Program 4:

Build Logistic Regression Model for a given dataset

Screenshot:

DATE:
PAGE

2/4/25

Q Consider binary classification problem where we want to predict whether a student will pass or fail based on their study hours. logistic regression model has been trained, and learned parameters are $a_0 = -5$ (intercept) & $a_1 = 0.8$ (coefficient of study hours)

(a) Write logistic regression eqn for this problem

$$p(y=1|x) = \frac{1}{1+e^{-(-5+0.8x)}}$$

(b) calculate probability that student who studies for 7 hrs will pass

$$p(pass) = \frac{1}{1+e^{-0.6}} = 0.6479 \quad \left(\begin{array}{l} x = 7 \\ z = -5 + 0.8(7) = 0.6 \end{array} \right)$$

(c) Determine predicted class for this student based on threshold of 0.5

if $p(pass) \geq 0.5$, student will pass else he will fail

(d) Consider $z = (2,1,0)$ for three classes, apply softmax fn to find prob of values three classes

$$softmax(z_0) = \frac{e^{z_k}}{\sum_{i=1}^{k} e^{z_i}}$$

$$p(1) = \frac{e^2}{e^2+e^1+e^0} = 0.665$$

$$p(2) = \frac{e^1}{e^2+e^1+e^0} = 0.295$$

$$p(3) = \frac{e^0}{e^2+e^1+e^0} = 0.09$$

DATE: PAGE:

Q After building logistic regression models, answer following questions:

(1) For dataset file "HR_comma_sep.csv"

(i) Which variables did you identify as having a direct and clear impact on employee retention? why?

(ii) What was accuracy of your logistic regression model? Do you think this is good accuracy? why or why not?

(2) For 200 dataset

(i) Did you perform any data preprocessing steps? If yes, what were they, and why were they necessary?

(ii) Were there any missing or inconsistent values in dataset? How did you handle them?

(iii) What does confusion matrix tell you about performance of your model

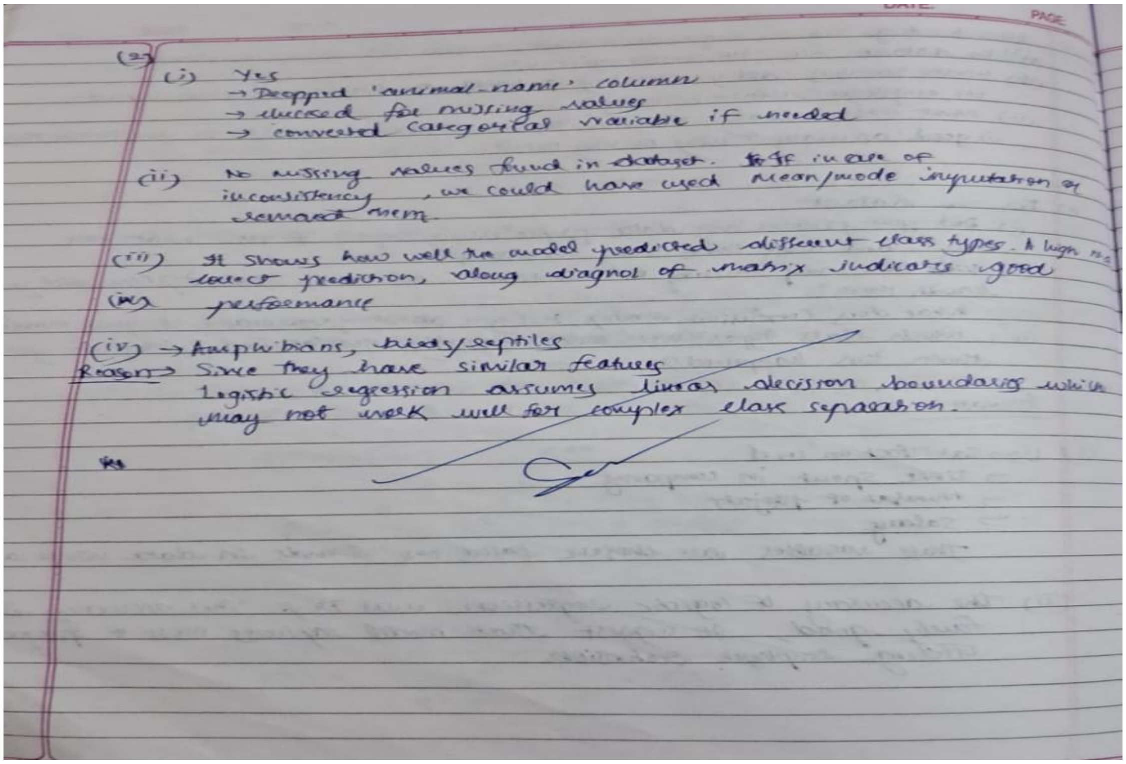
(iv) which class types were most frequently misclassified? Why do you think this happened?

