to calculate the first derivative or the infinitesimal rate of change, for function f(x), if y=f(x) is the function of distance, and $y\prime = \frac{df(x)}{dx}$ is the velocity, then the second rate (acceleration) of change $y\prime\prime = \frac{d}{dx}\frac{df(x)}{dx}$ is calculate using the same rules above.

2. note that for a function f(x) to be differentiable is has to be continuous over x.

10 partial derivatives

1. so far we discussed function defined in single variable, what if f is defined in two dimensions, or 3 dimensions, for example, f(x,y,z) could be the function of position of variable defined in x,y,z axis, the derivative is defined similarly in x,y,z respectively, and denoted as such:

$$\frac{\partial f}{\partial x} = \frac{\partial f(x,y,z)}{\partial x} \tag{15}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f(x,y,z)}{\partial y} \tag{16}$$

$$\frac{\partial f}{\partial z} = \frac{\partial f(x,y,z)}{\partial z} \tag{17}$$

$$\nabla f(x,y,z) = \frac{\partial f(x,y,z)}{\partial x}\hat{\imath} + \frac{\partial f(x,y,z)}{\partial x}\hat{\jmath} + \frac{\partial f(x,y,z)}{\partial x}\hat{k} \tag{18}$$

2. for example $f(x,y,z) = x^2 + y^2 + 2z + 1$, then

$$\nabla f(x,y,z) = 2x\hat{\imath} + 2y\hat{\jmath} + 2\hat{k} \tag{19}$$

3. meaning that the derivative of f in x-axis = 2x, and on y-axis is 2y, and on z-axis is 2

11 chain-rule of partial derivatives

1. for function f defined in terms of second function g, where g is defined in terms of third function z, where z is function of x as follows f(g(z(x))), what is $\frac{df}{dx}$?

$$\frac{df}{dx} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial z}\frac{\partial z}{\partial x} \tag{20}$$

2. Example assume $f(g(z(x)))$ is the function of position of a particle, $f(g) = g^2$, $g(z) = z^2, z(x) = 2x$, what is the speed of the particle? particle speed: $\frac{\partial f}{x}$:

$$\frac{\partial f}{\partial g}\frac{\partial g}{\partial z}\frac{\partial z}{\partial x} = 2g * 2z * 2 \tag{21}$$

12 integrals

1. starting with the derivative $\frac{f(x)}{dx}$, how can we derive the original function f(x)?
2. remember $y\prime = \frac{f(x)}{dx}$, then $y\prime dx = df(x)$, the reverse of the d operator is the integral operator $\int$, thus $f(x) = \int y\prime dx$
3. since the integral operator is the inverse of the derivative operator, we can reverse the direction of the derivatives above to derive the rules for integration:

   - $$\int ax^{a-1} = x^a + c \tag{22}$$

- $$\int e^x = e^x + c \tag{23}$$
- $$\int a^x ln(a) = a^x + c \mid a > 0 \tag{24}$$
- $$\int \frac{1}{x} = ln(x) \mid x > 0 \tag{25}$$
- $$\int \frac{1}{xln(a)} = \log_a(x) + c \mid x, a > 0 \tag{26}$$
- $$\int cos(x) = sin(x) + c \tag{27}$$
- $$\int sin(x) = -cos(x) + c \tag{28}$$
- $$\int sec^2(x) = tan(x) + c \tag{29}$$

4. note that the c is the derivative constant, but why it's there?
5. imagine the original function $y = f(x) = x^2 + 1$

$$y\prime = \frac{df(x)}{dx} = 2x \tag{30}$$

$$\int y\prime dx = \int 2x = x^2 + c \tag{31}$$

6. note if we didn't add the constant c, then there is no way to derive the intercept 1, and the way to recover is through any valid point from the original function, for example the point$(1, 2)$ satisfies f(x), $2 = 1 + c$, then $c = 1$, thus the original equation $y = x^2 + 1$.

13 integration defined over a range

1. the integration for function $\frac{df(x)}{dx}$, defined over an interval [a-b], denoted by:

$$\int_a^b \frac{df(x)}{dx} dx = f(x)|_a^b = f(b) - f(a) \tag{32}$$

1. Exercise, for a car moving to the left from the origin at rest moving in a straight line, such that for a short time it's velocity is defined by $v = (3t^2 + 2t)$ ft/s, where t in seconds, determine its position and acceleration when $t = 3$ s, when $t = 0$, $s = 0$.

2. for the position:

$$v = \frac{ds}{dt} = (3t^2 + 2t) \tag{33}$$

$$\int_0^2 ds = \int_0^t (3t^2 + 2t)dt \tag{34}$$

$$s|_0^s = t^3 + t + 2|_0^t \tag{35}$$

3. at time t=3 s,

$$s = 3^3 + 3^2 = 36ft \tag{36}$$

4. for acceleration:

$$a = \frac{v}{t} = \frac{d(3^2 + 2t)}{dt} \tag{37}$$

$$= 6t + 2 \tag{38}$$

5. when t=3s,

$$a = 6(3) + 2 = 20ft/s^2 \tag{39}$$

14 Linear Algebra

15 vector

1. what is a vector? so far we have been dealing with scale variables, or variables expressed in single dimension thus scalar, imagine a hinge fixed on the wall, the reactionary force from the wall against he hinge is expressed in three dimension, and can be expressed in x,y,z axis respectively:

$$F = \begin{vmatrix} f_x \\ f_y \\ f_z \end{vmatrix} \tag{40}$$

2. variables expressed in terms of vectors have it's own rules of arithmetic, for two vectors

$$v_a = \begin{vmatrix} 3 \\ 6 \end{vmatrix}, and v_b = \begin{vmatrix} 1 \\ 2 \end{vmatrix} \tag{41}$$

.

3. addition, and subtractions are straight forward, and executed for each dimension separately

15.1 addition

(a) .

$$v_a + v_b = \begin{vmatrix} 3+1 \\ 6+2 \end{vmatrix} \tag{42}$$

15.2 subtraction

(a) .

(b)

$$v_a - v_b = \begin{vmatrix} 3-1 \\ 6-2 \end{vmatrix} \tag{43}$$

15.3 dot product

(a) the dot product is representation of how var two vectors are aligned to each others, the dot product between two vector is maximum if the two vectors are aligned along the same line, and is zero if both vectors are perpendicular.

$$v_a.v_b = |v_a| * |v_b| * cos(\theta) \tag{44}$$

(b) θ is the angle between the two vectors

15.4 cross product

(a) the cross product is representation of area between two vector, the resultant vector is perpendicular to the plane wrapping the two vectors.

$$v_a \times v_b = |v_a|\,|v_b| * sin(\theta) \tag{45}$$

(b) θ is the angle between the two vectors

[10]:
```
import numpy as np
```

[11]:
```
V_a = np.array([1, 2, 3])
V_b = np.array([2, 4, 6])
```

[12]:
```
V_c = np.add(V_a, V_b)
```

[13]:
```
V_c
```

[13]:
```
array([3, 6, 9])
```

[14]:
```
V_c = np.subtract(V_a, V_b)
```

[15]:
```
V_c
```

[15]:
```
array([-1, -2, -3])
```

[16]:
```
V_c = np.dot(V_a, V_b)
```

[17]:
```
V_c
```

[17]:
```
28
```

[18]:
```
V_c = np.cross(V_a, V_b)
```

[19]:
```
V_c
```

[19]:
```
array([0, 0, 0])
```

15.4.1 why the cross product is zero?!

