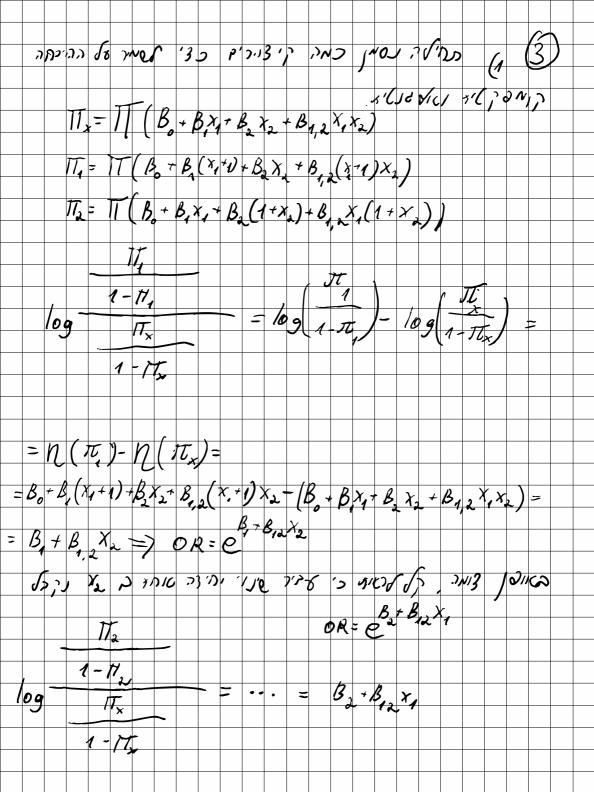
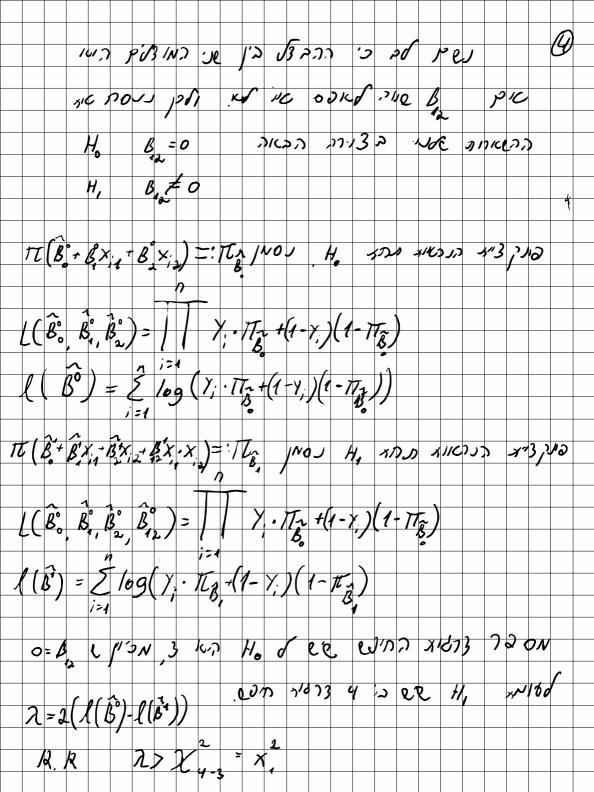


																_			
										, 1	りょ	'),	7	Jß	NU	e	' S) /	
						か	ڔ	ر کی ا	73	Ó	ζ		e	1	H	 	31.)D	
																			<u> </u>
					ها	71)	- 1	16.	1	3	}		l	<u>'</u>	H_1	,	11) <i>(</i>)	<u> </u>
																			<u> </u>
																			<u> </u>



												/ \$						
						J	4	16	1	\mathcal{I}_{z})	` }	ď,	1	رم	Q		<u> </u>
									,	_								
																		<u> </u>
																		<u> </u>
																		<u> </u>
																		<u> </u>



$$S(x, B) \in \mathbb{R}^{n} \neq d \mid S(x, B) \in \mathbb{R}^{n} \Rightarrow PeJ(e^{(S)})$$

$$\mathbb{R}^{n} \Rightarrow T(B) = \mathbb{R}^{n} \Rightarrow PeJ(e^{(S)})$$

$$S(x, B) = \mathbb{R}^{n} \Rightarrow PeJ(e^{(S)})$$

$$\frac{\partial f(x, B)}{\partial B_{0}} \Rightarrow \frac{\partial f(x$$

$$I(\theta)_{i,i} = E \begin{bmatrix} \frac{1}{2B_{i}} & (1 - \pi(B^{T}x))x_{i} \end{bmatrix}$$

$$= E \begin{bmatrix} \frac{1}{2B_{i}} & (1 - \pi(B^{T}x))x_{i} \end{bmatrix} = E \begin{bmatrix} \frac{1}{2B_{i}} & x_{i} \end{bmatrix} IU(B^{T}x)x_{i} \end{bmatrix}$$

$$= E \begin{bmatrix} \frac{1}{2B_{i}} & \frac{1}{2B_{i}}$$

```
from sklearn.linear_model import LogisticRegression
import pandas as pd
import numpy as np
from scipy.stats import norm
```

A.

```
\pi_i = \pi(eta X_i) = \pi(eta_0 + eta_{alcohol_i} * X_{alcohol_i} + eta_{BMI_i} X_{BMI_i} + eta_{age_i} X_{age_i}
```

B.

```
#load the data
df = pd.read_csv("/data/workspace_files/ex4.csv")
X = df[["alcohol", "BMI", "age"]]
X.insert(0, 'one', 1, True)
y = df["chd"]
```

```
reg = LogisticRegression(penalty="none", fit_intercept=False).fit(X, y)
beta_star_hat = reg.coef_[0]
a= [print(f"the beta_{x[0]} is {x[1]}") for x in zip(["zero", "alcohol", "BMI"
```

the beta_zero is -9.906057116302838 the beta_alcohol is 0.019612208107966704

1 of 3

```
the beta_BMI is -0.021686645217299128 the beta_age is 0.20531261330705375
```

C.

```
all_log_probs = reg.predict_log_proba(X)
sum_log_probs = sum([all_log_probs[i][yi] for i, yi in enumerate(y)])
print(f"the log likelyhood is {sum_log_probs}")
```

the log likelyhood is -6.827115831144858

D.

[-1.17187406e+00 7.03496142e-04 5.35554305e-02 -5.50322653e-03] [-5.28539086e-01 -1.78907336e-04 -5.50322653e-03 1.35966443e-02]]

2 of 3

E.

```
X_sample = np.array([1, 8, 27,50])
logit = X_sample@np.array(beta_star_hat).T
p = np.exp(logit)/(1+np.exp(logit))
```

0.48273980924207194

```
se = (X_sample @ var @ X_sample)**0.5
z_alpha = norm.ppf(0.975)
almost_ci = [p-se*z_alpha, p+se*z_alpha]
ci = [1/(1+np.exp(-x)) for x in almost_ci]
print(se)
print(f"the confindence interval is {ci}")
```

0.7668761040764567 the confindence interval is [0.26496707223277977, 0.8792969588274144]

3 of 3