**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | Data preparation and visualization, Machine Learning, Statistics and Programming |
| **Assessment Title:** | Foreign Nationals in Employment |
| **Lecturer Name:** | David McQuaid, Muhammad Iqbal, Taufique Ahmed, Sam Weiss |
| **Student Full Name:** | Fernando Tupa Leniz |
| **Student Number:** | 2023341 |
| **Assessment Due Date:** | 10/11/2023 |
| **Date of Submission:** | 10/11/2023 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

**Introduction**

In today's data-driven world, the application of machine learning models, statistical knowledge, and my coding skills in Python using Jupyter notebooks has become central to my work in the field of data analytics and machine learning. The objective of this CA aligns perfectly with my professional interests, as it entails predicting the weekly salaries of employees within the industrial sector. To accomplish this task, I'll be using a robust and informative dataset from the Central Statistics Office (CSO) of Ireland, which contains a wealth of information about employees, their characteristics, and their earnings.

You can explore the code and documentation related to this project in my GitHub repository: <https://github.com/FernandoDataAnalitycs> .

The ability to predict weekly salaries within the industrial sector is of great importance to a wide range of stakeholders, including policymakers, businesses, and employees themselves. Accurately forecasting salaries is not just a statistical exercise; it holds the potential to uncover valuable insights into wage determination, wage disparities, and labour market dynamics.

In this first CA, I will delve into data preprocessing and visualization techniques to clean and explore the dataset, making it ready for machine learning model development.

The role of data preparation and visualization is fundamental in uncovering patterns and relationships within the dataset. Visualization tools will enable me to gain a deeper understanding of the data's underlying structures and correlations, thereby informing feature selection and model creation. In addition, I will look for the best way to execute coding using python libraries or methods that can help me to type less and do more.

This CA not only serves as a practical exercise but is closely aligned with the work I do in my professional capacity. By the end of this project, I aim to create a machine learning model capable of predicting weekly salaries with a high degree of accuracy. This will contribute to a deeper understanding of the labour market in Ireland and serve as a powerful tool for informing labour market policies and guiding organizations in making data-driven decisions regarding employee compensation.

Throughout this assignment, I will apply various machine learning algorithms, statistical techniques, and my coding skills to develop and fine-tune our predictive model. The ultimate goal is to leverage data-driven insights to make a tangible impact in the field of labour market analysis.

**Data preparation and visualisation part:**

**1)**

**Machine Learning part:**

**1)**

In this first part, I think CRISP-DM (Cross-Industry Standard Process for Data Mining) is better than KDD (Knowledge Discovery in Databases) or SEMMMA (Sample, Explore, Modify, Model, and Assess) for a data science project.I will justify with some explanation and examples:

First of all, CRISP-DM is considered an industry standard and is widely adopted in the field of data science. It provides a structured approach that is recognized and accepted across industries. In contrast, KDD lacks the practical guidelines and widespread acceptance that CRISP-DM offers. SEMMA, developed by SAS for its software products, is less known outside of SAS-centric environments, limiting its applicability in organizations using various tools. For example, consider that a huge corporation wants to adopt a standardized data science approach across various departments. In such a case, CRISP-DM's industry-wide recognition and acceptance make it the preferred choice.

Secondly, CRISP-DM is a flexible methodology that can be adapted to different types of data science projects and is not tied to specific tools or techniques. KDD, is more theoretical and less prescriptive in terms of specific project phases and techniques. SEMMA is being closely integrated with SAS tools, may not be easily adaptable for organizations using alternative data science platforms. For instance, a startup with limited resources that seeks agility in adopting different data science techniques based on project requirements. CRISP-DM's flexibility allows them to do so seamlessly.

Thirdly, CRISP-DM places strong emphasis on an iterative process, encouraging revisiting and refining earlier stages as new insights and data become available. KDD does not explicitly emphasize iteration and lacks detailed guidance on revisiting stages. SEMMA also lacks a strong focus on iteration, which may be less suitable for projects requiring continuous improvement. For example, in the healthcare industry, where new patient data continually becomes available, a data science project aimed at improving patient outcomes benefits from CRISP-DM's iterative approach.

Finally, CRISP-DM places a strong emphasis on understanding the business problem and aligning data science goals with business objectives. This is crucial for the success of a data science project. KDD, on the other hand, focuses more on data mining techniques and may not provide as strong a link to business understanding. SEMMA does include elements of business understanding but is not as comprehensive as CRISP-DM in this regard. For instance, In the retail industry, where understanding customer behaviour is essential for improving sales and marketing strategies, CRISP-DM's focus on business understanding proves advantageous.