16 Matrix

1. so far we talked about scalar variables, and vector variables, the variable be a collection of vectors?
2. in many fields such as robotics, machine learning, and even pure mathematics, there is always a need to perform the same computation, or arithmetic on a collection of vectors at once to either save computation time, or mathematical elegance, and simplicity, for this reason matrices are required, but what are they?
3. matrix is a collection of vectors of any size, for example for vectors in 3-dimensions, a matrix of 2 vectors is 3X2 matrix, a matrix of 4 vectors is 3X4 matrix, and mXn matrix is written as follow: $\begin{vmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & & & \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{vmatrix}$
4. but what a matrix represent? ti can be set of force vectors acting on a rigid body, pixels on an image, robot kinematics state, rotation, or transformation matrix, you can think of a matrix as a data-set of different numerical features.

16.1 arithmetic on the matrix

5. for two matrices of the same shape, 3X3 matrices $A = \begin{vmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \\ 13 & 15 & 17 \end{vmatrix}$, and $B = \begin{vmatrix} 0 & 2 & 4 \\ 6 & 8 & 10 \\ 12 & 14 & 16 \end{vmatrix}$.
6. in addition, and subtraction you need to make sure the two matrices are of the same size, otherwise, the operation fails, just like in vectors.

16.2 addition

7. A+B:

$$\begin{vmatrix} 0+1 & 2+3 & 4+5 \\ 6+7 & 8+9 & 10+11 \\ 12+13 & 14+15 & 16+17 \end{vmatrix} \tag{46}$$

16.3 subtraction

8. A-B:

$$\begin{vmatrix} 0-1 & 2-3 & 4-5 \\ 6-7 & 8-9 & 10-11 \\ 12-13 & 14-15 & 16-17 \end{vmatrix} \tag{47}$$

16.4 multiplication

9. A*B:

$$\begin{vmatrix} 0*1+2*7+4*13 & 0*3+2*9+4*15 & 0*5+2*11+4*17 \\ 6*1+8*7+10*13 & 6*3+8*9+10*15 & 6*5+8*11+10*17 \\ 6*1+8*7+10*13 & 6*3+8*9+10*15 & 6*5+8*11+10*17 \end{vmatrix} \tag{48}$$

10. notice we multiply each row's entries from the first matrix by each column's entries, from the second matrix, also notice that matrix multiplication isn't symmetric, for example $A \times B \neq B \times A$

11. what is the size of the output matrix?

12. $A_{m\times k}$ is a matrix of size $m \times k$, and $B_{k\times n}$ is a matrix of shape $k \times n$, the multiplication is possible only if the number of columns in the first matrix equals to the number of rows in the second matrix, and the resultant matrix is $m \times n$ matrix.

13. for example $A_{3X2} \times B_{2X4} = C_{3X4}$, mean while $B_{2X4} \times A_{3X2}$ isn't valid operation for the columns of B(first) is 4, and the rows of A(second) is 3, and $4 \neq 3$, thus, the operation isn't valid.

14. the product of two matrices can be interpreted as the transformation of one matrix by another.

16.5 Determinant

15. Determinant is a metric of measuring the spread of a square-matrix $n \times n$ vector, or scalar value for the volumn between the matrix vectors.

16. Determinant of matrix A is $|A|$:

16.6 determinant of 2X2 matrix

17. for a square 2X2 matrix:

18. $|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$

16.7 determinant of 3X3 matrix

19. for a square 3X3 matrix:

20. $|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a.det(\begin{vmatrix} e & f \\ h & i \end{vmatrix}) - b.det(\begin{vmatrix} d & f \\ g & i \end{vmatrix}) + c.det(\begin{vmatrix} d & e \\ g & h \end{vmatrix})$

21. note that the odd indices have positive sign, and the even indices have negative sign.

16.8 determinant of nXn matrix

22. the determinant of nXn matrix can be generalize previous maintaining positive signs for odd indices, and negative sign for even indices

23. for example for 4X4 matrix:

$$det(\begin{vmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{vmatrix}) = \tag{49}$$

$$a.det(\begin{vmatrix} f & g & h \\ j & k & l \\ n & o & p \end{vmatrix}) - b.det(\begin{vmatrix} e & g & h \\ i & k & l \\ m & o & p \end{vmatrix}) + c.det(\begin{vmatrix} e & f & h \\ i & j & l \\ m & n & p \end{vmatrix}) - d.det(\begin{vmatrix} e & f & g \\ i & j & k \\ m & n & o \end{vmatrix}) \tag{50}$$

16.9 Transpose

24. the transpose of a matrix A is A^T the function of replacing rows with columns in the same relative order.

$$A = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \mid A^T = \begin{vmatrix} a & c \\ b & d \end{vmatrix} \tag{51}$$

25. note that $(A^T)^T = A$

16.10 Adjugate

26. adjugate is a function over a square matrix, has wide spread in geometry, robotics, and machine learning, and used to define the inverse of a matrix.

27. adjugate of a matrix A, is the transpose of the Cofactor of A.

16.11 adjugate of 2X2 matrix

28. for $A = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$, $adj(A) = \begin{vmatrix} d & -b \\ -c & a \end{vmatrix}$.

16.12 adjugate of nXn matrix

29. adjugate of more than 2X2 matrix can get very complex, and usually calculated with numerical libraries.

16.13 inverse

30. inverse of a matrix is analogous to the inverse in scalar variables, but calculated differently, the inverse of A, is $A^-1 = \frac{1}{A}$.

$$A^-1 = \frac{1}{det(A)} adj(A) \tag{52}$$

16.14 Identity

31. the identity matrix I is a square matrix with value 1 on the left diagonal, zeroed-out in the upper, and lower triangles of the matrix.

$$I_n = \begin{vmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & 1 \end{vmatrix} \tag{53}$$

32. for matrix $A_{n \times n}$, and identity matrix $I_{n \times n}$,

    - $A * I = A = I * A$
    - $I^T = I$
    - $I^-1 = I$

```
[20]: import numpy as np
      M_a = np.array([[1,3,5],[3,4,6],[1,2,3]])
      M_b = np.array([[3,1,6], [3,1,7],[3,1,6]])
```

```
[21]: M_c = np.add(M_a, M_b)
      M_c
```

```
[21]: array([[ 4,  4, 11],
             [ 6,  5, 13],
             [ 4,  3,  9]])
```

```
[22]: M_c = np.subtract(M_a, M_b)
      M_c
```

```
[22]: array([[-2,  2, -1],
             [ 0,  3, -1],
             [-2,  1, -3]])
```

```
[23]: M_c = np.dot(M_a, M_b)
      M_c
```

```
[23]: array([[27,  9, 57],
             [39, 13, 82],
             [18,  6, 38]])
```

```
[24]: M_c = M_a*M_b
      M_c
```

```
[24]: array([[ 3,  3, 30],
             [ 9,  4, 42],
             [ 3,  2, 18]])
```

```
[25]: inv = np.linalg.inv(M_a)
      inv
```

```
[25]: array([[-5.92118946e-16,  1.00000000e+00, -2.00000000e+00],
             [-3.00000000e+00, -2.00000000e+00,  9.00000000e+00],
             [ 2.00000000e+00,  1.00000000e+00, -5.00000000e+00]])
```

```
[26]: det = np.linalg.det(M_a)
      det
```

```
[26]: 1.0000000000000001
```

17 Statistics

18 data

1. statistics is theory of making sense of the data, through certain tools we will explore.

