MACHINE LEARNING HOMEWORK 1

		Q1	Q2	Q3	Q4	Q5	Total
Grade	Max	1	1	1	1	1	5 points
	Expected	1	1	1	1	1	5

STUDENT

NAME SURNAME : ALPEREN KANTARCI

STUDENT NUMBER : 504191504

DEADLINE : 17/10/2019

COURSE NAME : MACHINE LEARNING

INSTRUCTOR : YUSUF YASLAN

Report Name : Homework1
Student Name : Alperen Kantarcı

QUESTION 1:

You can see the hand written solution of the question below.

For given table we can find marginal probabilities for Y=0

	X=-1	X=0	X=1
Y=0	0.15	0.3	0.2

$= 0.65$

Q1-) a-) $P(X=1 \mid Y=0) = ?$

when Y=0 and X=1 we can see the probability is $P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)}$

$$P(X=1 \mid Y=0) = \frac{P(X=1 \cap Y=0)}{P(Y=0)} = \frac{0.20}{0.65} = \boxed{0.307}$$

b-) Two random variables are independent if $P(X=a \mid Y=b) = P(X=a)$ for all a, b pairs. So finding an a, b pair that doesn't hold this assumption show that two random variables are dependent.

$P(X=-1 \mid Y=-1) = \frac{0.2}{0.35} = 0.571 \neq 0.35$ So two variables are not independent.
$(P(X=-1))$

c-) $E(5.X + 3.Y.Y) = E(5X) + E(3.Y.Y) = 5E(X) + 3E(Y^2)$

$E(X) = \sum_x x f_X(x) = \sum_x x P(X=x)$ for a discrete random variable.

For $E(f(x)) = \sum . f(x) . P(X=x)$ we can apply this to find $E(Y^2) = \sum . y^2 . P(Y=y)$

value of random var.

From the table

$E(X) = -1.(0{,}2+0{,}15) + 0.(0{,}1+0{,}3) + 1.(0{,}05+0{,}20) = \underline{-0{,}10} = E(X)$

$E(Y) = -1.(0{,}2+0{,}1+0{,}05) + 0.(0{,}15+0{,}3+0{,}20) = \underline{-0{,}35} = E(Y)$

$E(Y^2) = y^2.P(Y=y) = (-1)^2(0{,}2+0{,}1+0{,}05) + 0^2(0{,}15+0{,}3+0{,}20) = \underline{0{,}35} = E(Y^2)$

$$5E(X) + 3E(Y^2) = 5 . -0{,}10 + 3 . 0{,}35 = \boxed{0{,}55}$$

QUESTION 2:
A)

To compute the least squares regression line we first should implement polynomial regression function. When we give polynomial degree as 1 to this function, we would create the line that passes from data points. For given x data points and y values. Assumption of points are :

m is degree and n is number of data points. To calculate the polynomial coefficients for unique least squares solution. We can create matrix A and vector ß. In this model we can assume that ε (epsilon) values are all zeros.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_m x_i^m + \varepsilon_i \; (i = 1, 2, \dots, n)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

$$\widehat{\vec{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\, \mathbf{X}^\mathsf{T}\vec{y},$$

From these multiplications we can find the unique polynomial regression coefficients. By having the coefficients we can make a prediction for every point. Note that for m and n values m < n should always hold to calculate the inverse of the matrix. You can see the Python code for the calculation below.

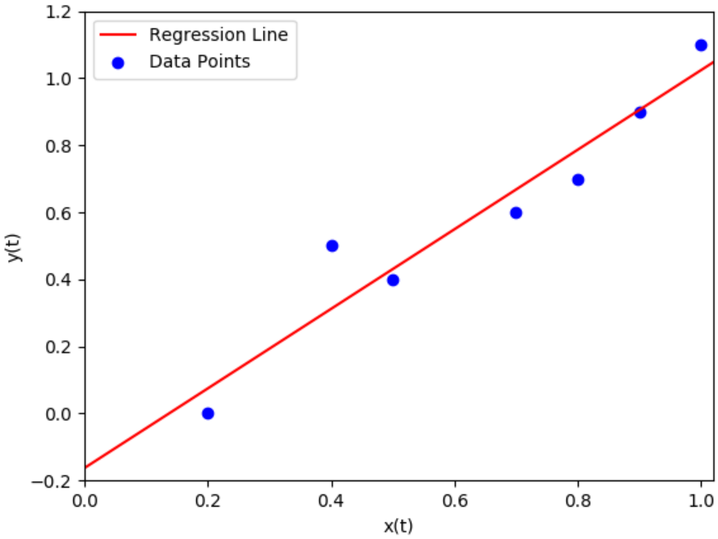
```python
def polynomial_regression(x,y,degree = 1):
    if degree >= len(x):
        raise Exception("Degree cannot be higher than (number of data points - 1)")

    # Create column vector of y
    y = np.array(y).reshape(len(y),1)
    # Create big x matrix which holds values
    X = np.ones((len(x),degree+1))
    for row_ind,row in enumerate(X):
        for column_i in range(len(row)):
            row[column_i] = x[row_ind]**column_i
    b_vect =  np.dot(np.dot(np.linalg.inv(np.dot(X.T,X)),X.T),y)
    return b_vect
```

For any given x vector of data points or just a point and given coefficients of polynomials (b_vect) we can return a prediction list with the function below.

```
# Make predictions by given points
predict_points = lambda x,b_vect: np.sum([beta*(x**i) for i,beta in enumerate(b_vect)])
```

With this function we calculate the coefficients for the degree 1 (just a line). You can see the visualization of the line.



B)

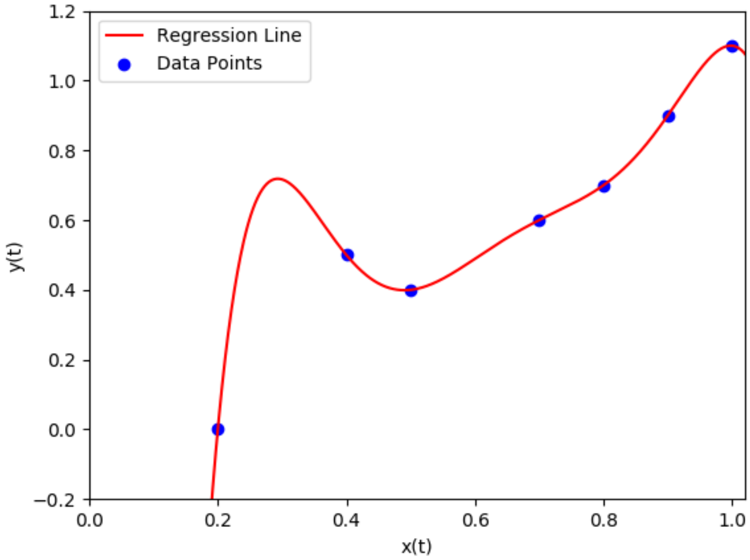
For the given x = -3 and x = 5 data points

Prediction for x = -3 is: -3.723

Prediction for x = 5 is: 5.771

C)

For the polynomial degree 4 you can see the fitted line below.



For leave one out variance comparison, we exclude a data point each time and we calculate the beta parameters with given degree.Then for the point we make an estimation and we calculate the squared error by this estimation.Then for each point we create an error then we calculate the variance of these errors for model given by the degree.We can interpret from this solutions that higher degree of polynomial shows higher variance since it fits to training data not the test data.But also this show us that it will predict bad with new data since it fits on the training data. Therefore the expresiness power of the higher degree polynomial model is higher than degree of 1.Low variance show us that model predictions doesn't change much, it cannot represent complex data points

```
Leave one out variance for polynomial degree 1 (linear) :0.00043524336928233325
Leave one out variance for polynomial degree 4           :0.9810366263815621
```

```
def leave_one_out(x,y,degree):
    errors = []
    for ind,x_i in enumerate(x):
        true_y = y[ind]
        remaining_x = np.delete(x,ind)
        remaining_y = np.delete(y,ind)
        # Train the model with data except the leaved
        b_vect = polynomial_regression(remaining_x,remaining_y,degree)
        errors.append( (true_y - predict_points(x_i,b_vect))**2 )

    return np.var(errors)
```

QUESTION 3:

Q3-) This probability can be calculated by the Bayes formula.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$ where A = "Man sitting next to you is a terrorist"
B = "Man sitting next to you tested positive"

Given that

$P(A=1) = 0{,}01$, $P(A=0) = 0{,}99$

$P(B=1|A=1) = 0{,}95 \rightarrow P(B=0|A=1) = 0{,}05$

$P(B=0|A=0) = 0{,}95 \rightarrow P(B=1|A=0) = 0{,}05$

We are looking for $P(A=1|B=1) = ?$ $P(B=1) = P(B=1|A=1)P(A=1) + P(B=1|A=0)P(A=0)$

