Machine Learning Lab Sheet 05

In this lab, you will delve into the world of **neural networks** and **Bayesian networks for classification**. The primary focus will be on implementing a simple neural network using Python and exploring the fundamentals of Bayesian networks. We will use Google Colab for a collaborative and interactive coding experience.

Part 1: Neural Network for Classification

Task 1.1: Implementing a Simple Neural Network

Import necessary libraries:

Begin by importing essential libraries for neural network implementation.

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, classification_report
```

Load and preprocess the dataset:

- Load a suitable classification dataset (e.g., the Iris dataset).

Load the Iris dataset

```
iris = load_iris()
X = iris.data
y = iris.target
```

Implement a Simple Neural Network:

- Create and train a simple neural network model using **MLPClassifier** from scikit-learn.

Split the data into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Implement and train a simple neural network
neural_network = MLPClassifier(hidden_layer_sizes=(50,), max_iter=1000)
```

```
neural_network.fit(X_train, y_train)
# Make predictions on the test set
y_pred = neural_network.predict(X_test)
# Evaluate the model's accuracy and provide a classification report
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, target_names=iris.target_names)
print(f"Accuracy: {accuracy:.2f}")
print("Classification Report:")
print(report)
```

Part 2: Bayesian Network for Classification

Import necessary libraries:

- Start by importing libraries for Bayesian network implementation.

```
# Import necessary libraries for Bayesian network
# (Note: You may need to install a Bayesian network library such as pgmpy using pip)
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from pgmpy.estimators import ParameterEstimator
from pgmpy.models import BayesianModel6

# Import necessary libraries for Bayesian network
# (Note: You may need to install a Bayesian network library such as pgmpy using pip)
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from pgmpy.estimators import ParameterEstimator
from pgmpy.models import BayesianModel
```

Load and preprocess the dataset:

- Load the same classification dataset used in Task 1.1.

Implement Bayesian Network for Classification:

Create and train a Bayesian network model using a suitable library (e.g., pgmpy).

```
# Split the data into training and testing sets

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


# Implement Bayesian Network for Classification (Example using pgmpy)

# Note: This is a simplified example; Bayesian networks are more commonly used for probabilistic reasoning.

model = BayesianModel([('Feature1', 'Target'), ('Feature2', 'Target'), ('Feature3', 'Target'), ('Feature4', 'Target')])

model.fit(pd.DataFrame(np.column_stack([X_train, y_train]), columns=['Feature1', 'Feature2', 'Feature3', 'Feature4', 'Target']))

# Make predictions on the test set (Note: Simplified prediction for illustration)

y_pred = model.predict(pd.DataFrame(X_test, columns=['Feature1', 'Feature2', 'Feature3', 'Feature4']))

# Evaluate the model's accuracy (Note: This is a simplified accuracy calculation)

accuracy = np.mean(y_pred['Target'].values == y_test)

print(f"Accuracy: {accuracy:.2f}")
```

Homework Tasks:

- Summarize your findings, including dataset details, preprocessing steps, model implementations, parameter exploration, and a comparative analysis of neural and Bayesian networks. Discuss any challenges encountered.
- Compare the performance of the neural network and Bayesian network on the same dataset. Discuss the strengths and limitations of each approach.
- Experiment with different parameters of the neural network, such as the number of hidden layers and neurons, and observe their impact on performance.