2. but first what is data? what is the data looks like? data set is matrix, or list of vectors, a vector can be Qualitative, or Quantitative.

18.1 data types

3. **Qualitative data**:

   - **Nominal**: nominal data are categorical data types i.e sex, status, religion, there is no priorities between different states.

   - **Ordinal**: ordinal data type is a discrete categorical data type meaning there are priorities, or in other words comparison between different states is possible i.e shoes, or shirt size.

4. **Quantitative data**:

   - **Discrete**: discrete data is monotonically increasing numerical discrete data i.e number of electrons in the cell, or pages of a book.

   - **Continuous**: continuous numerical data such as temperature of a day: 45.3°, or height 177.3 cm.

5. given a data-set of tens, hundreds or thousands of entries, or even millions of entries, how can we make sense of it all?!

19 Anecdotal Evidence

1. we agreed we have a large data-set from hundreds, thousands, to millions of entries, we can't derive a conclusion about the whole data-set from a single entry! for example if the data-set represent the temperature of Cairo in the previous 3 months, assume a single entry was for midnight 10° we can take this entry as a representative of the temperature of Cairo! this is called **Anecdotal evidence**, this is very common in the biased media, and journalism, and considered a bad trait of course.

20 Descriptive Statistics

1. How can we reach a conclusion over the whole data-set from a single number?

2. there are different aspects to look at **Central Tendency**, **Dispersion**, and **Correlation**.

21 Central Tendency

1. central tendency are set of statistical tools to summarizing the most effective entries of the data-set

21.1 Mean

2. the most common of them all is the mean, defined as follows: for data-set of **m** entries of variable x

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{54}$$

3. for example what is the average of for 3 days temperature? the measurements are 44, 45, and 46, the average is $\frac{44+45+46}{3} = 45$

4. in short the mean of variable X is the average of variable X.

21.2 Median

5. in a variable X of numerical values, median is the middle entry, if we said Ahmed is 25, Alaa 27, heba is 24, Ali is 31 mohammed is 29, what is the median, or the middle value? first let's wort this age variable 24, 25, 27, 29, 31, the middle value is 27, so the Media is Alaa's age.

21.3 Mode

6. the most recurrent entry in the data-set variable is coined the Mode, for example if we extend the previous example, if we said Ahmed is 25, Alaa 27, heba is 24, Ali is 31, sara is 25 mohammed is 29, what is the Mode, or the most recurrent value? notice here the 24 is the most recurrent, with frequency of two, thus 24 is the mode.

22 Dispersion

1. the dispersion of two different variable from different distributions of the same features can varies, for example X_1 can be the variable for change in temperature in London (which is cold most of the time), and X_2 can be the variable for change in temperature in Cairo (which varies greatly between season), the dispersion of variable X_2 spans larger extend/range compared with X_1, for example [1,2,3,4,5] is more wide-spread compared to

$$2, 3, 4$$

, how can we measure dispersion?

22.1 variance, and standard deviation

2. variance of variable X is of size m the measurement of how far variable X is disperse, and defined as such:

$$\sigma = \frac{\sum_{i=1}^{m}(x_i - \mu)^2}{m} \tag{55}$$

3. we can express the variance as the normalization, or average of the sum of demeaned entries, this represents the average of how far each entry from the mean μ

4. standard deviation is defined as $\sqrt{\sigma}$

22.2 percentile

5. the **n**th percentile is a threshold percentage before which **n**th percent of date falls behind.
6. for example what is the 25th percentile is 100 for a variable X? this means that 25% of the data are less than 100.

22.3 range

7. the difference between the highest, and lowest entries values for variable X is coined the range, for example the X = [3,6,9,12], the range of $x = 12 - 3 = 9$.

23 Inter Quartile Range IQR

23.1 Outliers

1. outliers are few entries with values greatly deviating from the mean.
2. let's taken an example, consider class of 5 students Omar is 22, Asmaa is 23, Sara is 34, Magid is 26, while Saad is 105, what is the average age of the students in the class?

$$\mu = \frac{22 + 23 + 34 + 26 + 105}{5} = 42 \tag{56}$$

3. note 42 is larger than most students in the class, how can this makes any sense? does the mean in this case represent most of the students? No, this is due to the very old age of Saad being 105! if we removed this entry from the data set, what is the average age in the class once again?

$$\mu = \frac{22 + 23 + 34 + 26}{4} = 26.25 \tag{57}$$

4. does this make any sense? is it close most of the students?
5. there is a way to represent most of the data?
6. let's define IQR, it's a bounding box around the data between the **25**th percentile/first Quartile, and **75**th percentile/third Quartile while the **50**th/the median in between, and the Inter Quartile **Range** is the range from first quartile to the third quartile, this range captures most of the data and at least 50% of the data for uniform distribution, so we can safely clam that most of the data resides within the boundaries of IQR.

23.2 Outliers detection

7. a test made by statisticians for outliers detection is based of this assumption those entries further away from the median are considered Qutliers, but how much further? any entry outside the following range is considered outlier:

$$[1^{st}Quartile - 1.5 * IQR \rightarrow 3^{rd}Quartile + 1.5 * IQR] \tag{58}$$

8. so far we have learned to derive general conclusion over the whole data set using the tools of Central Tendency, and Dispersion we seen so far, How can we derive conclusions on the relation between different variables in the data-set?

24 Correlation

1. for two variables X, Y, how var those two variables are correlated?
2. please keep in mind that positive correlation means as X increase Y also increase, and the reverse, and negative correlation means as X increase, Y decreases, and the reverse is also true.
3. let's take few examples:
   - what is the correlation between Weight(X), and Height(Y) of humans? are they correlated? positively, or negatively?
   - what is the correlation between Educational Level(X), and Salary(Y)? is it positive or negative?
   - what is the correlation between Smoking(X), and Life Expectancy(Y)? is it positive or negative

24.1 Pearson correlation

4. pearson's score is within -1, and 1, -1 means absolutely negatively correlated variables, and being highest positive correlation possible, 0 means no correlation at all, and values within]0-1] means positive corelation, while [-1,0[means negative correlation.
5. definition:

$$r = \frac{\sum_{i=1}^{m}(X - \mu_x)(Y - \mu_y)}{(m-1)\sigma_x\sigma_y} \tag{59}$$

24.2 Covariance

6. Covariance is less neat compared to Pearson correlation, it's the same equation without the normalization by the standard deviation of both variables, meaning the range of correlation is between $-\infty$, and ∞

$$S_{xy}^2 = \frac{\sum_{i=1}^{m}(X - \mu_x)(Y - \mu_y)}{(m-1)} \tag{60}$$

25 distribution

1. Distribution if a probability function of all possible outcomes.
2. there are wide range of distribution as there are wide range of function, the distribution can be uniform meaning each even occur with the same probability as each other event, or completely non-uniform, and almost indescribable, there are several of studied distributions, the most common in nature, and almost all field of sciences is the Normal/Gaussian distribution, also known as Bell curve.
3. assume f(x) is the function of height, and x is the variable for Egyptian males**What is the probability of Egyptian male being 177 cm tall?** what if i told you that this is the average height for Egyptian male, can you make a guess? is it 0.3 (5

4. how to reformulate the question to be more realistic, or sensible? recall the definition of **pdf**, or the density mass function as integral defined over a range, or the area under the curve within that range, $\int_a^b f(x)dx$

5. once again what is the probability for Egyptian male to be around 177cm? how can you formulate this mathematically? using pdf we can write it as such

$$\int_{167.5}^{177.5} f(x)dx \tag{61}$$

6. now we are certainly sure that this is much larger than zero, and it's interpreted as follow what is the probability of Egyptian male's height being within

$$167.5 - 177.5$$

   cm that is 177cm.

