National College of Ireland

FORECASTING UKRAINIAN REFUGEE EMPLOYMENT IN IRELAND'S ACCOMMODATION & FOOD SERVICE SECTOR USING RANDOM FOREST, GRADIENT BOOSTING, AND NEURAL NETWORK MODELS

MSc Research Project
Data Analytics

Dikshant Bhosale
Student ID: x22193146

School of Computing
National College of Ireland

Supervisor: Abdul Qayum

National College of Ireland
Project Submission Sheet School of Computing

Student Name:	Dikshant Bhosale
Student ID:	X22193146
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Abdul Qayum
Submission Due Date:	02/12/2024
Project Title:	Forecasting Ukrainian refugee employment in Ireland's Accommodation & Food Service sector Using Random Forest, Gradient Boosting, and Neural Network models
Word Count:	XXX
Page Count:	8

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:	Dikshant Manohar Bhosale
Date:	2nd December 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):	

FORECASTING UKRAINIAN REFUGEE EMPLOYMENT IN IRELAND'S ACCOMMODATION & FOOD SERVICE SECTOR USING RANDOM FOREST, GRADIENT BOOSTING, AND NEURAL NETWORK MODELS

Dikshant Bhosale
X22193146

1 Environment Requirements

This configuration manual discusses the hardware, and the software required to implement the research work. The steps taken are mentioned so it will be easy for anyone replicating the experiments.

2 System Specification

2.1 Hardware Requirements

The system specification where all experiments implemented are discussed below:

- **Processor:** Intel Core i5.
- **System Memory:** 1TB Hard disk, 256GB SSD.
- **RAM:** 16GB.

2.2 Software Requirements

The software requirements are discussed below:

- **Windows Edition:** Windows 10+
- **Integrated Development Environment:** Jupyter Notebook version 6.1+

  Jupyter Notebook reference figure 1

Loading Dataset

```
In [2]: emp_df = pd.read_csv('Employment_Number.csv')
```

Data Preprocessing

```
In [3]: emp_df.shape
Out[3]: (384, 5)

In [4]: emp_df.dtypes
Out[4]: Statistic Label    object
        Day                object
        NACE Sector        object
        UNIT               object
        VALUE               int64
        dtype: object

In [5]: emp_df.head()
```

Out[5]:

	Statistic Label	Day	NACE Sector	UNIT	VALUE
0	Number of Employments	2022 June 20	Financial and insurance activities (K)	Number	23
1	Number of Employments	2022 June 20	Transportation and Storage (H)	Number	42
2	Number of Employments	2022 June 20	Information and communication (J)	Number	62
3	Number of Employments	2022 June 20	Public Administration and Defence (O)	Number	66

Figure 1: Jupyter Notebook

- **Scripting Language:** Python 3.

2.3 Jupyter Notebook setup

If any user does not have Jupyter Notebook setup on their machine then,

- Install Anaconda.
- Python 3 version or python 3+ version is mandatory.

Alternatively, to install Jupyter notebook using python's package manager,

- "pip3 install --upgrade pip" run this command to ensure latest pip.
- Then install Jupyter notebook by using this command "pip3 install jupyter".

2.4 Libraries

required libraries are mentioned in
Figure 2
Documentation of libraries:-

- NumPy:- It is numerical library for machine learning and statistical analysis. https://numpy.org/doc/
- SciPy:- It is used for scientific and technical computing. https://docs.scipy.org/doc/scipy/

- Scikit-Learn (sklearn) :- Modelling library which offers tools for regression machine learning models. https://scikitlearn.org/stable/

Importing Libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import learning_curve
from sklearn.model_selection import GridSearchCV
```

Figure 2: Required libraries

3 Dataset Implementation

3.1 Simple model implementation

- Download the necessary dataset from open source such as Central Statistics Office (CSO).
- Put those dataset in project directory..

Loading Dataset

```
emp_df = pd.read_csv('Employment_Number.csv')
```

Figure 3.1: Dataset Implementation

- Link of "Employment_Number" dataset is https://data.cso.ie/table/UA29

4 Data Cleaning

- In data cleaning check duplicates and null values.
- Handled it based on requirements.

