L-Università ta' Malta
Faculty of Information &
Communication Technology

Department of
Computer Information
Systems

# CIS3190 Data Analytics

**Coursework for Academic Year 2024-25**

John Abela

## Overview

The objective of this coursework is to create a Jupyter Notebook or Lab.
The work will consist of three stages:

Stage 1 – EDA Exploratory Data Analysis.
Stage 2 – Creation of MLP using Keras and Tensorflow.
Stage 3 – Post Factum Data Analysis

This coursework will involve use a neural network implemented in Keras for a classification task and to then build dashboards for the visualization of the results using Matplotlib and Seaborn. The first dashboard will be statistics on the dataset itself, and the second dashboard will be statistics on the results of the training. All development will be using Python and various packages including Keras, Tensorflow, Numpy, Pandas, Matplotlib, and Seaborn. The dataset will be split in training/testing on an 80:20 ratio. The testing dataset will be tested after every batch and the performance (accuracy for classification) is plotted. This will help identify when/if the model starts to overfit.

## Data Set – Telco Customer Churn Dataset (5 hours)

- **Description**: A dataset from a telecommunications company used to predict customer churn (i.e., whether a customer will leave the company).
- **Size**: 7,043 rows.
- **Features**: Contains categorical (e.g., contract type, payment method) and numerical features (e.g., tenure, monthly charges).
- **Task**: Binary classification (churn or no churn).
- **Source**: Available on Kaggle

## Stage 1 – EDA and Data Preparation

- Load and clean the dataset.
- Conduct basic EDA: summary statistics, distribution plots, correlations (if applicable).
- Split the dataset into training (80%) and testing (20%) sets.
- Build the first dashboard using Matplotlib and Seaborn that visualizes:
  - Basic statistics (mean, median, mode).
  - Distribution of features.

## Stage 2 – Creation of the MLP using Keras and TensorFlow (6 hours)

**Objective**: Implement a simple neural network using Keras and train it.

- Define the neural network architecture using Keras (e.g., a basic fully connected model).
- Compile the model, selecting the appropriate loss function and optimizer:
  - Use accuracy for classification tasks.
- Train the model with the following:
  - Test performance after every epoch (or batch, as per your design).
  - Track metrics such as accuracy (classification).

## Stage 3 – Post Factum Data Analysis

**Objective**: Visualize the performance of the model over time and interpret the results.

- Build a second dashboard with:
  - Loss/accuracy curves (training vs testing).
  - Confusion matrix for classification.
  - Highlight any overfitting behaviour by examining the gap between training and testing curves.
- Provide insights into the model's performance:
  - When does overfitting occur, if at all?
  - Are there any actions you could take to improve the model (early stopping, regularization)?
- Try to improve the model through hyperparameter tuning (e.g., learning rate, number of layers, activation functions).
- Optionally apply techniques like dropout to reduce overfitting.

## Technical Notes

- You do not need a GPU for this coursework.
- 16GB or RAM may be required although it might also suffice to have 8GB of RAM.
- You may use other programming languages. Email me if you want to use another programming language.

## Notes

- Please create a **single Jupyter Notebook or Lab** for all your work. You can then cut-and-paste all the code from the Jupyter Notebook or Lab into your coursework document.
- You must submit a **single PDF document** that includes all the code, and any comments and observations, to the VLE to the designated area. Do not use images for the code.
- You must also upload the Notebook or Lab, and all the code and scripts, as a **single ZIP file** to the VLE to the designated area. Do not use password protection.
- The deadline for submission is **Friday 17th January 2025 at Midnight (23:59)**.
- Do not forget the **Plagiarism Declaration Form** which you have to sign.
- **15% of the marks** will be assigned to your use of Generative AI. Add an appendix to your documentation explaining how Generative AI was used, the prompts you used, and how Generative AI improved the quality of your submission.

Please email me at john.abela@um.edu.mt if you encounter any problems or difficulties.

**This study unit is assessed 40% by coursework and 60% by exam on campus.**