To continue with this first part, I would choose the supervised machine learning as technique for my dataset. It is important to remark the differences between supervised, unsupervised, and semi-supervised learning as I list below:

Supervised Learning:

- Trained on labeled data.

- Learns a mapping from input to output.

- Used for classification and regression tasks.

Unsupervised Learning:

- Works with unlabeled data.

- Identifies patterns or clusters in the data.

- Used for clustering and dimensionality reduction.

Semi-Supervised Learning:

- Uses both labeled and unlabeled data.

- Leverages labeled data to guide learning on unlabeled data.

- Balances supervised and unsupervised learning principles.

According to the differences, the dataset match well with labeled data like “**Type of Employee**”, when there are different categories of employees. The purpose of using supervised machine learning is that is useful for regression in the current dataset.

**2)**

In this second part, I chose XGBoost Regression and Random Forest Regression, because they are both powerful machine learning models with unique advantages when it comes to estimating weekly salaries from my dataset of industry employees. These models are particularly useful in regression tasks where the goal is to predict continuous numeric values, such as salary amounts.

XGBoost Regression stands out for its high predictive accuracy. It is renowned for its ability to provide precise predictions and often outperforms other regression algorithms on various datasets. This is primarily attributed to its ensemble of decision trees and the optimization of bias and variance, resulting in accurate salary estimates. Additionally, XGBoost incorporates L1 (Lasso) and L2 (Ridge) regularization terms in its objective function, helping to prevent overfitting and improving the model's robustness in the face of noisy data. The use of gradient boosting is another key advantage, as it enables the model to build trees sequentially, learning from the mistakes of previous trees, which enhances overall performance.

I used GridSearchCV to look for the best hyperparameters. It systematically tests all possible hyperparameter combinations within the defined grid. It provides the benefit of a comprehensive and thorough search, which can result in finding the best combination of hyperparameters that optimizes the model's performance, accuracy and generalization.

Random Forest Regression, on the other hand, offers a different set of advantages for this regression task. As an ensemble model, Random Forest combines multiple decision trees to make predictions, which significantly reduces the risk of overfitting and enhances model stability. This is especially valuable when working with salary estimation data, as overfitting can lead to inaccurate predictions. Random Forest provides feature importance scores, allowing you to identify the most influential variables affecting weekly salaries. This feature importance analysis aids in feature selection and model interpretation, which can be critical when explaining salary predictions to stakeholders. Random Forest is relatively robust to outliers and noisy data, making it a solid choice for datasets with varying data quality, which is common in salary estimation tasks. Additionally, it's easy to use and requires fewer hyperparameters to fine-tune compared to XGBoost, making it a practical option for quick model development and deployment. It can also be trained in parallel, saving time and resources when dealing with large datasets. Another convenience is that Random Forest models do not require extensive feature scaling, simplifying the data preprocessing phase.

**Statistics part:**

**Programming part:**

Procedural programming:

When I created the function draw\_histogram(), I thought how it can help me to reuse code. I think the paradigm's emphasis on modularity, achieved through the division of programs into procedures or functions, facilitates effective code management and maintenance, allowing changes to be localized without impacting the entire program. This modular structure also promotes reusability, as procedures can be utilized across different parts of a program or in other programs, contributing to time and effort savings. Additionally, the paradigm's efficiency, attributed to the straightforward, linear nature of the code, is evident in terms of execution speed and memory usage in specific situations. Debugging procedural code is often more straightforward due to the clear flow of execution, aiding in issue identification and resolution. The paradigm's widespread adoption is underscored by the extensive use of procedural languages, contributing to a well-established ecosystem of resources, libraries, and tools that support its application in diverse programming contexts.

In other context I wanted to try Object-Oriented Programming (OOP), and make my own methods when I used different parameters. I think it is important, because it offers a modular and reusable approach to software development, promoting code organization through encapsulation of related functionalities into classes and objects. This modularity enhances readability and maintenance, while the reusability of objects across different parts of a program fosters efficiency. OOP's flexibility and extensibility allow for the addition of new classes without modifying existing code, supporting scalability in evolving projects. With abstraction, OOP simplifies development by focusing on relevant details, and encapsulation ensures data integrity and security. Inheritance facilitates code reuse, and polymorphism enables flexible coding. Easier maintenance is achieved by isolating changes to specific parts of the code, and OOP's real-world modelling enhances conceptualization. Lastly, the collaborative nature of OOP, with its support for classes and objects, facilitates teamwork, making it well-suited for diverse and complex software development projects.