```
emp_df.isna().mean()
```

```
emp_df.duplicated()
```

```
# Remove 'All NACE Economic Sectors'
emp_df = emp_df[emp_df['NACE Sector'] != 'All NACE Economic Sectors']
```

Figure 4 : Data Cleaning

- Now data is cleaned but do check null values and duplicates again for confirmation.

5 Data Visualization

Figure 5.1 describes Employment Trends Over Time By Sector For Ukrainian Refugees.

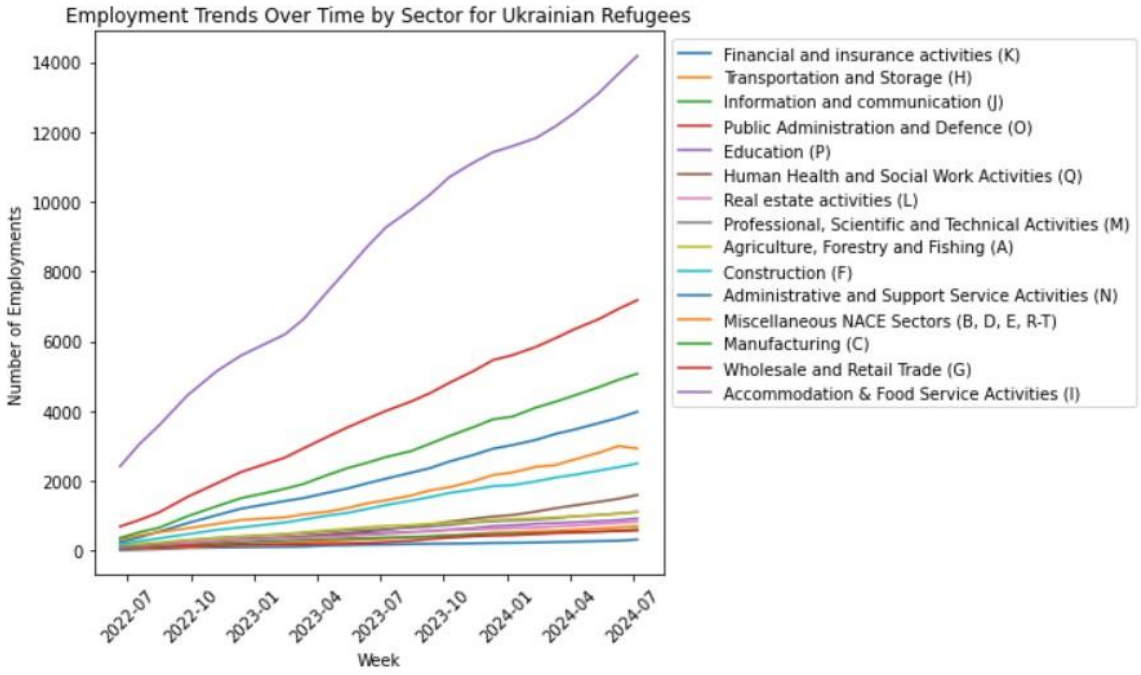

Figure 5.1: Employment Trends Over Time By Sector For Ukrainian Refugees.

6 Splitting Data

```
# Split and scale data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Figure 6: Splitting data

- Splitting data into training and testing using standard ration which is 80 : 20.

7 Machine Learning And Evaluation

- Applying models like Random Forest, Gradient Boosting and Neural network.

```
# Initialize model
random_forest = RandomForestRegressor(random_state=42)
```

```
# Train model
random_forest.fit(X_train_scaled, y_train)
```

```
RandomForestRegressor(random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# Make predictions on training and testing sets
y_train_pred_rf = random_forest.predict(X_train_scaled)
y_test_pred_rf = random_forest.predict(X_test_scaled)
```

```
# Calculate metrics for training set
rf_train_results = {
    'R2': r2_score(y_train, y_train_pred_rf),
    'RMSE': np.sqrt(mean_squared_error(y_train, y_train_pred_rf)),
    'MAE': mean_absolute_error(y_train, y_train_pred_rf)
}
```

Figure 7.1: Random Forest

```
# Initialize model
gradient_boosting = GradientBoostingRegressor(random_state=42)
```

```
# Train the model
gradient_boosting.fit(X_train_scaled, y_train)
```

```
GradientBoostingRegressor(random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# Training results
y_train_pred_gb = gradient_boosting.predict(X_train_scaled)
gb_train_results = {
    'R2': r2_score(y_train, y_train_pred_gb),
    'RMSE': np.sqrt(mean_squared_error(y_train, y_train_pred_gb)),
    'MAE': mean_absolute_error(y_train, y_train_pred_gb)
}
```

Figure 7.2: Gradient Boosting

```
# Initialize model
neural_network = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42)
```

```
# Train the model
neural_network.fit(X_train_scaled, y_train)
```

```
C:\Users\Dikshant Bhosale\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:
Stochastic Optimizer: Maximum iterations (1000) reached and the optimization hasn't converged yet.
  warnings.warn(
```

```
MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# Training results
y_train_pred_nn = neural_network.predict(X_train_scaled)
nn_train_results = {
    'R2': r2_score(y_train, y_train_pred_nn),
    'RMSE': np.sqrt(mean_squared_error(y_train, y_train_pred_nn)),
    'MAE': mean_absolute_error(y_train, y_train_pred_nn)
}
```

