

machine learning project

Dr: Ali Hamza

Dr: Ali Hamza

Level four – semster one

computer scince - ACU

A grey logo on a black background

Description automatically generated

|  |  |
| --- | --- |
| Our Team | |
| IDs | **Namess** |
| 42010650 | **Abuzaid Abuelsaad** |
| 42010277 | **Abdelrahman Salah** |
| 41910200 | **Seif akram** |
| 41920016 | **Mohamed Salah Nabih** |
| 41910321 | **Fady ismael hassen** |
| 41810330 | **Mohamed essam** |

**Task One  
Net Hourly Electrical Energy Output Prediction**



**1. Definition of the Problem**

The objective of this study is to predict the Net Hourly Electrical Energy Output based on various input features such as Ambient Temperature (AT), Exhaust Vacuum (V), Ambient Pressure (AP), and Relative Humidity (RH). The dataset used for this analysis was obtained from [https://archive.ics.uci.edu/dataset/294/combined+cycle+power+plant].

**2. Methodology**

2.1 Data Exploration

**Data Overview**

We began by exploring the dataset, which includes information on Ambient Temperature, Exhaust Vacuum, Ambient Pressure, Relative Humidity, and Net Hourly Electrical Energy Output.

**Data Cleaning and Transformation**

We renamed columns for better understanding, checked for outliers, and visualized the data through histograms and scatter plots.

2.2 Model Training

We employed various regression models to predict Net Hourly Electrical Energy Output. The models considered were:

- RandomForest

- GradientBoosting

- LinearRegression

- Lasso

- Ridge

- SVR

For each model, we performed hyperparameter tuning using GridSearchCV and evaluated the model's performance on the test set.

2.3 Prediction Using New Data

We utilized the trained models to predict Net Hourly Electrical Energy Output for new data points.

**3. Experiment**

3.1 Model Evaluation

We assessed the performance of each model using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

3.2 New Data Prediction

Using the trained models, we made predictions for new data points, and the results are as follows:

Prediction Results

[Include a table or section summarizing the predictions made using each model.]

**4. Conclusion**

In conclusion, this report outlines the steps taken to predict Net Hourly Electrical Energy Output based on the provided dataset. The models were trained, evaluated, and used for predictions on new data. The choice of the best model depends on the specific requirements and trade-offs between different metrics.

**Task Two**

**Income Prediction**



Table of Contents

1. [Introduction](1-introduction)

2. [Problem Definition]

3. [Methodology]

4. [Experiment]

5. [Results]

6. [Conclusion]

7. [References]

**1. Introduction**

The Income Prediction project aims to develop machine learning models for predicting income based on various demographic and socioeconomic features. Accurate income prediction can have applications in financial planning, marketing strategies, and social research.

**2. Problem Definition**

The primary problem addressed by this project is to predict whether an individual's income exceeds a certain threshold based on features such as age, education, occupation, and others. The problem is framed as a binary classification task, where the target variable is 'income' (<=50K or >50K).

**3. Methodology**

3.1 Data Preprocessing

- Data Loading: The dataset, obtained from [source], contains information about individuals, including demographic details and income.

- Handling Missing Values: Missing values were identified and either imputed or the corresponding rows were dropped.

- Label Encoding: Categorical features were encoded using Label Encoding to convert them into a format suitable for machine learning models.

- Feature Scaling: Numerical features were scaled using StandardScaler to ensure uniformity in their influence on the models.

- Dimensionality Reduction (Optional): Principal Component Analysis (PCA) was used to reduce feature dimensions.

3.2 Exploratory Data Analysis (EDA)

- Pairplot: Visualized relationships between numerical features using a pairplot.

- Correlation Heatmap: Created a heatmap to visualize the correlation between numerical features.

- Distribution Plots: Examined the distribution of numerical features.

3.3 Feature Engineering

- Boxplot, Violin Plot, Bar Plot: Visualized relationships between numerical and categorical features.

