INTRODUCTION MACHINE LEARNING

EXERCISE 2

Bauhaus-Universität Weimar

TEACHER:
Johannes Kiesel

GROUP:
Group 16

SUBMITTED BY:

Aaron Perez Herrera
Cesar Fernando Gamba Tiusaba
Chun Ting Lin
Olubunmi Emmanuel Ogunleye

Exercise 1: Machine Learning Basics (0.5+0.5+0.5+0.5+0.5+0.5=3 Points)

a. Name these five concepts: x, x, X, X, X

- **x:** A single feature or variable (e.g., age, height).
- **x:** A feature vector, which represents a single data point with multiple features (e.g. (x1,x2,…,xp)) where each xi is a feature.
- **X:** A set of all feature vectors in a dataset, typically called the design matrix in a dataset, where each row represents a feature vector for one instance, in other words is the feature space, Cartesian product of the domains of the p dimensions of a feature vector x.
- **X:** Random variable (randomness regarding feature x of an object o)
- **X:** Multivariate random variable, random vector (randomness regarding feature vector x of an object.

b. Give the hypothesis space H of linear regression with p features.

In linear regression with ρ features, the hypothesis space H is the set of all linear functions that can be used to model the relationship between the features and the target variable. For ρ features, this is represented as:

H={h(x)=w_0+w_1*x_1+w_2*x_2+⋯ 〖+w〗 _p*x_p | w_0,w_1,w_2,…,w_p ϵ R}

This space contains all possible linear combinations of the features, parameterized by the weights w0,w1,…,wpw_0, w_1, \ldots, w_pw0,w1,…,wp.

c. Explain the Bayes error.

The Bayes error is the lowest possible error rate for a classifier. It represents the irreducible error due to overlapping distributions in the feature space. Even with a perfect classifier, the Bayes error is the error resulting from inherent uncertainty in the data itself, where the best prediction could still be incorrect due to ambiguity in class labels for certain feature values.

d. How can one reduce the Bayes error?

The Bayes error cannot be eliminated, but it can be reduced by:

- Gathering more relevant features that accomplish a better separation of the classes.
- Improving data quality to ensure more distinct separability between classes.
- Increasing the sample size, which could potentially reveal patterns or distributions with less overlap if the classes are indeed separable.

e. Give an example of a dataset D1 with (label) noise: D1 = {. . .}

An example of a data set with noise could be

$$D_2 = \{(x_1, c_1), (x_2, c_2), (x_3, c_2), (x_4, c_3)\}$$

Where C_3 and C_4 are the appropriate values for X_3 and X_4

f. Give this dataset D_2, take a (class-) stratified sample $D_{2,tr}$ of D_2 with $|D_{2,tr}| = 6|$

The dataset is:

$$D_2 = \{(x_1, c_1), (x_2, c_2), (x_3, c_3), (x_4, c_2), (x_5, c_2), (x_6, c_3), (x_7, c_1), (x_8, c_3), (x_9, c_2), (x_{10}, c_2), (x_{11}, c_3), (x_{12}, c_2)\}$$

We start with class counts

- C1: 2 instances (X_1,X_7)
- C2: 6 instances (X_2, X_4, X_5, X_9, X_{10}, X_{12})
- C3: 4 Instances (X_3, X_6, X_8, X_{11})

Proportions

- $C_1 = \frac{2}{12} = 0.167$
- $C_2 = \frac{6}{12} = 0.5$
- $C_3 = \frac{2}{12} = 0.167$

We compute the sample size per class for $D_{2,tr}$

Total size of $D_{2,tr} = 6$, we allocate samples proportionally.

- $C_1: 0.167 * 6 = 1, we\ select\ 1\ sample\ from\ C_1, \{(x_1, c_1)\}$
- $C_2: 0.5 * 6 = 3, we\ select\ 3\ samples\ from\ C_2, \{(x_2, c_2), (x_4, c_2), (x_9, c_2)\}$
- $C_3: 0.333 * 6 = 2, we\ select\ 2\ samples\ from\ C_3\ \{(x_3, c_3), (x_6, c_3)\}$

The finals subset $D_{2,tr}$ is.

$$\{(x_1, c_1), (x_2, c_2), (x_4, c_2), (x_9, c_2), (x_3, c_3), (x_6, c_3)\}$$

Exercise 2: Probabilistic Foundation of the True Misclassification Rate (1.5 + 1.5 + 1 = 4 Points)