Figure 7.3: Neural Network

8 Learning Curve

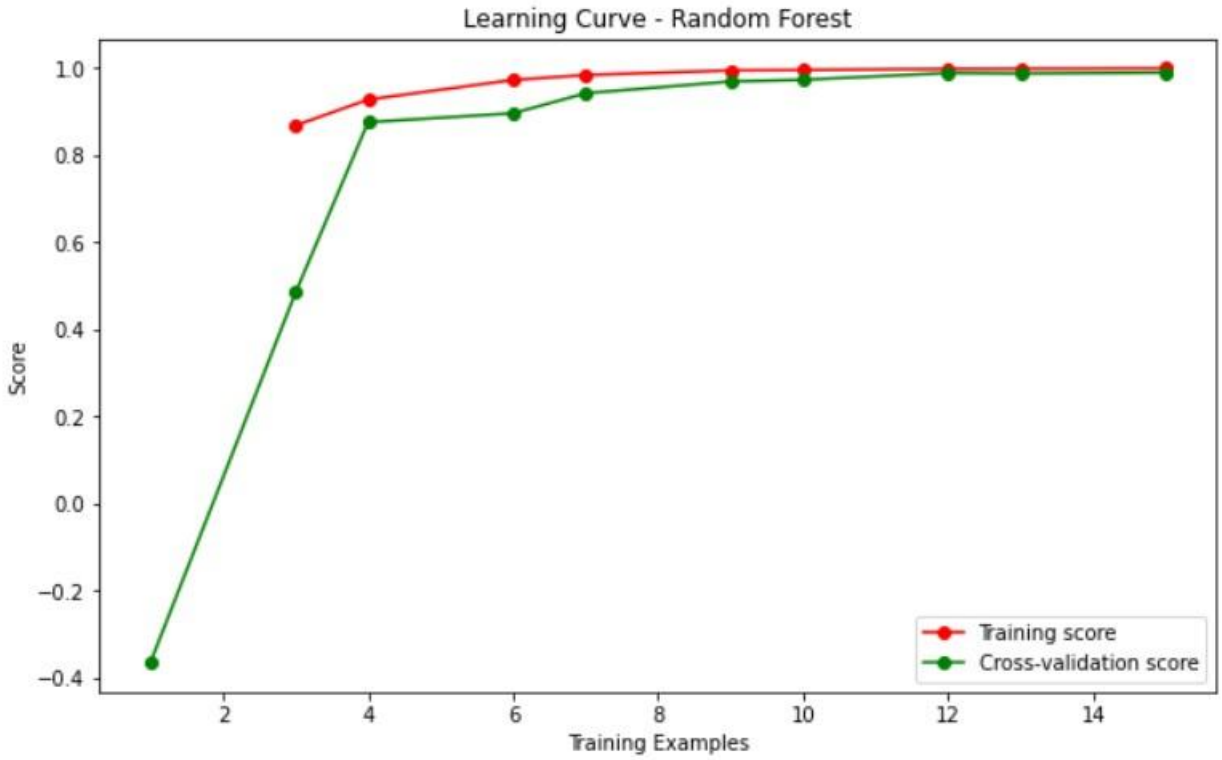

Figure 8.1: Random Forest's Learning Curve.