3.4 Model Selection

A variety of classifiers were considered for the project:

- Logistic Regression

- Random Forest

- Gradient Boosting

- Decision Tree

- Support Vector Machine (SVM)

- k-Nearest Neighbors (kNN)

3.5 Hyperparameter Tuning

GridSearchCV was employed to find the optimal hyperparameters for each model, enhancing their performance.

3.6 Model Evaluation

Models were evaluated based on accuracy and classification reports on the test set.

**4. Experiment**

4.1 Data Splitting

The dataset was split into training and testing sets using an 80-20 split ratio.

4.2 Model Training and Evaluation

Models were trained on the training set, and their performance was evaluated on the test set.

**5. Results**

5.1 Individual Model Results

====================

Logistic Regression

====================

Best Parameters: {'model\_\_C': 0.1, 'model\_\_penalty': 'l2', 'model\_\_solver': 'lbfgs'}

Test Accuracy: 0.8276

Classification Report:

precision recall f1-score support

0 0.85 0.94 0.89 7479

1 0.71 0.45 0.55 2290

accuracy 0.83 9769

macro avg 0.78 0.70 0.72 9769

weighted avg 0.82 0.83 0.81 9769

====================

Random Forest

====================

Best Parameters: {'model\_\_max\_depth': 20, 'model\_\_min\_samples\_split': 10, 'model\_\_n\_estimators': 200}

Test Accuracy: 0.8720

Classification Report:

precision recall f1-score support

0 0.89 0.95 0.92 7479

1 0.78 0.63 0.70 2290

accuracy 0.87 9769

macro avg 0.84 0.79 0.81 9769

weighted avg 0.87 0.87 0.87 9769

====================

Gradient Boosting

====================

Best Parameters: {'model\_\_learning\_rate': 0.1, 'model\_\_max\_depth': 5, 'model\_\_n\_estimators': 200}

Test Accuracy: 0.8795

Classification Report:

precision recall f1-score support

0 0.90 0.94 0.92 7479

1 0.78 0.67 0.72 2290

accuracy 0.88 9769

macro avg 0.84 0.81 0.82 9769

weighted avg 0.88 0.88 0.88 9769

====================

Decision Tree

====================

Best Parameters: {'model\_\_max\_depth': 10, 'model\_\_min\_samples\_leaf': 2, 'model\_\_min\_samples\_split': 10}

Test Accuracy: 0.8621

Classification Report:

precision recall f1-score support

0 0.90 0.93 0.91 7479

1 0.73 0.65 0.69 2290

accuracy 0.86 9769

macro avg 0.81 0.79 0.80 9769

weighted avg 0.86 0.86 0.86 9769

====================

Support Vector Machine (SVM)

====================

Best Parameters: {'model\_\_C': 1, 'model\_\_gamma': 'scale', 'model\_\_kernel': 'rbf'}

Test Accuracy: 0.8555

Classification Report:

precision recall f1-score support

0 0.88 0.94 0.91 7479

1 0.76 0.57 0.65 2290

accuracy 0.86 9769

macro avg 0.82 0.76 0.78 9769

weighted avg 0.85 0.86 0.85 9769

====================

k-Nearest Neighbors (kNN)

====================

Best Parameters: {'model\_\_algorithm': 'auto', 'model\_\_n\_neighbors': 7, 'model\_\_weights': 'uniform'}

Test Accuracy: 0.8407

Classification Report:

