109550198 卜銳凱

Introduction to Data Science
HW10
Report

This report describes the analysis of the dataset from file ‘Churn_Modelling.csv'.

The uploaded Python file contains code for implementing an Artificial Neural Network (ANN) using TensorFlow and other libraries. Here's an overview of the code structure:

Library Imports: The code begins by importing necessary libraries - numpy, pandas, and tensorflow. It also checks the TensorFlow version.

Data Preprocessing:

- Dataset Import: It imports a dataset named 'Churn_Modelling.csv'. The dataset seems to be used for a churn prediction model.
- Feature Selection: It selects features from the dataset, excluding the first three columns and the last column.
- Target Variable: The last column of the dataset is assigned as the target variable (y).
- Encoding Categorical Data: It encodes categorical data, particularly the "Gender" column, using LabelEncoder from sklearn.

More Data Preprocessing:

- One-Hot Encoding: The "Geography" column of the dataset undergoes one-hot encoding. This is done using ColumnTransformer and OneHotEncoder from sklearn, suggesting that "Geography" is a categorical variable with more than two categories.
- Dataset Splitting: The dataset is split into training and test sets using train_test_split from sklearn, with 20% of the data allocated for testing. This is a common practice in machine learning to evaluate the performance of the model.
- Feature Scaling: The code snippet ends with a comment about feature scaling, which is a critical step in preparing data for neural networks.

The remaining part of the code completes the implementation of the Artificial Neural Network (ANN) for churn prediction. Here's a detailed overview:

Feature Scaling:

The code uses StandardScaler for scaling the features. This is essential for neural network models as it helps in faster convergence.

Building the ANN:

- **Initialization**: The ANN is initialized as a sequential model using TensorFlow's Keras API.

- **Adding Layers**: The model includes an input layer and two hidden layers, each with 6 units and 'relu' activation. There's also an output layer with 1 unit and 'sigmoid' activation, which is typical for binary classification tasks like churn prediction.

Training the ANN:

- **Compilation**: The ANN is compiled with the 'adam' optimizer and 'binary_crossentropy' loss function, which is standard for binary classification problems. The metric used for evaluation is 'accuracy'.
- **Model Training**: The model is trained on the training set for 100 epochs with a batch size of 32.

Making Predictions and Evaluating the Model:

- **Single Observation Prediction**: There's a detailed example showing how to use the model to predict whether a specific customer will leave the bank.
- **Test Set Prediction**: The model predicts outcomes for the test set. These predictions are thresholded at 0.5 to determine the binary outcome.
- **Confusion Matrix and Accuracy**: The code calculates and prints the confusion matrix and accuracy score for the test set predictions.

The code is well-structured and covers key aspects of building and evaluating a neural network model using TensorFlow. It includes data preprocessing, encoding of categorical variables, dataset splitting, feature scaling, model building, training, and evaluation. This implementation is typical for binary classification problems in machine