7. what is the probability of Egyptian male being of any height? first how to formulate this mathematically?

$$\int_{-\infty}^{\infty} f(x)dx \tag{62}$$

8. the answer is of course 1

25.1 Gaussian distribution

9. the bell curve is described by the standard mean, and the standard deviation.

10. the mean of the Gaussian distribution is happens to be the mode, and median.

25.2 Examples on the Gaussian distribution

11. why the Gaussian is so important?
    - the arrival of train at the station is distributed normally.
    - human intelligence anywhere in the world is normally distributed.
    - the quality of manufacturing fits normal distribution.

12. can you think of more examples?

25.3 empirical rule

13. the empirical rule indicate that:
    - 68
    - 95
    - 100

14. the Gaussian distribution for variable x is defined as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{63}$$

15. note e is Euler constant 2.7, σ is the standard deviation, while μ is the mean of variable x, π is the famous constant 3.14!
16. also notice there is a negative sign on the power.

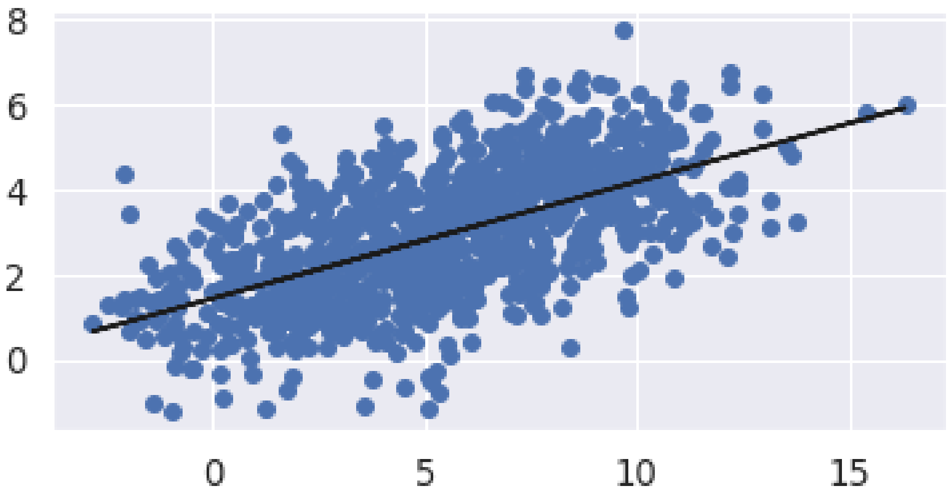
how random noise around a line looks like?

[27]:

```
from utils import *
```

[28]:

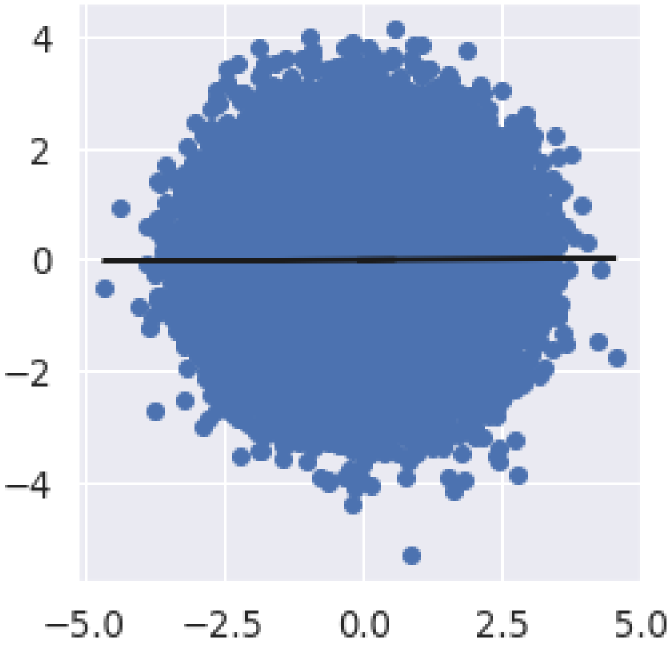
```
from numpy.random import randn
X = np.linspace(1, 10, 1000) + randn(1000)*2
Y = np.linspace(1, 5, 1000) + randn(1000)
plot_correlated_data(X, Y)
print(np.cov(X, Y))
```



```
[[10.98071753  3.02229812]
 [ 3.02229812  2.25218324]]
```

[29]:

```
X = randn(100000)
Y = randn(100000)
plot_correlated_data(X, Y)
print(np.cov(X, Y))
```

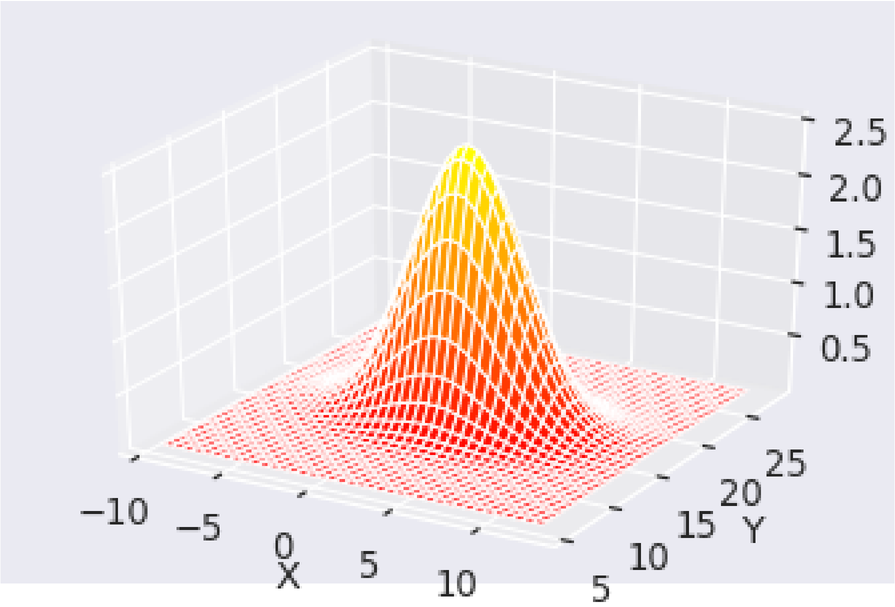


```
[[1.00261493 0.00493068]
 [0.00493068 1.00126455]]
```

here is how the gaussian looks like?