Consider a sample space Omega = {o {1}, o{2}, o{3}, o{4}, o{5}, o{6}} with six outcomes; i.e., each elementary event {o{i}} corresponds to observing one of six distinct objects. Let X subset R ^ 2 be a feature space, C = {0, 1} be of two classes, and P be a probability measure defined on {P}. Further, let X: Omega -> X and C : Omega -> C be two random variables defined according to this table:

o_i	$P(\{o_i\})$	$\mathbf{X}(o_i)$	$C(o_i)$
o_1	0.1	$(0,1)^T$	0
o_2	0.3	$(0,1)^T$	1
o_3	0.2	$(0,1)^T$	0
o_4	0.2	$(1,0)^T$	1
o_5	0.1	$(1,0)^T$	0
o_6	0.1	$(0,0)^T$	0

a. Specify the joint distribution function p(x, c) := P (X=x, C=c) by completing this table:

X	C	P(x,c)
$(0,0)^T$	0	0.1
$(0,1)^T$	0	0.3
$(0,1)^T$	1	0.3
$(1,0)^T$	0	0.1
$(1,0)^T$	1	0.2

b. Specify the Bayes classifier y*() by completing this table (potentially more than one correct answer):

X	C	P(x,c)
$(0,0)^T$	0	0.1
$(0,1)^T$	0	0.3
$(1,0)^T$	1	0.1

c. Specify the true misclassification rate Err $*$ of the Bayes classifier.

Err* = p(X = (0,1)T, C = 1) + p(X = (1,0)T, C = 0) = 0.3 + 0.1 = 0.4

Exercise 3 : Evaluating Effectiveness (2+1+0=3 Points)

Consider the following family of classification models:

$$y_\pi(\mathbf{x}) = w \cdot x_\pi$$

where w ∈ {1,−1} is a model parameter learned from data, and π ∈ {1, . . . , p} is a hyperparameter selected manually beforehand. During training, the parameter w is chosen according to the simple learning algorithm shown on the left:

Input: Hyperparameter π and dataset D.

Output: Model Parameter w.

Learn(D, π)

1. **Initialize:** $\mathcal{L}_+ = 0, \mathcal{L}_- = 0$
2. **Loop:** For each example $(\mathbf{x}, c_\mathbf{x}) \in D$
   $\mathcal{L}_+ = \mathcal{L}_+ + I_{\neq}(x_\pi, c_\mathbf{x})$
   $\mathcal{L}_- = \mathcal{L}_- + I_{\neq}(-x_\pi, c_\mathbf{x})$
3. **If** $\mathcal{L}_+ \leq \mathcal{L}_-$ **Then**
   Return $w = 1$
   Else
   Return $w = -1$

Hint:
The indicator function $I_{\neq}$ i defined as in the lecturenote slides:

$$I_{\neq}(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{otherwise} \end{cases}$$

Example:
Given
$D = \{((1, -1), 1), ((-1, 1), -1)\}$
we get:
Learn$(D, 1) = 1$ and
Learn$(D, 2) = -1$.

You are given the following dataset D (with $p = 2$):

	x_1	x_2	c
$\mathbf{x}_1$	1	1	1
$\mathbf{x}_2$	-1	1	-1
$\mathbf{x}_3$	1	1	-1
$\mathbf{x}_4$	1	1	1
$\mathbf{x}_5$	1	1	-1
$\mathbf{x}_6$	-1	1	1
$\mathbf{x}_7$	1	1	-1
$\mathbf{x}_8$	-1	-1	-1
$\mathbf{x}_9$	1	1	1
$\mathbf{x}_{10}$	1	-1	1

With DataSet: $D = \{([x_{1,1}, x_{1,2}], C_1), \dots, ([x_{p,1}, x_{p,2}], C_p)\}$

Function: $y'_\pi = w * x_\pi$ Weight: $w = \{1, -1\}$

ζ+ = Acumulator of Missclassifications

ζ- = Acumulator of Correct Classifications

a. Let the hyperparameter π be fixed at π = 1. Using the algorithm Learn given above, train a classifier y1() on all of D, and determine the training error Err(y1(),D).

