National College of Ireland

Configuration Manual

MSc Research Project
MSc Data Analysis

Nishika Gala
Student ID: XXX

School of Computing
National College of Ireland

Supervisor: XXX

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Nishika Gala……

Student ID: X22214062……

Programme: MSc Data Analysis…… **Year:** 2023-24

Module: ……

Lecturer: 12th August 2024……

Submission Due Date: ……

Project Title: Utilizing machine learning for predictive analysis and optimizing restautant operations……

Word Count: ……895…… **Page Count: 4**……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Nishika Gala……

Date: 12th August 2024……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Forename Surname
Student ID:

1. System Configuration

1.1 Operating System
- Recommended: Ubuntu 20.04 LTS or Windows 10/11
- Supported: macOS 10.15 or later, Windows 7/8

1.2 Hardware Requirements
- Processor: Intel Core i5 or AMD Ryzen 5 equivalent or higher
- Memory: Minimum 8 GB RAM (16 GB or higher recommended)
- Storage: At least 100 GB free space on SSD

1.3 Software Configuration
- Python: Version 3.8 or higher
- Jupyter Notebook: Installed via Anaconda or pip
- IDE: (Optional) Visual Studio Code, PyCharm

2. System Requirements

2.1 Required Software
- Python: Python 3.8 or higher (Anaconda distribution recommended)
- Jupyter Notebook: Available via the Anaconda distribution or install separately using pip
- Git: Version control system for managing your project

2.2 Python Environment Setup
To set up the Python environment, follow these steps:
1. Install Anaconda: Download and install the [Anaconda distribution](https://www.anaconda.com/products/individual).

2. Create a Virtual Environment:

```
conda activate restaurant_analysis
```

3. Install Required Libraries: Once inside the environment, install the required libraries.

3. Python Libraries

Ensure the following Python libraries are installed in your environment:

```
pip install pandas numpy matplotlib seaborn scikit-learn tensorflow keras xgboost jupyterlab
```

- pandas: For data manipulation and analysis
- numpy: For numerical computations
- matplotlib & seaborn: For data visualization

- scikit-learn: For machine learning algorithms
- tensorflow & keras: For deep learning models
- xgboost: For gradient boosting algorithms
- jupyterlab: For working with Jupyter Notebooks

4. Dataset

4.1 Dataset Description
- The dataset used for this thesis focuses on restaurant operations, including customer demographics, sales, inventory, and employee performance data.
- Example datasets:
  - Sales Data: Date, Time, Order ID, Items Sold, Sales Amount, Customer ID
  - Customer Data: Customer ID, Age, Gender, Visit Frequency, Spending Patterns
  - Inventory Data: Item ID, Item Name, Stock Level, Supplier, Cost
  - Employee Data: Employee ID, Name, Position, Shift Hours, Performance Score

4.2 Dataset Sources

Publicly available datasets from sources like Kaggle

4.3 Loading the Dataset
Load the dataset using pandas:

```python
import pandas as pd

sales_data = pd.read_csv('sales_data.csv')
customer_data = pd.read_csv('customer_data.csv')
inventory_data = pd.read_csv('inventory_data.csv')
employee_data = pd.read_csv('employee_data.csv')
```

5. Data Preprocessing

5.1 Data Cleaning
All the data is combined into one common dataset to begin the project.
Data is cleaned and not required columns were dropped from the data frame

5.2 Feature Engineering
-Create New Features(e.g., Total Sales Per Customer):
```python
sales_data['Total_Sales_Per_Customer'] = sales_data.groupby('Customer_ID')['Sales_Amount'].transform('sum')
```

5.3 Data Normalization/Scaling
- Standard Scaling:
```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaled_data = scaler.fit_transform(sales_data[['Sales_Amount', 'Total_Sales_Per_Customer']])
```

6. Data Analysis

6.1 Exploratory Data Analysis (EDA)
- Summary Statistics:
```python
sales_data.describe()
```

- Visualizations:
  - Sales Trends:
```python
import matplotlib.pyplot as plt

sales_data.groupby('Date')['Sales_Amount'].sum().plot()
plt.title('Daily Sales Trend')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.show()
```

  - Customer Segmentation:
```python
import seaborn as sns

sns.scatterplot(x='Age', y='Total_Sales_Per_Customer', data=customer_data)
plt.title('Customer Segmentation by Age and Spending')
plt.show()
```

6.2 Correlation Analysis
- Identify Relationships:
```python
correlation_matrix = sales_data.corr()
sns.heatmap(correlation_matrix, annot=True)
plt.title('Correlation Matrix')
plt.show()
```

7. Model Training and Testing

7.1 Splitting the Data
- Train-Test Split:
```python
from sklearn.model_selection import train_test_split

X = sales_data[['Age', 'Visit_Frequency', 'Total_Sales_Per_Customer']]
y = sales_data['Sales_Amount']

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

7.2 Model Selection
- Choosing the Right Model: Experiment with models like Linear Regression, Random Forest, and XGBoost.
```python
from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

7.3 Model Evaluation
- Evaluate Model Performance:

```python
from sklearn.metrics import mean_squared_error, r2_score

predictions = model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)

print(f"Mean Squared Error: {mse}")
print(f"R2 Score: {r2}")
```

7.4 Model Optimization
- Hyperparameter Tuning:
```python
from sklearn.model_selection import GridSearchCV

param_grid = {'n_estimators': [100, 200, 300], 'max_depth': [10, 20, 30]}
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)

best_model = grid_search.best_estimator_
```

7.5 Model Deployment
Saving the Model:
```python
import joblib

joblib.dump(best_model, 'restaurant_sales_predictor.pkl')
```
- **Loading the Model**:
```python
loaded_model = joblib.load('restaurant_sales_predictor.pkl')
```