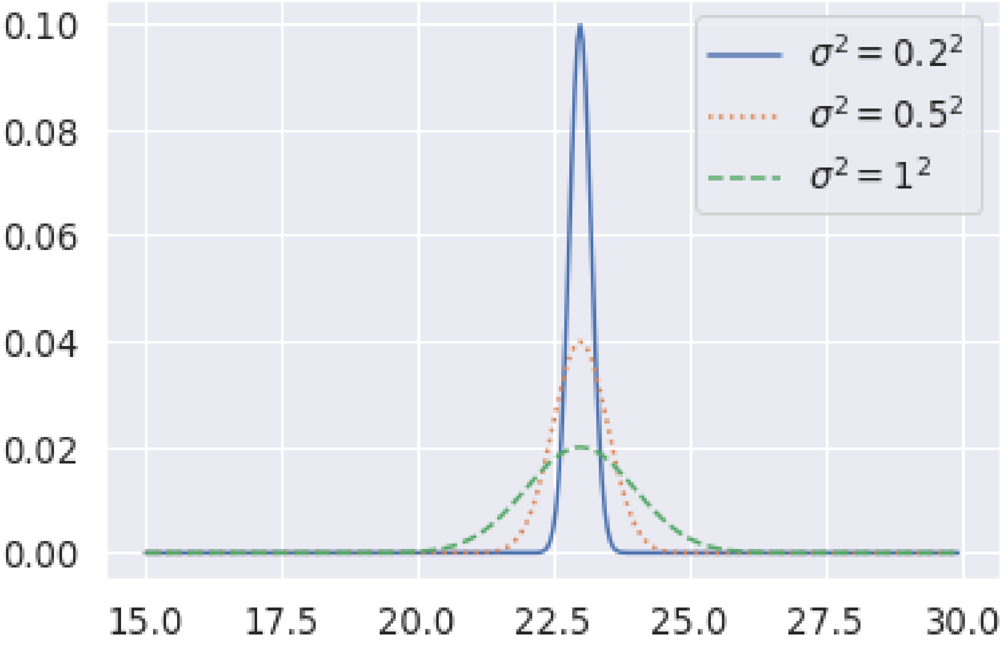
[30]:
```
mean = [2., 17.]
cov = [[10., 0.],
       [0., 4.]]

plot_3d_covariance(mean, cov)
```



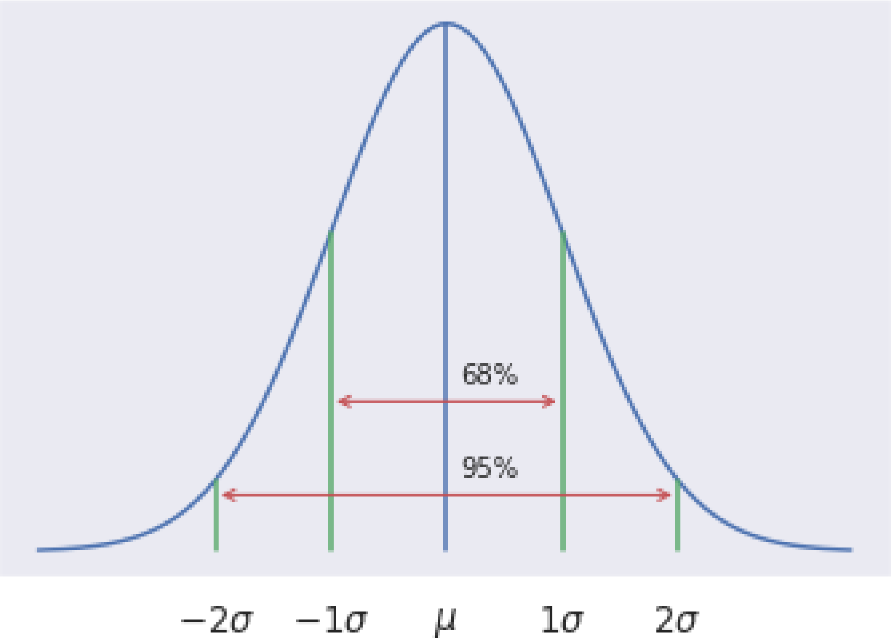
[31]:

```
xs = np.arange(15, 30, 0.05)
plt.plot(xs, gaussian(xs, 23, 0.2**2), label='$\sigma^2=0.2^2$')
plt.plot(xs, gaussian(xs, 23, .5**2), label='$\sigma^2=0.5^2$', ls=':')
plt.plot(xs, gaussian(xs, 23, 1**2), label='$\sigma^2=1^2$', ls='--')
plt.legend();
```



25.4 the imperical rule

```
[32]: display_stddev_plot()
```



26 Z-score

1. the z-score the normalization of the demeaned sample by the standard deviation, defined as:

$$\frac{x-\mu}{\sigma} \tag{64}$$

2. Here is an example of how a z-score applies to a real life situation and how it can be calculated using a z-table. Imagine a group of 200 applicants who took a math test. George was among the test takers and he got 700 points (X) out of 1000. The average score was 600 (μ) and the standard deviation was 150 (σ). Now we would like to know how well George performed compared to his peers.
3. Z-score for George is $\frac{700-600}{150} = 0.67$, from the z-table this is equivalent to the 75
4. note that using **pdf** function over the Gaussian distribution function we introduced the following table can be calculated.

26.1 negative z-table

<img src='https://www.ztable.net/wp-content/uploads/2018/11/negativeztable.png'>

26.2 positive z-table

27 Interval Estimate

1. consider the following problem, we are interested in the covid19 statistics, a team of researchers, started gathering samples, and their tasks is to estimate the percentage of true positive.

2. let's assume that the team have gathered sample from population of 1000 person at random, in a country of 100 million, that is 0.001%, in reality a research is usually inducted on sample of perhaps 200 person, so 1000 is a large sample, can we trust the results of this experiment, and generalize the results to the whole population?

3. instead we can calculate the same results with certain confidence rate, let's say, the domain of **confidence rate** in this experiment is 95% (note that this assumption varies from domain to another), hence the uncertainty is $1 - confidence$ let's define $\alpha = 1-$**confidence rate**, so in this case $\alpha = 1 - 95\% = 0.05$

4. note that a Gaussian distribution is almost perfectly symmetric, and the 95% confidence is the first 2σ around the mean, while the other 5% is distributed on the tails, 2.5% on the right tail, and another 2.5% on the other left tail, let's coin this the name $\alpha/2 = \frac{\alpha}{2}$

5. now we are interest in the estimation at uncertainty $\sigma/2$, how can we calculate that?

27.1 Reliability factor

6. reliability factor, it's an indicator of the threshold of acceptable certainty, and denoted by $Z_{\alpha/2}$, or in other words the Z-score for the uncertainty $\alpha/2$.

7. but how this indicate the Reliability?

8. let's take very high confidence rate of 99% the z-score for it is around -4 on the left hand side of the distribution, and 4 on the right hand side of the distribution, on the other hand, the if we have low certainty as 50% the equivalent z-score is 0!, so the higher the confidence rate, the higher the reliability factor.