X train	X1	C
1	1	1
2	-1	-1
3	1	-1
4	1	1
5	1	-1
6	-1	1
7	1	-1
8	-1	-1
9	1	1
10	1	1

	ζ+	ζ-
	0	0+1=1
	0	1+1=2
	0+1=1	2
	1	2+1=3
	1+1=2	3
	2+1=3	3
	3+1=4	3
	4	3+1=4
	4	4+1=5
	4	5+1=6
TOT.	**4**	**6**

$y'_\pi(x)$	$y'_\pi(x) == C$
1	T
-1	T
1	F
1	T
1	F
-1	F
1	F
-1	T
1	T
1	T

ζ+ < ζ-
So the value of w = 1

$$Err(y_\pi(\quad), D) = \frac{misclasified\ D\ \ elements}{D} = \frac{4}{10} = 0.4 \approx 40\%$$

b. Let Dtest = {x8, x9, x10} be the test set. Leaving π = 1 as before, train classifier y'1 () on Dtr = D \ Dtest and determine the holdout error Err(y'1 (),Dtest).

X train	X1	C
1	1	1
2	-1	-1
3	1	-1
4	1	1
5	1	-1
6	-1	1
7	1	-1

	ζ+	ζ-
	0	0+1=1
	0	1+1=2
	0+1=1	2
	1	2+1=3
	1+1=2	3
	2+1=3	3
	3+1=4	3
TOT.	**4**	**3**

ζ+ > ζ-
So the value of w = -1

X test	X1	C
8	-1	-1
9	1	1
10	1	1

$y'_\pi(x)$	$y'_\pi(x) == C$
1	F
-1	F
-1	F

$$Err(y'_\pi(\quad), Dtest) = \frac{misclasified\ Dtest\ \ elements}{Dtest} = \frac{3}{3} = 1 \approx 100\%$$

c. Let Dval 1 = {x1, x2, x3, x4} and Dval 2 = {x5, x6, x7} be the sets used for model selection with k = 2 validation sets (see slides). Determine $\pi*$, and then determine the holdout error for $y\pi*()$.

K = 1 ---- Dvalid = {x1,x2,x3,x4}; Dtrain = {x5,x6,x7}; Dtest = {x8,x9,x10};

X train	X1	C
5	1	-1
6	-1	1
7	1	-1

	ζ+	ζ-
	0+1=1	0
	1+1=2	0
	2+1=3	0
TOT.	3	0

ζ+ > ζ-
So the value of w = -1

X valid	X1	C
1	1	1
2	-1	-1
3	1	-1
4	1	1

$yk1\|$	$yk1 == C$
-1	F
1	F
-1	T
-1	F

$$Err(yk1(\quad), Dvalid) = \frac{misclasified\ Dvalid\ \ elements}{Dvalid} = \frac{3}{4} = 0.75 \approx 75\%$$

X test	X1	C
8	-1	-1
9	1	1
10	1	1

$y'_\pi(x)$	$y'_\pi(x) == C$
1	F
-1	F
-1	F

$$Err(yk1(\quad), Dtest) = \frac{misclasified\ Dtest\ \ elements}{Dtest} = \frac{3}{3} = 1 \approx 100\%$$

K = 2 ---- Dvalid = {x5,x6,x7}; Dtrain = {x1,x2,x3,x4}; Dtest = {x8,x9,x10};

X train	X1	C
1	1	1
2	-1	-1
3	1	-1
4	1	1

	ζ+	ζ-
	0	0+1=1
	0	1+1=2
	0+1=1	2
	1	2+1=3
TOT.	1	**3**

ζ+ < ζ-
So the value of w = 1

X valid	X1	C
5	1	-1
6	-1	1
7	1	-1

$yk2$	$yk2 == C$
1	F
-1	F
1	F

$$Err(yk2(\quad), Dvalid) = \frac{misclasified\ Dvalid\ \ elements}{Dvalid} = \frac{3}{3} = 1 \approx 100\%$$

X test	X1	C
8	-1	-1
9	1	1
10	1	1

$y'_\pi(x)$	$y'_\pi(x) == C$
-1	T
1	T
1	T

$$Err(yk2(\quad), Dtest) = \frac{misclasified\ Dtest\ \ elements}{Dtest} = \frac{0}{3} = 0 \approx 0\%$$

As result we have:

$$\pi^* = \frac{1}{k}\big(Err(yk1(\quad), Dvalid) + Err(yk2(\quad), Dvalid)\big) = \frac{1 + 0.75}{2} = 0.875 \approx 87.5\%$$