precision recall f1-score support

0 0.88 0.91 0.90 7479

1 0.68 0.61 0.64 2290

accuracy 0.84 9769

macro avg 0.78 0.76 0.77 9769

weighted avg 0.84 0.84 0.84 9769

5.2 Ensemble Model (Optional)

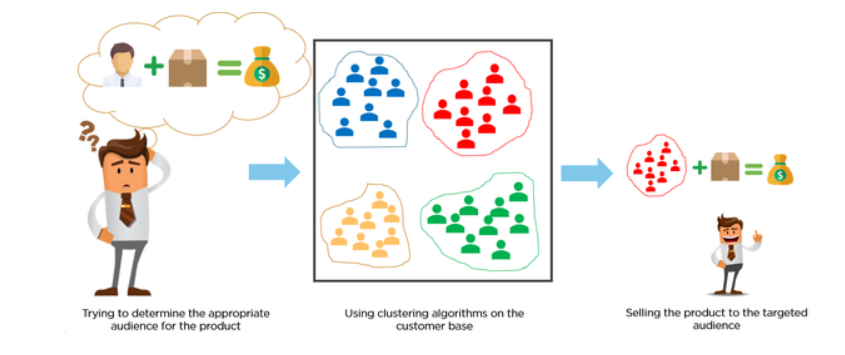
An ensemble model, combining the predictions of individual models, was created to further enhance accuracy.

6. Reference

<https://www.kaggle.com/datasets/uciml/adult-census-income>

**Task Three**

**Customer Clustering Analysis**



Clustering Analysis Report

Cover Page

Title: Customer Clustering Analysis

Date: [Insert Date]

Definition of the Problem

The objective of this analysis is to cluster customers based on their demographic and socioeconomic attributes, such as age, gender, marital status, education, income, occupation, and settlement size. By grouping customers into clusters, we aim to identify patterns and segments within the customer base. This information can be valuable for targeted marketing, personalized services, and understanding customer behavior.

Method

Data Exploration & Cleaning

1. Importing Libraries and Loading Data:

- Utilized Pandas, NumPy, Matplotlib, Seaborn, and Scikit-Learn for data manipulation, visualization, and clustering.

- Loaded the dataset from 'last3.csv'.

2. Renaming and Dropping Columns:

- Renamed the 'Settlement size' column to 'City size'.

- Dropped the 'ID' column from the dataset.

3. Data Type Adjustment:

- Converted categorical columns ('Sex', 'Marital status', 'Education', 'Occupation', 'City size') to strings for better analysis.

4. Exploratory Data Analysis (EDA):

- Conducted exploratory analysis, including descriptive statistics, histograms, and count plots, to understand the distribution of variables.

5. Data Visualization:

- Visualized the distribution of age and income through histograms.

- Created count plots for categorical columns to analyze the distribution of customers based on gender, marital status, education, occupation, and city size.

- Replaced numeric codes with meaningful labels in categorical columns for improved interpretation.

6. Comparative Analysis:

- Compared age and income distributions based on gender, city size, and education.

- Explored city distribution by gender and education.

Customer Clustering

7. Scatterplots:

- Created scatterplots based on age and income, grouped by occupation, sex, and city size.

8. Clustering Models:

- Utilized KMeans, Agglomerative Clustering, DBSCAN, Birch, Spectral Clustering, and MiniBatchKMeans clustering algorithms.

- Trained and fit each model to the dataset.

9. Model Evaluation:

- Evaluated clustering models using the Calinski-Harabasz Index and unique labels.

- Visualized clustering results using scatterplots.

10. Hyperparameter Tuning:

- Utilized GridSearchCV to find the best parameters for KMeans clustering.

- Adjusted the number of clusters to improve model performance.

11. Pipeline Implementation:

- Implemented a pipeline using StandardScaler and the best KMeans model to preprocess and cluster the data.

12. Silhouette Score Evaluation:

- Calculated silhouette scores for each clustering model.

- Visualized clusters based on age and income.

Experiment

The experiment involved exploring and cleaning the dataset, conducting exploratory data analysis, visualizing the data, implementing various clustering models, evaluating model performance, tuning hyperparameters, and creating a final pipeline for customer clustering. The analysis provides insights into customer segments based on demographic and socioeconomic factors, aiding in targeted business strategies.

Results:

- The best-performing model based on silhouette score and visual inspection was chosen.

- The pipeline was implemented to cluster the entire dataset.