27.2 Confidence Interval

9. confidence interval for variable x is the answer for this question, instead making a point estimate $\bar{x}$, we evaluate the latter with how var we are uncertain about the results using the **Reliability Factor** $Z_{\sigma/2}$, and **Standard Error** $\sigma_{\bar{x}}$, defined as follows:

$$\bar{x} \pm Z_{\sigma/2} * \sigma_{\bar{x}} \tag{65}$$

28 Hypothesis Testing

- assume a research in conducting a research, academically what hypothesis the unbiased researcher ought to adopt before and after the experiments?

- the academic starts with null hypothesis, and based off research he, or she can adopt the alternative if the experiment shows otherwise.

28.1 Null against Alternative Hypothesis

- **Null Hypothesis**: it's the first hypothesis of equality, for example that the patient is sick, or the blood pressure is ≥ 100, or ≤ 80.
- **Alternative Hypothesis**: it's the final hypothesis of inequality, for example the patient isn't sick, or the blood pressure is < 100, or > 80.
- assume there is an experiment, after which is measure the uncertainty, and compare it with the original α of the null hypothesis.
- if the uncertainty of the new experiment against the null hypothesis is lower than the uncertainty of the null hypothesis, then the believe/certainty in the alternative hypothesis is higher, and thus we reject the null hypothesis, and adopt the alternative one.
- how can we interpret the uncertainty?

28.2 p-value

- **p-value**: under the assumption that the null hypothesis is correct, it's the probability of obtaining test results at least as extreme as the observed results.
- how p-value in interpreted? a very small p-value means that the observed extreme outcomes are very unlikely, thus the null hypothesis is rejected, on the other hand, for a large p-value the null hypothesis is asserted that the observed outcomes are highly likely.
- how to express p-value mathematically? p-value is the z-score from the z-table, for example p-value of 3.4 correspond to 99% z-score meaning that the confidence in the observed data is extremely high, and the conducted experiment assert the null hypothesis, on the other hand -3.4 z-value has corresponding 0.03% z-score, this means that the believe in the null hypothesis is very unlikely.
- what threshold at which we determine what the z-score is low, or high? actually this changes from domain to another, and specific to each problem, for each problem the researcher decide confidence rate, or level of certainty, and compare p-value against α, if **p-value $<\alpha$** then the null hypothesis is rejected.
- **Example**: Current average waiting period for the customers who call the customer service helpline is 100 seconds with a standard deviation of 20 seconds. Certain changes were recently done to the IVR menu options as well as the overall customer service processes. After a week, the management picked-up a sample of 100 calls and found that the average waiting period was 95 seconds. Have the process implementations resulted in the waiting period reduction?
- **Null hypothesis**: there is no change in the waiting period.
- **Alternative hypothesis**: the waiting period has reduced.
- **significance Level** is assumed for this experiment to be 95%, thus $\alpha = 5\%$
- parameter read: $\mu = 100$, $\sigma = 20$, $N = 100$, $\bar{X} = 95$
- let's compute the parameters of the experiment: $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}} = 2$, and $z = \frac{\bar{X}-\mu}{\sigma_{\bar{x}}} = -2.5$
- the p-value is the corresponding z-score, from the z-table p-value is 0.62%

- $0.62 < 5$, thus the null hypothesis is rejected, and the we can deduce that the waiting period has reduced.

29 mathematics of Gradient Descent (case study Logistic regression)

30 definitions

1. we need an estimate function $\hat{y}$ for the input x, and weight prameters w$\in R^{n_x}, b \in R$.
2. logstic function is $\hat{y} = p(y = 1|x)$, and can be defined as follows: $\hat{y} = \sigma(w^T x + b)$, where the sigma function is defined by $\sigma(z) = \frac{1}{1+e^{-z}}$, and notice when z $\to \infty, \sigma = 1, z \to -\infty, \sigma = 0$.
3.

31 cost function

1. starting with a estimation linear forward model $\hat{y}$, we calculate the difference between our estimate, and the real value y, and through optimization we try to minimize the difference, or loss/cost through gradient descent, then we update our model's parameters.
2. loss function is minimizing the difference between estimation $\hat{y}, y$, and can be defined as least squre $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)$, but least squares leads to non-convex loss function(with multiple local minimums).
3. there are different loss functions, but the most efficient is that which maximize the difference. we can define $P(y|x^{(i)}, \theta) = h(x^{(i)}, \theta)^{y^{(i)}}(1 - h(x^{(i)}, \theta)^{1-y^{(i)}}$, to increase the sensitivity to the training set we take the likelihood function, as the loss, $L(\theta) = \prod_{i=1}^{m} P(y|x^{(i)}, \theta)$.
4. one final step in our model is that as m gets larger L tend to go to zero, to solve this we define the average sum of log-likelihood, or loss function to be our Cost function.
5. we multiply by -1 since the sum of the log-likelihood function is negative.
6. the Cost function $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} log(h(x^{(i)}, \theta)^{y^{(i)}}(1 - h(x^{(i)}, \theta)^{1-y^{(i)}})$
7. loss function is defined as $L(\hat{y}, y) = -[ylog(\hat{y}) - (1 - y)log(1 - \hat{y})]$, L$\in [0 - 1]$.
8. cost function is defined as the average of loss function $J(w, b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y^{(i)}}, y)$

32 Gradient Descent

1. gradient descent is a way to tune the weighting parameters, the objective is the lean toward the fittest weights with respect to the least cost.
2. iterate through cost function **J** tuning with respect to weight parameters **w**, **b**.
3. iterate through: $w := w - \alpha\frac{\partial J}{\partial w}$, $b := b - \alpha\frac{\partial J}{\partial b}$, for tuning w, b for the least **J** possible, such that α is the learning rate of GD.
4. for simplicity $\partial J/\partial w$ replaced for ∂w, and similarly $\partial J/\partial b$ is replaced for ∂b.

5. forward propagation,
$$\partial w = \frac{\partial J}{\partial L}\frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial z}\frac{\partial z}{\partial w}$$
, similarly
$$\partial b = \frac{\partial J}{\partial L}\frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial z}\frac{\partial z}{\partial b}$$
.

6.
$$\partial L/\partial \hat{y} = \frac{-y}{\hat{y}} + \frac{(1-y)}{1-\hat{y}}$$
,
$$\partial \hat{y}/\partial z = \frac{-e^{-z}}{1+e^{-z}} = \hat{y}(1-\hat{y}).$$

7. $\partial L/\partial z = \hat{y} - y$.

8. then we can deduce that the final iteration gradient descent step after calculating sigma, loss, and cost functions can be
$$w := w - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial b} = \frac{\alpha}{m}X^T(\hat{y}-y)$$
, and
$$b := b - \frac{\alpha}{m}\sum_{i=1}^{m}(\hat{y}-y)$$
.

33 Update parameters

1. we implement the following algorithm with a fixed number of iteration that is customized per application, and tuned by the Engineer, such that each application would require different tuning parameters from which is the iteration number.

2. we iterate the following: update the parameters ω, b, in the back propagation step using $\partial\omega$, ∂b.