So if we only use Hyperplane "π=1" we obtain an approx. error of = 87.5%

$$Err(y'_\pi(\quad), Dtest) = \min\big(Err(yk1(\quad), Dtest), Err(yk2(\quad), Dtest)\big)$$
$$Err(y'_\pi(\quad), Dtest) = Err(yk2(\quad), Dtest) = 0 \qquad \rightarrow With\ a\ weight\ of\ w = 1$$

Exercise 4: Receiver Operating Characteristic (ROC) (2+2.5+2+0.5=7 Points)

Consider the binary classification scenario of spam mail detection with two classes: mail is spam (c = 1, "positive") and mail is not spam (c = 0 "negative").

a. Look up (e.g., on Wikipedia) the following six concepts and define them for the spam mail scenario: true positive, false positive, false negative, true negative, false positive rate, true positive rate.

- **True Positive (TP)**: These are cases in which a spam email is correctly identified as spam by the algorithm. In other words, when is properly send it to spam box.

- **False Positive (FP)**: These are the cases in which a normal (non-spam) email is incorrectly detected as spam by the algorithm. Here the system make user lose a possible valuable email.

- **False Negative (FN)**: This is when a spam email is incorrectly identified as non-spam by the algorithm. In this case, the system fails to detect spam and allows it into the inbox.

- **True Negative (TN)**: This is when a non-spam email is correctly identified as non-spam by the system. In other words, a legitimate email is correctly allowed to go into the inbox of the user.

- **False Positive Rate (FPR)**: This is the proportion of actual non-spam emails that are incorrectly classified as spam. It is calculated with the following formula: (Indicates how often algorithm misclassified normal emails as spam)

$$FPR = \frac{FP}{FP + TN}$$

- **True Positive Rate (TPR)**: Also known as sensitivity or recall, this is the proportion of actual spam emails that are correctly identified as spam by the system. It is calculated as:

$$TPR = \frac{TP}{TP + FN}$$

b. Consider a large dataset D for spam mail detection with balanced class distribution, i.e., P(C = 0) = P(C = 1) = 0.5. Calculate the (expected) false positive rate and true positive rate for each of these classifiers:

- A classifier that classifies every mail as spam

DATA		
TP	1	100 %
FP	1	100 %
FN	0	0 %
TN	0	0 %

$FPR = \frac{FP}{FP+TN} = \frac{1}{1+0} = 1 = 100\,\% \quad \rightarrow 100\%\ of\ Incorrect\ non-spam\ Clasification$

$TPR = \frac{TP}{TP+FN} = \frac{1}{1+0} = 1 = 100\,\% \quad \rightarrow 100\%\ of\ Correct\ spam\ Clasification$

- A classifier that classifies every mail as not spam

DATA		
TP	0	0 %
FP	0	0 %
FN	1	100 %
TN	1	100 %

$FPR = \frac{FP}{FP+TN} = \frac{0}{0+1} = 0 = 0\,\% \quad \rightarrow 0\%\ of\ Incorrect\ non-spam\ Clasification$

$TPR = \frac{TP}{TP+FN} = \frac{0}{0+1} = 0 = 0\,\% \quad \rightarrow 0\%\ of\ Correct\ spam\ Clasification$

– A classifier that classifies every mail correctly

DATA		
TP	1	100 %
FP	0	0 %
FN	0	0 %
TN	1	100 %

$FPR = \frac{FP}{FP+T} = \frac{0}{0+1} = 0 = 0\ \%$ $\rightarrow 0\%\ of\ Incorrect\ non-spam\ Clasification$

$TPR = \frac{TP}{TP+F} = \frac{1}{1+0} = 1 = 100\ \%$ $\rightarrow 100\%\ of\ Correct\ spam\ Clasification$

– A classifier that classifies every mail incorrectly

DATA		
TP	0	100 %
FP	1	100 %
FN	1	100 %
TN	0	100 %

$FPR = \frac{FP}{FP+TN} = \frac{1}{1+0} = 0 = 100\ \%$ $\rightarrow 100\%\ of\ Incorrect\ non-spam\ Clasification$

$TPR = \frac{TP}{TP+FN} = \frac{0}{0+1} = 0 = 0\ \%$ $\rightarrow 0\%\ of\ Correct\ spam\ Clasification$

– A classifier that classifies every mail randomly with equal class probability

Random Classification So:
C = {0, 1} ------- Prob. C(1) = 0,5 = 50% Prob. C(0) = 0.5 = 50%

