AI & Computational MIS Task 3 Report

Walid Mougharbel – 6370524

# I- Dataset, Supervised, and Unsupervised Learning Models

While working on this project, I had to learn various new concepts: finding patterns within a dataset, learning a new Python framework for designing and implementing dashboards, and building an application out of it. However, the toughest part of the task was to find a suitable dataset to study. My initial goal was to predict whether jobs will be replaced by AI in the future but couldn’t find an open-source study that offers a dataset, as it might still be too early. I then started looking for data about unemployment rate and found a dataset that displays unemployment rate in different Canadian states.

The data begins on January 1st, 1976, and ends on January 5th, 2015. I chose Linear Progression as the supervised learning model to predict possible future rates of unemployment in different Canadian states on specific dates. Linear regression seemed like a better choice here given the type of prediction that I wanted to make. I used one independent variable (time) to predict a dependent variable (unemployment rate).

To capture a pattern in unemployment and time, I chose the KMeans clustering model for the unsupervised learning model. The model managed to capture a relationship between time and unemployment. The model showed that unemployment rates peak right after the second third of January and peak around the final third or December and first third of January. That is probably due to the holidays.

# II- Dashboard and Plotly Dash

Dash is a Python framework that is built on React.js and Plotly.js, and it is used to create dashboards and visualize data. I started by creating an application for each Task and route it using Flask.

For the first Task, I had to create a scatter plot displaying relationships between different features. I used the **scatter()** function from **plotly.express**. The default features are Sepal Length and Petal Length. The user can select different features for x and y axes to visualize different features. The data set used here was the Iris dataset, which is used by importing the **load\_iris** function from **sklearn.datasets**.

For the second task, I needed to visualize the price of different stocks in function of time on a line chart. I started by extracting the data using the API of **yfinance** ticker. To avoid making multiple API calls, I placed the data in a dataframe and saved in a **.csv** file called **stock\_data.csv**. I would then make an API request only if the said file didn’t exist. I also calculated the relative strength index (RSI) of the stock and add an option to display it on the chart or not. To select specific time frames, I added a range by including the **xaxis\_rangeslider\_visible=True** option in the **plotly.express update\_layout()** function.

In the third Task, I chose a dataset about unemployment rate in different Canadian states. I prompted the user to select a state and date to predict the unemployment rate on, and my Linear Regression model would predict it for that specific date, and display the state, date, predicted unemployment rate, and the root mean squared error (RMSE) in a table. The model scored a RMSE of 3.35. Given the simplicity of the model, the RMSE value is reasonable. However, exploring different algorithms might further reduce the error. I was also required to use an unsupervised learning model to capture a pattern in the data, so I used KMeans clustering. Linear Regression and KMeans can be respectively found in the **sklearn.linear\_model** and **sklearn.cluser** modules. I got the dataset from the Canadian Open Government Portal.

The main application file (**main.py**) is the file where the app is running. It redirects the user to the Task 1 page with the option to change to other tasks’ respective pages by clicking on buttons that are linked to these tasks.

To run the dashboard, install the following dependencies using **pip**: dash, scikit-learn, pandas, numpy, plotly, datetime, and yfinance, then, run **“python3 main.py”**, and navigate to [**http://127.0.0.1:8050**](http://127.0.0.1:8050) on your browser.

# III- Results and Insights

The predicted values showed that the unemployment rate dips towards the beginning of the year and slightly soars during the year in most Canadian states. The results of the KMeans clustering displayed this connection between time and unemployment as well. This study shows that employment increases during the holidays (probably due to Christmas markets or holiday-themed markets) and slightly decrease right after.

# IV- Acknowledgements

I would like to acknowledge the usage of the Canadian Open Government Portal for the dataset to complete this study.

Canadian Open Government Portal: <https://open.canada.ca/data/en/dataset/f212a64f-92f0-430c-a04f-06436b1239d2/resource/2ec8b2de-c9be-48b9-8fab-c0ee2d96d482>