3.
$$\omega = \omega - \frac{\alpha}{m}X^T(y-\hat{y})$$

4.
$$b = b - \frac{\alpha}{m}(y-\hat{y})$$

33.1 How to implement those functions in python?

[33]:

```
#initialize vector of shape (dim, 1)
# @return b initialized scalar (corresponds to the bias)
def initialize_with_zeros(dim):
    w = np.zeros((dim, 1))
    b = 0
    return w, b

## Randomly initialize the parameters W, b.
#
# @param n_x size of the input layer
# @param n_h size of the hidden layer
# @param n_y size of the output layer
# @return params dictionary of parameters:
#                     W1 -- weight matrix of shape (n_h, n_x)
#                     b1 -- bias vector of shape (n_h, 1)
#                     W2 -- weight matrix of shape (n_y, n_h)
#                     b2 -- bias vector of shape (n_y, 1)
def initialize_parameters(n_x, n_h, n_y):
    np.random.seed(2)
    W1 = np.random.randn(n_h, n_x)*0.01
    b1 = np.zeros((n_h, 1))
    W2 = np.random.randn(n_y, n_h)*0.01
    b2 = np.zeros((n_y, 1))
    assert (W1.shape == (n_h, n_x))
    assert (b1.shape == (n_h, 1))
    assert (W2.shape == (n_y, n_h))
    assert (b2.shape == (n_y, 1))
    parameters = {"W1": W1,
                  "b1": b1,
                  "W2": W2,
                  "b2": b2}
    return parameters

## linear activation function
#
# @param w (m,1) weight matrix
# @param X (m,n) input matrix
# @param b (1) bias
# @return z (1,m) linear estimation
def linear(w, X, b):
    z = np.dot(w.T, X) + b
    return z

## sigmoid activation
#
# @param z is the input (can be a scalar or an array)
# @return h the sigmoid of z
def sigmoid(z):
    return 1/(1+np.exp(-1*z))

## compute log likelihood of the logistic regression
#
```

```python
# @param Y (1,m) labeled vector Output
# @param h (1,m) estimated output
# @return J (1) scalar cost function output
def compute_cost(Y, h):
    m=Y.shape[1]
    def compute_loss():
        L=np.dot(Y.T, np.log(h)) + np.dot((1-Y).T, np.log((1-h)))
        return L.squeeze()
    J = -1.0/m * compute_loss()
    return J

## Update weight values with single gradient descent step.
#
# @param w (n,1) weight vector .
# @param dw (n,1) weight vector of dJ/dw.
# @param b scalar bias.
# @param db scalar dJ/db.
# @param alpha scalar is the learning reate.
# @return tuple of new (w,b).
def update_weight(w, dw, b, db, alpha):
    w = w - alpha * dw
    b = b - alpha * db
    return w, b


## Implement Forward, and Backward propagation, the cost function and its gradient .
#
# @param w weights, a numpy array of size (n,1)
# @param b bias, a scalar
# @param X data of size (n,m)
# @param Y true "label" vector {0,1} (1,m)
# @return cost negative log-likelihood cost for logistic regression
# @return dw gradient of the loss with respect to w, thus same shape as w
# @return db gradient of the loss with respect to b, thus same shape as b
def propagate(w, b, X, Y):
    m = X.shape[1]
    z = linear(w, X, b)
    h = sigmoid(z)
    cost = compute_cost(Y, h)
    dw = 1.0/m * (np.dot(X, (h-Y).T))
    db = 1.0/m * np.sum(h-Y)
    assert(dw.shape == w.shape)
    assert(db.dtype == float)
    assert(cost.shape == ())
    grads = {"dw": dw,
             "db": db}
    return grads, cost

## This function optimizes w and b by running a gradient descent algorithm
#
# @param w weights, a numpy array of size (n,1)
# @param b bias, a scalar
# @param X data of shape (n,m)
# @param Y true "label" vector (containing 0 if non-cat, 1 if cat), of shape␣
 ↪(1, number of examples)
# @param num_iterations number of iterations of the optimization loop
# @param learning_rate learning rate of the gradient descent update rule
# @param print_cost True to print the loss every 100 steps
# @return params dictionary containing the weights w and bias b
# @return grads dictionary containing the gradients of the weights and bias␣
 ↪with respect to the cost function
# @return costs list of all the costs computed during the optimization, this␣
 ↪will be used to plot the learning curve.
def gradient_descent(w, b, X, Y, num_iterations=1000, alpha=0.001, print_cost =␣
 ↪False):
    costs = []
    for i in range(num_iterations):
        grads, cost = propagate(w, b, X, Y)
        dw = grads["dw"]
        db = grads["db"]
        w,b = update_weight(w, dw, b, db, alpha)
        # track cost
        if i % 100 == 0:
            costs.append(cost)
        # Print the cost every 100 training iterations
        if print_cost and i % 100 == 0:
            print ("Cost after iteration %i: %f" %(i, cost))

    params = {"w": w,
              "b": b}

    grads = {"dw": dw,
             "db": db}

    return params, grads, costs
```

Motivational example, are you ready?

34 Linear Regression

Linear Regression is a statistical technique which is used to find the linear relationship between dependent and one or more independent variables. This technique is applicable for Supervised learning Regression problems where we try to predict a continuous variable.

Linear Regression can be further classified into two types – Simple and Multiple Linear Regression. In this project, I employ Simple Linear Regression technique where I have one independent and one dependent variable. It is the simplest form of Linear Regression where we fit a straight line to the data.

```
[34]: import numpy as np
      import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns
      from sklearn.model_selection import train_test_split
      from sklearn.linear_model import LinearRegression

      %matplotlib inline
      sns.set(rc={'figure.figsize': [7, 7]}, font_scale=1.2)
```