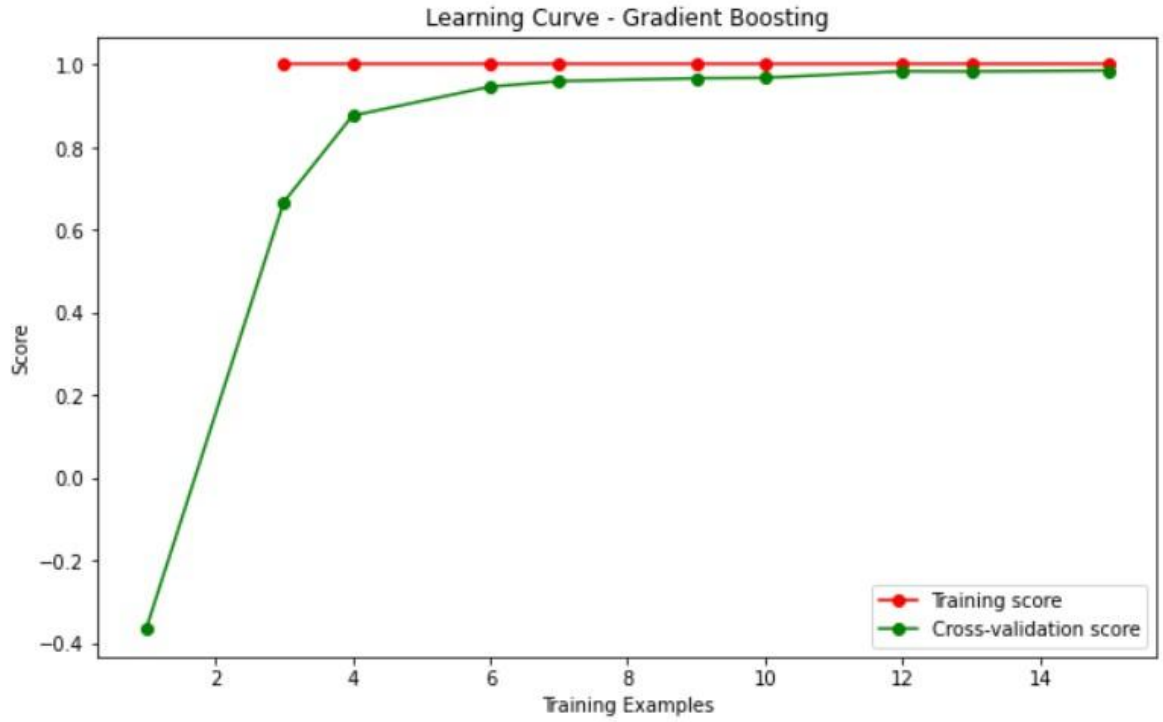

Figure 8.2: Gradient Boosting's Learning Curve

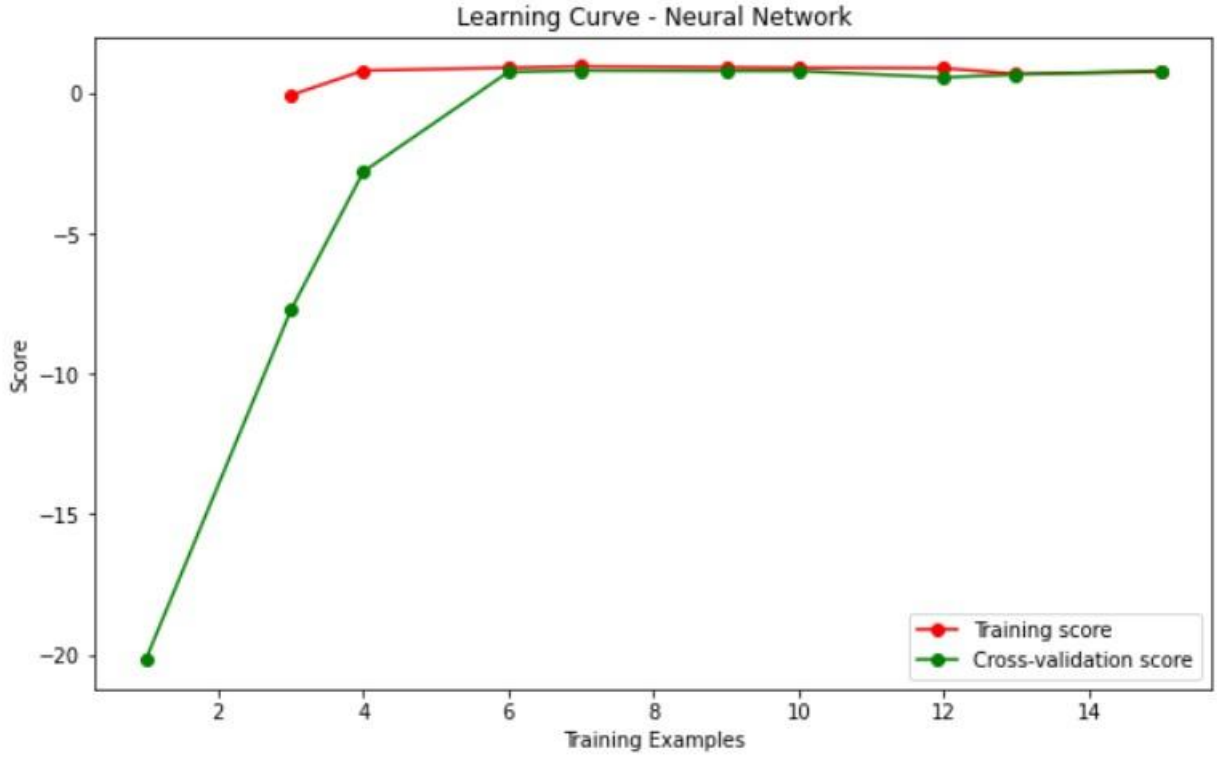

Figure 8.3: Neural Network's Learning Curve