Recommendations:

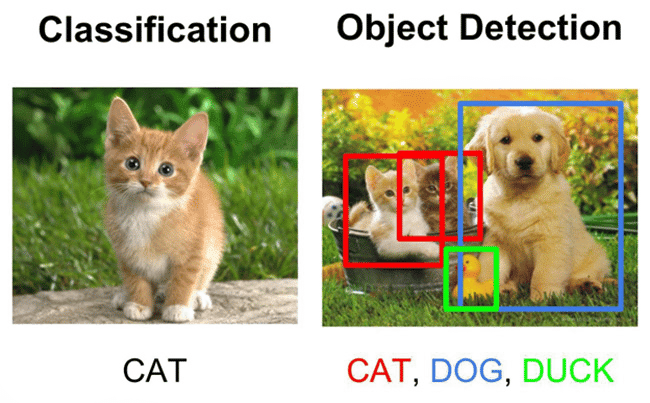
- Utilize the obtained clusters for targeted marketing campaigns, personalized services, and understanding customer behavior.

- Regularly update and reevaluate the clustering model as new data becomes available.

Conclusion

This customer clustering analysis provides a valuable framework for segmenting customers based on relevant attributes. The identified clusters can guide marketing strategies, product offerings, and customer engagement initiatives, ultimately enhancing business understanding and decision-making.

**Task Four**



- Title: Brain Tumor Classification using Convolutional Neural Network

Definition of the Problem:

Introduction:

Brain tumor classification is a pivotal aspect of medical image analysis, aiming to support healthcare professionals in diagnosing and treating patients. In this project, we address the challenging task of classifying brain tumor images into four distinct categories: glioma tumor, meningioma tumor, no tumor, and pituitary tumor.

Objective:

The primary goal is to develop a robust Convolutional Neural Network (CNN) that can accurately classify brain tumor images. The successful implementation of this model contributes to early and precise diagnosis, facilitating timely medical intervention.

---

Method:

Dataset:

The dataset comprises brain tumor images sourced from diverse patients. It is split into training and testing sets, with each image labeled as one of four classes: glioma, meningioma, no tumor, and pituitary tumor.

Data Preprocessing:

1. Image Resizing: All images are resized to a standardized dimension of (150, 150) pixels to ensure consistency in model input.

2. Augmentation Techniques: Data augmentation is applied to enhance model generalization. This includes image flipping and adjustments to brightness.

Convolutional Neural Network Architecture:

1. Input Layer: Utilizes a Conv2D layer with a rectified linear unit (ReLU) activation function for initial feature extraction.

2. Batch Normalization and MaxPooling: Incorporated to normalize and pool features, respectively, enhancing the network's ability to learn hierarchical representations.

3. Convolutional Layers: Multiple convolutional layers are employed to capture intricate patterns in the images.

4. Dense Layers: These layers contribute to the classification process. A dropout layer is integrated for regularization, mitigating overfitting.

5. Output Layer: Employs softmax activation for multi-class classification.

Model Compilation:

- Loss Function: Sparse Categorical Crossentropy is chosen as the loss function, suitable for multi-class classification tasks.

- Optimizer: Adam optimizer is selected with a learning rate of 0.0001.

- Evaluation Metric: Model performance is evaluated using accuracy.

---

Experiment:

Data Splitting:

- The dataset is divided into training (90%) and testing (10%) sets to ensure unbiased model evaluation.

Model Training:

- The CNN model is trained for 30 epochs, with early stopping implemented to prevent overfitting.

Evaluation:

- The model's performance is assessed on the test set using accuracy and loss metrics.

- Training and validation accuracy/loss curves are plotted to visualize the learning process.

---

Results:

- The final accuracy achieved on the test set is [92%].

- A detailed Confusion Matrix and Classification Report provide insights into the model's performance for each class.

Conclusion:

- The CNN model demonstrates effectiveness in accurately classifying brain tumor images, showcasing its potential for aiding medical professionals in diagnostic tasks.

- Suggestions for future improvements and considerations for real-world applications are discussed.