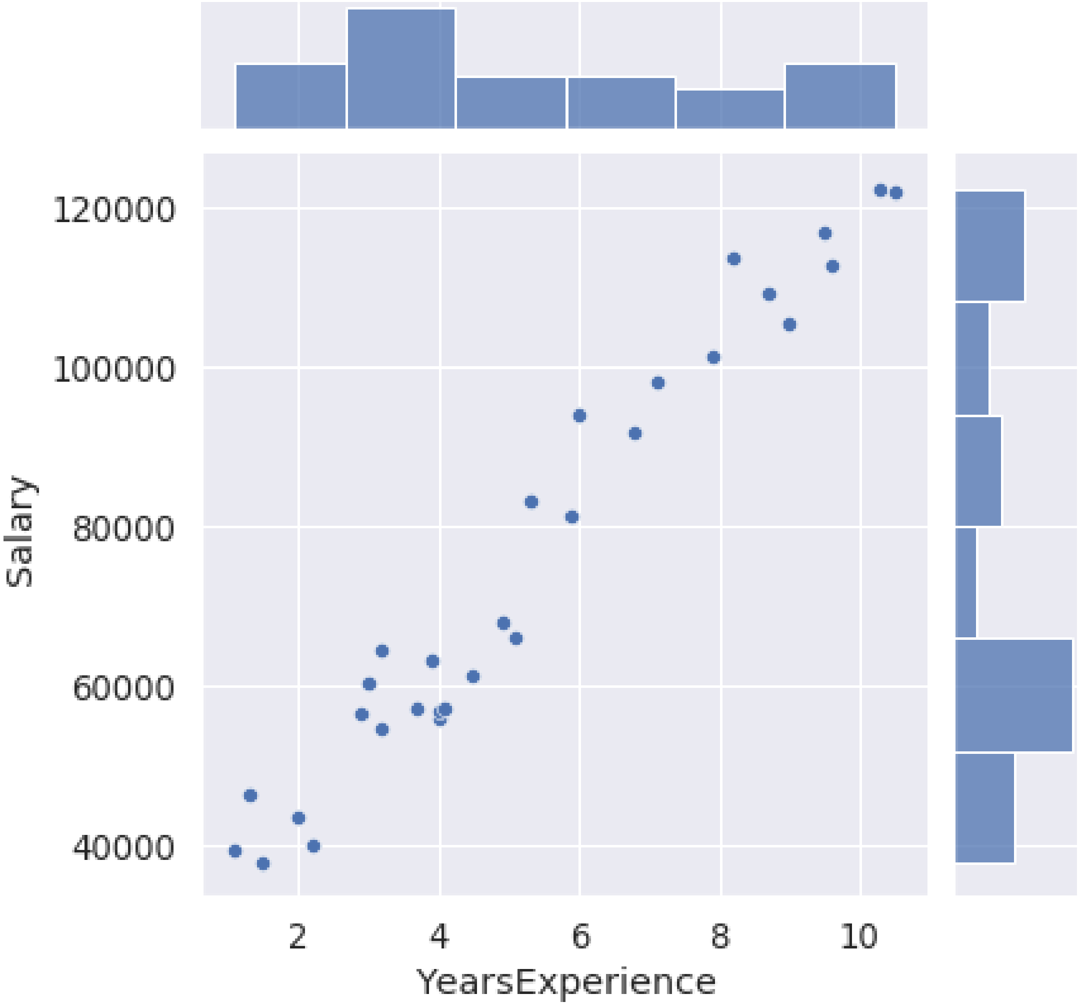
```
[35]: df = pd.read_csv('Salary_Data.csv')
      df
```

```
[35]:     YearsExperience    Salary
      0               1.1   39343.0
      1               1.3   46205.0
      2               1.5   37731.0
      3               2.0   43525.0
      4               2.2   39891.0
      5               2.9   56642.0
      6               3.0   60150.0
      7               3.2   54445.0
      8               3.2   64445.0
      9               3.7   57189.0
      10              3.9   63218.0
      11              4.0   55794.0
      12              4.0   56957.0
      13              4.1   57081.0
      14              4.5   61111.0
      15              4.9   67938.0
      16              5.1   66029.0
      17              5.3   83088.0
      18              5.9   81363.0
      19              6.0   93940.0
      20              6.8   91738.0
      21              7.1   98273.0
      22              7.9  101302.0
      23              8.2  113812.0
      24              8.7  109431.0
      25              9.0  105582.0
      26              9.5  116969.0
      27              9.6  112635.0
      28             10.3  122391.0
```

```
29              10.5  121872.0
```

[36]:
```
sns.jointplot(x='YearsExperience', y='Salary', data=df)
```

[36]:
```
<seaborn.axisgrid.JointGrid at 0x7f260f381880>
```



[37]:
```
x = df['YearsExperience'].values.reshape(-1, 1)
y = df['Salary']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,␣
 ↪random_state=42)
```

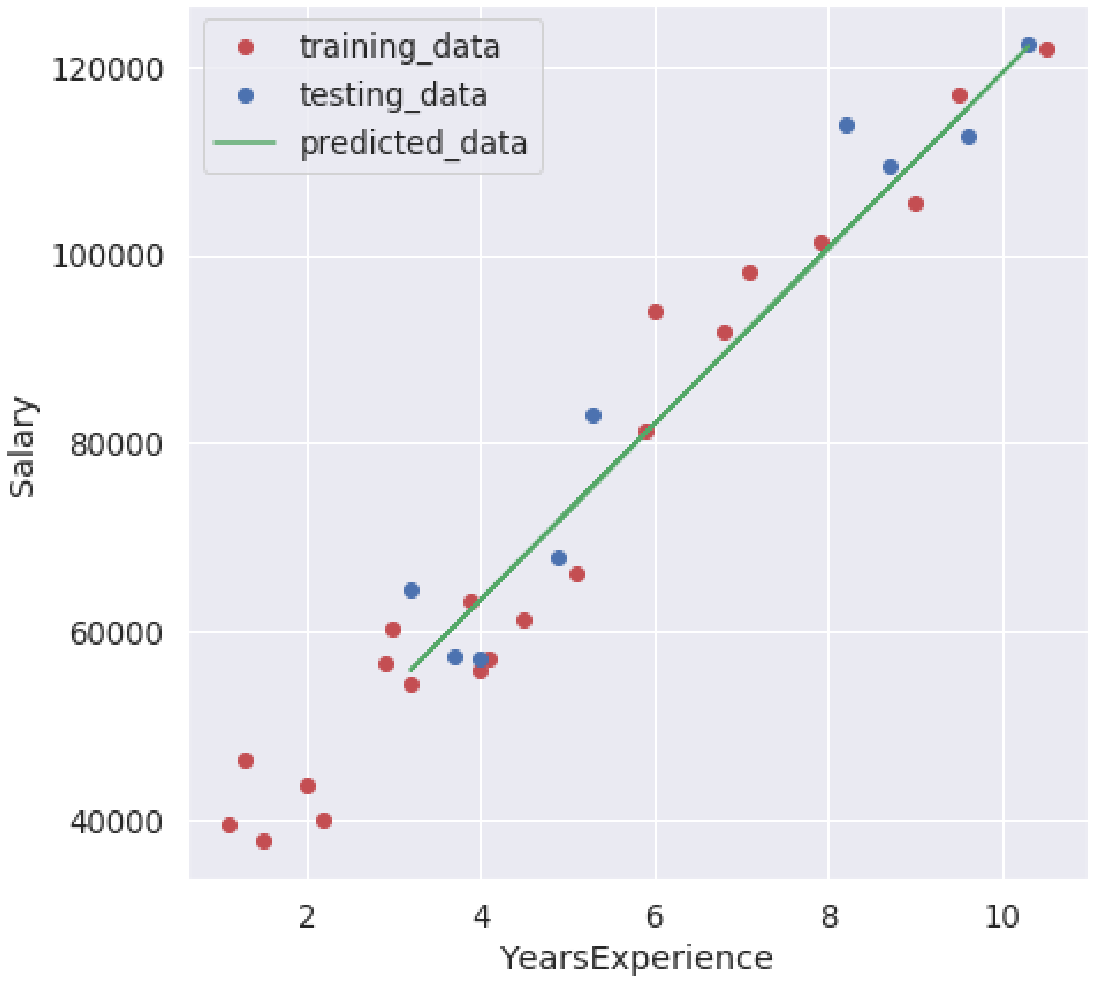
[38]:
```
model = LinearRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

[39]:

```python
plt.plot(x_train, y_train, 'ro', label='training_data')
plt.plot(x_test, y_test, 'bo', label='testing_data')
plt.plot(x_test, y_pred, 'g-', label='predicted_data')
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt.legend()
```

[39]:

```
<matplotlib.legend.Legend at 0x7f26024a46d0>
```



[ ]: