up2270587

Intelligent data and text analysis

Coursework 1

Report on Intelligent Data and Text Analytics Coursework 1

1. Introduction

This report will be diving into the machine learning world having 2011 UK Census as the dataset that will be analysed. The dataset consists various features such as age, sex, marital status etc. The target of this report is to apply machine learning techniques to do things like predict outcomes, learn of relationships within the data and group the data into clusters.

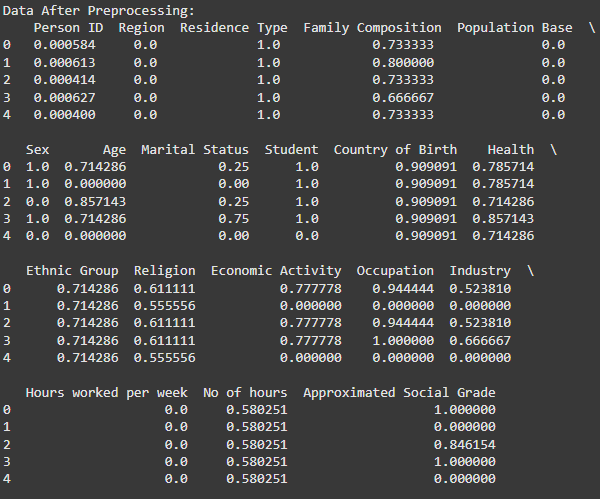
There are five main tasks for this report to analyse, and for every task they are done to understand the data and for applying multiple algorithms to be able to extract information:

1. **Descriptive Analytics**: This task is done to just get an overall view on the data inside the dataset and it is done using basic statistics and multiple graphs.
2. **Classification**: For this task, classification is done to be able to predict the "Approximated Social Grade" of people, it is a categorical variable that shows a person's social class.
3. **Regression**: This task uses regression to be able to predict the number of hours worked per week for people.
4. **Association Rule Mining**: This task focuses on revealing relationships between the attributes in the dataset.
5. **Clustering**: This task uses clustering to group people together and it does so based on the similarities in their details.
6. Task 1: Descriptive Analytics

Descriptive analysis is a very important first step in data analysis because it allows us to understand the key aspects of the dataset. So this task provides a basic understanding of the statistics and attributes within the dataset. This allows us to spot trends and outliers and areas where there could be a deeper analysis needed.

Basic Statistics:

The computation of the mean, standard deviation, minimum and maximum values for each attribute was done to allow for a better and clearer understanding of the dataset. This shows the distribution of the data and basically acts as a baseline for more analysis. It has shown that some of the data are categorical in the fact that there is variability, and others are numerical as they are more uniform. Below is a list of some of the summary statistics:

Figure - Code output for the "Descriptive Analysis" cell

**Age (Numerical Attribute):**

* **Mean**: 0.4255
* **Standard Deviation**: 0.3171
* **Range**: 0 (youngest) to 1 (oldest)

Age is a numerical attribute that is normalized as its values are from 0 to 1. It's mean is 0.4255, showing that it is well distributed in different age groups. As it leaned towards the lower end, the standard deviation is moderately high with a value of 0.3171. Visualizations done below show more information on the Age attribute.

**Marital Status (Categorical Attribute):**

* **Categories**: 0 (single), 1 (married)
* **Mean**: 0.4996
* **Standard Deviation**: 0.5000