Value that is X,C(i) = Prob. C(i) * 0,5 (0.5 = Random value 50%)
Prob (spam + true detect) = 0,5 *0,5 = 0.25

DATA		
TP	0.5*0.5 = 0.25	25 %
FP	0.5*0.5 = 0.25	25 %
FN	0.5*0.5 = 0.25	25 %
TN	0.5*0.5 = 0.25	25 %

$FPR = \frac{FP}{FP+T} = \frac{0.25}{0.25+02.5} = 0.5$ $\rightarrow 50\%\ of\ Incorrect\ non-spam\ Clasification$

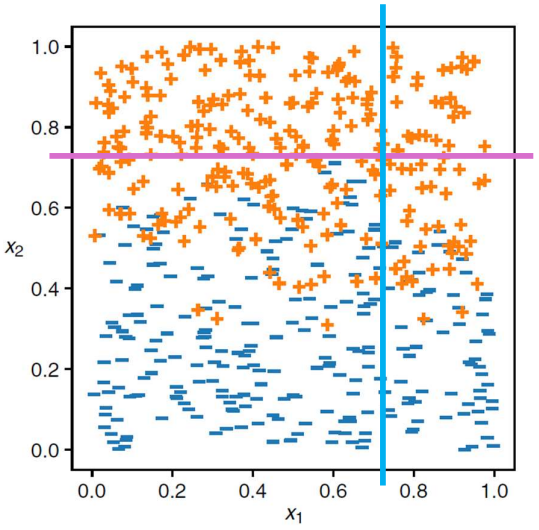
$TPR = \frac{TP}{TP+FN} = \frac{0.25}{0.25+0.25} = 0.5$ $\rightarrow 50\%\ of\ Correct\ spam\ Clasification$

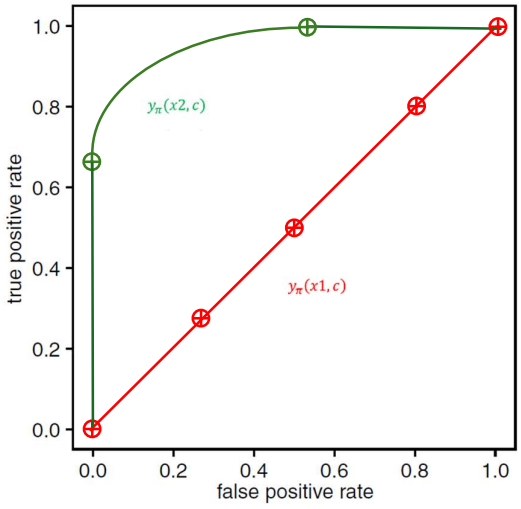
c. Now assume the plot on the left-hand side shows D with spam examples (c = 1) represented as + and non-spam examples (c = 0) represented as −. Consider two classifiers yπ(x) for π ∈ {1, 2} which use a threshold w0 to classify instances solely based on either x1 or x2:

$$y_\pi(\mathbf{x}) = \begin{cases} 1 & \text{if } x_\pi \geq w_0 \\ 0 & \text{otherwise} \end{cases}$$

As one continuously increases w0 from 0 to 1, both classifiers change from classifying every mail as spam to classifying no mail as spam, with the true positive rate and false positive rate changing accordingly. This continuous change of rates for a classifier corresponds to a line in the false positive rate / true positive rate scatter plot (empty plot on the right-hand side), which is known as receiver operating characteristic (ROC) curve. Roughly sketch the ROC curves of y1 and y2.

w0 = 0	----------	all emails C=1 (Is Spam)	P1 = (FPR=1,TPR=1)
w=0.25	----------	P2(x1) = (FPR=0.8,TPR=0.8)	P2(x2)= (FPR=0.5,TPR=1)
w0=0.5	----------	P3(x1) = (FPR=0.5,TPR=0.5)	P4(x2)= (FPR=0.25,TPR=0.8)
w0=0.75	----------	P5(x1) = (FPR=0.3,TPR=0.3)	P5(x2)= (FPR=0,TPR=0.6)
w0= 1	----------	all emails C=0 (Non Spam)	P6 = (FPR=0,TPR=0)





d. Based on the ROC curves you sketched in (c), which classifier do you prefer? Argue solely based on the ROC curves!

I would use the second classifier, because it has a curve which separates the data in a way that provides in almost each weight that can be assigned the lowest value of "FPR" and the highest value of "TPR", thus achieving a lower error in classification rate than its counterpart (that is almost a straight-line function) and therefore a more effective classification.

Exercise 5: Linear Regression (2+1+0+0+0=3 Points)

This table describes four cars by their age, mileage, and stopping distance for a full braking at 100km/h:

Car	Wartburg	Moskvich	Lada	Trabi
Age (year)	5	7	15	28
Mileage (km)	30 530	90 000	159 899	270 564
Stopping distance (meter)	50	79	124	300

a. Determine the linear regression weights wi for predicting the stopping distance from only the age.

		x	y		
n	Car	Age (years)	Stop Distance (m)	x*y	x1^2
1	Wartburg	5	50	5	25
2	Moslvisch	7	79	14	49
3	Lada	15	124	45	225
4	Trabi	28	300	112	784
		55	553	176	1.083

x med	13,75
y med	138,25

$$w_1 = \frac{n*\sum xy - \sum x * \sum y}{n*\sum x^2 - \sum x^2} = 10.5868$$

$$w_0 = \bar{y} - w1 * \bar{x} = -7.3191$$

$$y_{(x1)} = w_0 + x * w_1 = -7.3191 + 10.5868 * x$$

b. Extrapolate the expected average stopping distance for the Lada car (i.e., age = 15 years) using the model from (a).

$$y_{(15)} = w_0 + x * w_1 = -7.3191 + 10.5868 * 15 = 151.4836\, m$$

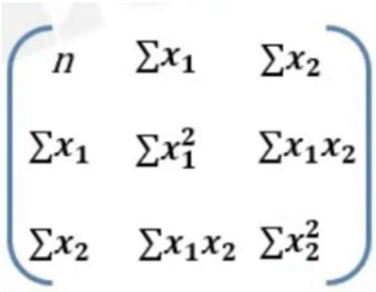
c. Consider the mileage of the cars as an additional variable and repeat (a) and (b) under this setting.

n	Car	Age (years)	Lileage (km)	Stop Distance (m)
1	Wartburg	5	30.530	50
2	Moslvisch	7	90.000	79
3	Lada	15	159.899	124
4	Trabi	28	270.564	300
		55	550.993	553

x1*y	x2*y	x1^2	x2^2
250	1.526.500	25	932.080.900
553	7.110.000	49	8.100.000.000
1.860	19.827.476	225	25.567.690.201
8.400	81.169.200	784	73.204.878.096
11.063	109.633.176	1.083	107.804.649.197

x1 med	13,75
x2 med	137748,25
y med	138,25
k	2
x1, x2	

$\bar{M} * \bar{W} = \bar{Y}$

M = 

$$\sum y = n\beta_0 + \beta_1 \sum x_1 + \beta_2 \sum x_2$$
$$\sum (x_1 y) = \beta_0 \sum x_1 + \beta_1 \sum x_1^2 + \beta_2 \sum (x_1 x_2)$$
$$\sum (x_2 y) = \beta_0 \sum x_2 + \beta_1 \sum (x_1 x_2) + \beta_2 \sum x_2^2$$

$Ec1$: $0.072w_o + 0.0995w_1 + 996.3707w_2 = 1$

$Ec2$: $0.050w_o + 0.2734w_1 + 2739.2764w_2 = 1$

$Ec3$: $0.050w_o + 0.2764w_1 + 2769.1735w_2 = 1$

$Ec3 - Ec2 = Ec4$ $\quad 0.0003w_1 + 29.8971w_2 = 0$ $\quad w_2 = \dfrac{0.003 * w_1}{29.8971}$