Answer:

(1) (i) → Satisfaction level
→ Time spent in company
→ Number of projects
→ Salary

These variables are chosen based on trends in data visualization

(ii) The accuracy of logistic regression was 78%. This accuracy is fairly good, it suggest that model captures most of properties affecting employee retension.


(2)
(i) Yes
→ Dropped 'animal_name' column
→ checked for missing values
→ converted categorical variable if needed

(ii) No missing values found in dataset. If in case of inconsistency, we could have used mean/mode imputation or removed them.

(iii) It shows how well the model predicted different class types. A high no. of correct prediction along diagnol of matrix indicates good performance

(iv) → Amphibians, birds/reptiles
Reason → Since they have similar features
Logistic regression assumes linear decision boundaries which may not work well for complex class separation.

Code:

#LogisticRegression_Multiclass.ipynb

 # Import necessary libraries

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn import metrics
import matplotlib.pyplot as plt
```

```
# Load the Iris dataset

iris =  pd.read_csv("/content/iris (2).csv")

iris.head()


X=iris.drop('species',axis='columns')# Features (sepal length, sepal width, petal length, petal width)

y = iris.species # Target labels (0: Setosa, 1: Versicolor, 2: Virginica)


# Split the dataset into 80% training and 20% testing

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


# Initialize the Multinomial Logistic Regression model

# Use 'multinomial' for multi-class classification and 'lbfgs' solver

model = LogisticRegression(multi_class='multinomial')


# Train the model on the training data

model.fit(X_train, y_train)


# Make predictions on the test data

y_pred = model.predict(X_test)


# Calculate the accuracy of the model on the test data

accuracy = accuracy_score(y_test, y_pred)


# Display the accuracy
```

```
print(f"Accuracy of the Multinomial Logistic Regression model on the test set: {accuracy:.2f}")


confusion_matrix = metrics.confusion_matrix(y_test, y_pred)


cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels =
["Setosa", "Versicolor", "Virginica"])


cm_display.plot()

plt.show()
```

Binary Classification:

```
#LogisticRegression_Binary.ipynb


# Commented out IPython magic to ensure Python compatibility.

import pandas as pd

from matplotlib import pyplot as plt

# %matplotlib inline

#"%matplotlib inline" will make your plot outputs appear and be stored within the notebook.


df = pd.read_csv("/content/insurance_data (1).csv")

df.head()


plt.scatter(df.age,df.bought_insurance,marker='+',color='red')


from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =
```

```
train_test_split(df[['age']],df.bought_insurance,train_size=0.9,random_state=10)
X_train.shape

X_test

from sklearn.linear_model import LogisticRegression
model = LogisticRegression()

model.fit(X_train, y_train)

X_test

y_test

y_predicted = model.predict(X_test)
y_predicted

model.score(X_test,y_test)

model.predict_proba(X_test)

y_predicted = model.predict([[60]])
y_predicted
```

```
#model.coef_ indicates value of m in y=m*x + b equation
model.coef_


#model.intercept_ indicates value of b in y=m*x + b equation
model.intercept_


#Lets defined sigmoid function now and do the math with hand
import math
def sigmoid(x):
  return 1 / (1 + math.exp(-x))


def prediction_function(age):
   z = 0.127 * age - 4.973 # 0.12740563 ~ 0.0127 and -4.97335111 ~ -4.97
   y = sigmoid(z)
   return y


age = 35
prediction_function(age)


"""0.37 is less than 0.5 which means person with 35 will not buy the insurance"""
```