From law of total probability we calculate $P(B=1)$

$$P(A=1|B=1) = \frac{P(B=1|A=1)\,P(A=1)}{P(B=1|A=1)P(A=1) + P(B=1|A=0)P(A=0)} = \frac{0{,}95 * 0{,}01}{0{,}95 * 0{,}01 + 0{,}05 * 0{,}99} =$$

$$= \frac{0{,}0095}{0{,}059} = \underline{0{,}161}$$

QUESTION 4:

Q4-) For the novel input, model should assign the class which gives maximum estimated utility. For three class we will calculate the utility for every possibility (predicting true and false) then the highest class will be assigned to input x.

$$EU(\alpha_i|x) = \sum_k U_{ik} P(S_k|x)$$ where i is number of true classes and k is prediction.

By using utility matrix and given probabilities

$EU(c=1|x) = \underbrace{0{,}7*5}_{c=1} + \underbrace{0{,}2*3}_{c=2} + \underbrace{0{,}1*1}_{c=3} = \boxed{4{,}2}$

$EU(c=2|x) = 0{,}7*0 + 0{,}2*4 + 0{,}1*-2 = 0{,}6$

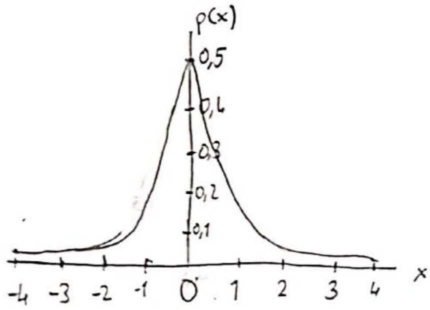
$EU(c=3|x) = 0{,}7*-3 + 0{,}2*0 + 0{,}1*10 = -1{,}1$

We should assign to the class which gi[ves] us the highest expected utility. Therefore more reasonable and best choice is to <u>assigning c=1 to novel x.</u>

QUESTION 5:

Q5-) a-) $p(x)=\frac{1}{2}\exp(-|x|)$ where laplacian distribution is $f(x|\mu,b)=\frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$

We can easily see that mean $\mu=0$ and $b=1$ for this distribution.



Probability of being $x>2$ is equal to the area under the line where $x>2$ in the plot. To find the area we can take integral of the function

$$Q(x>2)=\int_2^\infty \frac{1}{2}\exp(-|x|)=\frac{1}{2}\int_2^\infty \exp(-x)$$

$$=\frac{1}{2}\cdot -e^{-x}+c\Big|_2^\infty=\frac{1}{2}\left(0-\left(-\frac{1}{e^2}\right)\right)$$

$$=\boxed{\frac{1}{2e^2}\approx 0,0676}$$

b-) $p(x)=\binom{n}{x}p^x(1-p)^{n-x}$

$E[X]=\mu=\sum_{i=0}^{n} x_i P_i$, if we plug to binomial

$$E[X]=\mu=\sum_{i=0}^{n} i\binom{n}{i}p^i\cdot(1-p)^i$$

$$=np\sum_{i=0}^{n} i\,\frac{(n-1)!}{(n-i)!\,i!}\,p^{(i-1)}(1-p)^{(n-1)-(i-1)}$$

$$=np\sum_{i=1}^{n}\frac{(n-1)!}{((n-1)-(i-1))!}\,p^{(i-1)}(1-p)^{(n-1)-(i-1)}$$

Factors of binomial coefficient

$$=np\sum_{i=1}^{n}\binom{n-1}{i-1}p^{(i-1)}(1-p)^{(n-1)-(i-1)}$$

k J

$$=np\sum_{k=0}^{n-1}\binom{n-1}{k}p^k(1-p)^{(n-1)-k}$$

$$=np\sum_{k=0}^{J}\binom{J}{k}p^k(1-p)^{J-k}$$

$$=np(p+(1-p))^J$$

$$\boxed{=np}=E[y]$$

$E[x^2]=?$ Let $X=X_1+X_2+\dots+X_n$ where all random variables are independent and coming from Bernoulli distribution. $Var(X_i)=p(1-p)$

$$Var(X)=Var(X_1+X_2+\dots X_n)=Var(X_1)+Var(X_2)\dots+Var(X_n)$$

$$=n(Var(X_1))=\boxed{np(1-p)}$$

$$Var(X)=E[X^2]=np(1-p)$$