$1.44Ec2 - Ec1$: $\quad 0.2942w_1 + 1205.2816w_2 = 0.44$ $\quad w_1 = 1.0599$

$Ec4\ rep.w1$: $\quad w_2 = 0.0001$

$Ec1\ rep.w1\ and\ w2$: $\quad w_0 = 110.4032$

$y_{(x1,x2)} = w_0 + x1 * w_1 + x2 * w_2 = 110.4032 + 1.0599 * x1 + 0.0001 * x2$

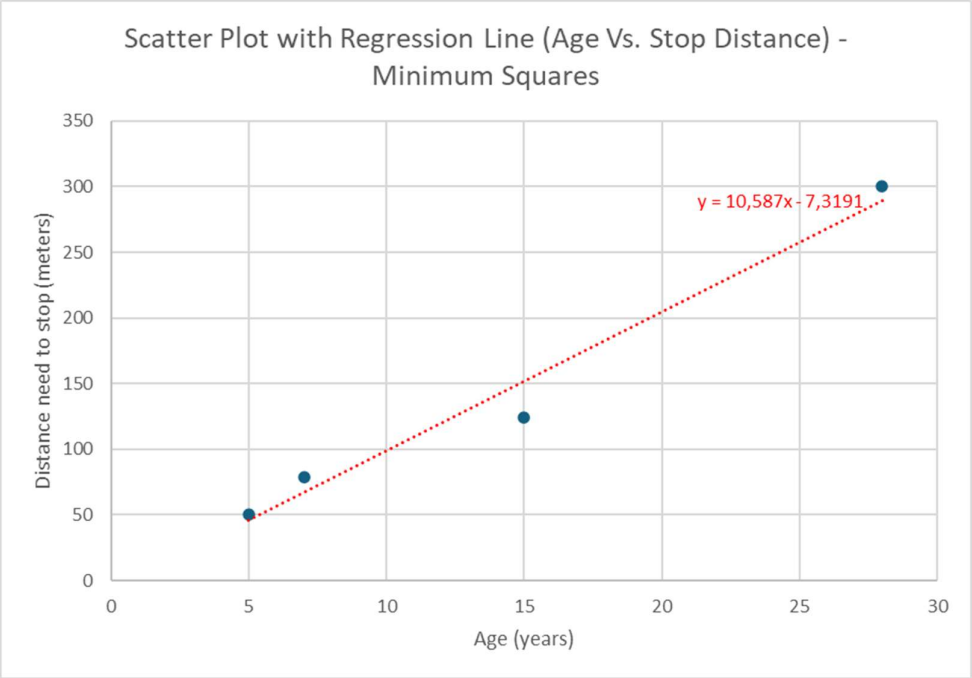
$$y_{(15,159.899)} = 110.4032 + 1.0599 * 15 + 0.0001 * 159.899 = 126.3177\, m$$

d. Draw a scatter plot of the data points, and the linear regression for a variable of your choice (i.e., either age or mileage on the x-axis).

To visualize the data:

1. Plot stopping distance (y-axis) against age (x-axis).
2. Add the regression line:

$$y_{(x1)} = w_0 + x * w_1 = -7.3191 + 10.5868 * x$$



e. Discuss the problems and pitfalls of extrapolation.

The problems of extrapolation in this set of regression lines are the following ones:

- Unreliable Predictions: The linear relationship may not hold outside the observed data range. For instance, predicting stopping distance for a 50-year-old car might yield unrealistic results.
- Limited Data: The dataset has only four cars, making it difficult to capture the true relationship between variables.
- Omitted Variables: Other factors (e.g., brake condition, tire wear) affect stopping distance but are not included.
- Nonlinearity: The actual relationship between age, mileage, and stopping distance might not be perfectly linear, especially for very old cars.
- Overfitting: With a small dataset, the model might perform well on the given data but fail to generalize.

Example: Predicting for cars older than 28 years may lead to unrealistic results.

Exercise 6 : P Basic Data Analysis and Linear Regression (1+2+1+1 Points)

For programming exercises like this one, write code in Python 3.10 or later. Submit all code that you write. You are restricted to built-in Python modules and functions, except NumPy, Pandas, and (for plotting) matplotlib or seaborn. We provide data in tab-separated-value (TSV) format – you can use the built-in csv library's DictReader and DictWriter with delimiter='\t' or Pandas read_csv and to_csv functions with sep='\t' for reading and writing.

Download and use these files from Moodle:

- features-train.tsv: Feature vectors for each example in the training set
- labels-train.tsv: Labels for each example in the training set indicating the class is_human (C = {True, False})
- features-test.tsv: Feature vectors for each example in the test set
-

a) Select two features (e.g. num_words and num_characters) and plot a scatterplot for the examples in the training set between the two features. Color the points according to the class is_human. Submit the plot.
b) Implement the LMS algorithm and use it to compute the weight vector (w0,w1) and add the line of best fit to your plot from (a). Submit your algorithm implementation and the updated plot.
c) Compute the residual sum of squares (RSS) for the weight vector from (b).
d) Use the weight vector from (b) to classify each example in the test set for is_human (C = {True, False}). Write the predicted classes to a predictions-test.tsv in the same format as the labels-train.tsv (columns id and is_human). Submit the file with the predictions.

Check the attached files into the .zip folder to look for each of this points.