

✓ Question 1:

Imagine you are developing a simple machine learning model to predict whether a student will pass a course based on the following five features:

1. Study Hours per Week: Number of hours the student studies per week.
2. Class Attendance (%): Percentage of classes attended by the student.
3. Previous Exam Score (%): Score in the previous exam.
4. Participation in Group Study: Whether the student participates in group study sessions (1 for Yes, 0 for No).
5. Submission of Assignments (%): Percentage of assignments submitted by the student.

Dataset: Below is an example dataset with 10 students (rows), 5 features, and a binary target label (Pass/Fail). The target label is 1 if the student is predicted to pass, and 0 if they are predicted to fail.

Study Hours per Week	Class Attendance (%)	Previous Exam Score (%)	Participation in Group Study	Submission of Assignments (%)	Pass/Fail
10	90	85	1	100	1
5	60	70	0	80	0
15	95	90	1	100	1
2	30	50	0	40	0
12	85	88	1	90	1
8	75	65	0	70	0
14	92	93	1	95	1
3	50	55	0	60	0
11	88	80	1	85	1
6	65	60	0	75	0

```
import pandas as pd
data = [[10, 90, 85, 1, 100, 1],
        [5, 60, 70, 0, 80, 0],
        [15, 95, 90, 1, 100, 1],
        [2, 30, 50, 0, 40, 0],
        [12, 85, 88, 1, 90, 1],
        [8, 75, 65, 0, 70, 0],
        [14, 92, 93, 1, 95, 1],
        [3, 50, 55, 0, 60, 0],
        [11, 88, 80, 1, 85, 1],
        [6, 65, 60, 0, 75, 0]]
df = pd.DataFrame(data, columns=['Study Hours per Week', 'Class Attendance (%)', 'Previous Exam Sc
df
```



	Study Hours per Week	Class Attendance (%)	Previous Exam Score (%)	Participation in Group Study	Submission of Assignments (%)	Pass/Fail
0	10	90	85	1	100	1
1	5	60	70	0	80	0
2	15	95	90	1	100	1
3	2	30	50	0	40	0
4	12	85	88	1	90	1
5	8	75	65	0	70	0
6	14	92	93	1	95	1
7	3	50	55	0	60	0
8	11	88	80	1	85	1
9	6	65	60	0	75	0

```
df.to_csv('student_data.csv', index=False)
df
```



	Study Hours per Week	Class Attendance (%)	Previous Exam Score (%)	Participation in Group Study	Submission of Assignments (%)	Pass/Fail
0	10	90	85	1	100	1
1	5	60	70	0	80	0
2	15	95	90	1	100	1
3	2	30	50	0	40	0
4	12	85	88	1	90	1
5	8	75	65	0	70	0
6	14	92	93	1	95	1
7	3	50	55	0	60	0
8	11	88	80	1	85	1
9	6	65	60	0	75	0

```
x = df.drop('Pass/Fail', axis=1)
y = df['Pass/Fail']
```

```
import numpy as np
list=[2.0,3.0,7.0,1.0,5.0]
weight = np.array(list)
weight
```



```
array([2., 3., 7., 1., 5.])
```

```
def update(p,e,i):
    for j in range(len(weight)):
        weight[j]=weight[j]+p*e*float(x.iloc[i:i+1,j:j+1].values)
```

```
print(x.iloc[0:1,1:2])
```

```
⇒ Class Attendance (%)
   0                      90
```

```
learning_rate = 0.001
def error(yp,i):
    e=y[i]-yp
    if(e!=0):
        update(learning_rate,e,i)
```

```
yp=-1
import numpy as np
for k in range(6):
    for i in range (len(x)):
        z=np.dot(x[i:i+1],weight)
        if z>0:
            yp=1
        else:
            yp=0
            error(yp,i)
            #print(yp)
```

```
print("Trained Weights:", weight)
```

```
def perceptron_predict(x, weights):
    z = np.dot(x, weights)
    return 1 if z >= 0 else 0
```

```
correct_predictions = 0
for i in range(len(x)):
    prediction = perceptron_predict(x[i:i+1], weight)
    if prediction == y[i]:
        correct_predictions += 1
```

```
accuracy = correct_predictions / len(x)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

```
⇒ Trained Weights: [2. 3. 7. 1. 5.]
   Accuracy: 50.00%
```

✓ 2nd Question

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
```

```
dp = pd.read_csv('/content/diabetes_data_upload.csv')
print(dp.head())
```

```

➡

```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	\
0	40	Male	No	Yes		No	Yes	No
1	58	Male	No	No		No	Yes	No
2	41	Male	Yes	No		No	Yes	Yes
3	45	Male	No	No		Yes	Yes	Yes
4	60	Male	Yes	Yes		Yes	Yes	Yes

	Genital thrush	visual blurring	Itching	Irritability	delayed healing	\
0	No	No	Yes	No	Yes	
1	No	Yes	No	No	No	
2	No	No	Yes	No	Yes	
3	Yes	No	Yes	No	Yes	
4	No	Yes	Yes	Yes	Yes	

	partial paresis	muscle stiffness	Alopecia	Obesity	class
0	No	Yes	Yes	Yes	Positive
1	Yes	No	Yes	No	Positive
2	No	Yes	Yes	No	Positive
3	No	No	No	No	Positive
4	Yes	Yes	Yes	Yes	Positive

```

dp['Gender'] = dp['Gender'].map({'Male': 1, 'Female': 0})
dp['class'] = dp['class'].map({'Positive': 1, 'Negative': 0})

```

```

dp.replace({'Yes': 1, 'No': 0}, inplace=True)

```

```

X = dp.drop(columns=['class']) # Drop the target column
y = dp['class'] # The target column (diabetes risk: Positive/Negative)

```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

```

```

model = Perceptron(max_iter=1000, tol=1e-3, random_state=42)

```

```

model.fit(X_train, y_train)

```

```

➡
▼ Perceptron
Perceptron(random_state=42)

```

```

y_pred = model.predict(X_test)

```

```

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")

```

```

➡ Accuracy: 90.38%

```

DONE BY K SHISHIR BHAGATH [2303A52164]