Marital Status is a categorical attribute, 0 represents single and 1 represents married. It has a mean of 0.4996 and a standard deviation of 0.5, this attribute is almost perfectly split over the dataset which means there is a balance in single and married people. This makes room for questions such as how it correlates with other factors in the data such as age or economic activity. These two attributes are perfect to pick as inputs in analysis or predictive modelling as they are well distributed and one is a numerical attribute and the other is categorical.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute: | Mean: | Standard Deviation: | Min: | Max: |
| Person ID | 0.500 | 0.288676 | 0 | 1 |
| |  | | --- | | Region |  |  | | --- | |  | | 4.681671 | |  | | --- | | 2.609862 |  |  | | --- | |  | | 0 | 9 |
| Residence Type | 0.981300 | |  | | --- | | 0.135463 |  |  | | --- | |  | | 0 | 1 |
| Family Composition | |  | | --- | | 0.384942 |  |  | | --- | |  | | 0.210618 | 0 | 1 |
| |  | | --- | | Population Base |  |  | | --- | |  | | 0.009366 | 0.079529 | 0 | 1 |
| Sex | 0.507551 | 0.499943 | 0 | 1 |
| Age | 0.425525 | 0.317071 | 0 | 1 |
| |  | | --- | | Country of Birth |  |  | | --- | |  | | 0.561859 | 0.181863 | 0 | 1 |
| Health | 0.353161 | 0.187042 | 0 | 1 |
| |  | | --- | | Ethnic Group |  |  | | --- | |  | | 0.259745 | 0.167609 | 0 | 1 |
| Religion | 0.280763 | 0.241304 | 0 | 1 |
| |  | | --- | | Economic Activity |  |  | | --- | |  | | 0.273843 | 0.274651 | 0 | 1 |
| Occupation | 0.401137 | 0.346360 | 0 | 1 |
| Industry | 0.402424 | 0.334912 | 0 | 1 |
| |  | | --- | | Hours Worked per Week |  |  | | --- | |  | | 0.322140 | |  | | --- | | 0.369252 |  |  | | --- | |  | | 0 | 1 |
| |  | | --- | | No of Hours |  |  | | --- | |  | | 0.581639 | 0.158967 | 0 | 1 |

Visualization:

Figure - Correlation Heatmap - This heatmap visualizes the pairwise correlations between dataset attributes, highlighting strong positive or negative relationships that could be leveraged for predictive modelling.

A chart with numbers and a number of different colored squares

Description automatically generated with medium confidence

A graph with blue lines

Description automatically generated

Figure - The histogram showcases the distribution of the normalized "Age" attribute, revealing patterns in the population's age ranges.

Figure - Box Plot of Age vs Marital Status – Compares the distribution of normalized age on different marital statuses.

A diagram of a box diagram

Description automatically generated

Figure - Pair Plot - This pair plot shows the relationship between the normalized "Age" and "Marital Status”

A graph of different age groups

Description automatically generated with medium confidence

Figure - Scatter Plot: Age vs Marital Status - The scatter plot visualizes the relationship between the normalized "Age" and "Marital Status"

A grid of blue dots

Description automatically generated

Figure - Bar Plot of Marital Status - This bar plot displays the frequency distribution of different marital statuses within the dataset, providing an overview of their prevalence.

A graph of a bar chart

Description automatically generated

Figure - Violin Plot of Age vs Marital Status - This violin plot illustrates the distribution of normalized "Age" across different "Marital Status" categories, showing both the density and range of values for each group.

A graph of a violin plot

Description automatically generated

These visualizations are key in receiving an understanding of the dataset. The pair plot and the scatter plot of Age vs Marital Status for example, give a clear view of the distribution of the marital categories in the dataset. The violin plot of Age vs Marital Status give information on how the Age is distributed through the marital categories.

The visualizations help spot trends and variations and enable further analysis.

1. Task 2: Classification

Classification is a type of supervised learning task that is aimed at predicting the Approximated Social Grade of people (their social class ), it is a categorical variable which values ranging from lower to higher. The task is to predict the social grade based on attributes like age, sex, hours worked per week, etc.

Four classification algorithms were applied to predict Approximated Social Grade:

* Decision Tree Classifier
* Random Forest Classifier
* K-Nearest Neighbors (KNN)
* CatBoost Classifier

Figure - Code output of Accuracy

A screenshot of a computer screen

Description automatically generated

The performance of each algorithm was evaluated using accuracy, precision, recall, and F1-Score. These metrics are great to analyze the performance of the model.

Figure - Code Output: Decision Tree Classification Report

A screenshot of a computer screen

Description automatically generated

Decision Tree Classifier:

A Decision Tree is very simple but effective algorithm that works by splitting the data into branches based on their feature values. It is easy to use but has a problem where it is very vulnerable to overfitting if the tree grows in a complicated way without enough pruning. Here is the performance of the Decision Tree Classifier:

* Accuracy: 0.806
* Precision:
  + Class 0 (0.93)
  + Class 2 (0.73)
* Recall:
  + Class 0 (0.93)
  + Class 2 (0.73)
* F1-Score:
  + Class 0 (0.93)
  + Class 2 (0.73)

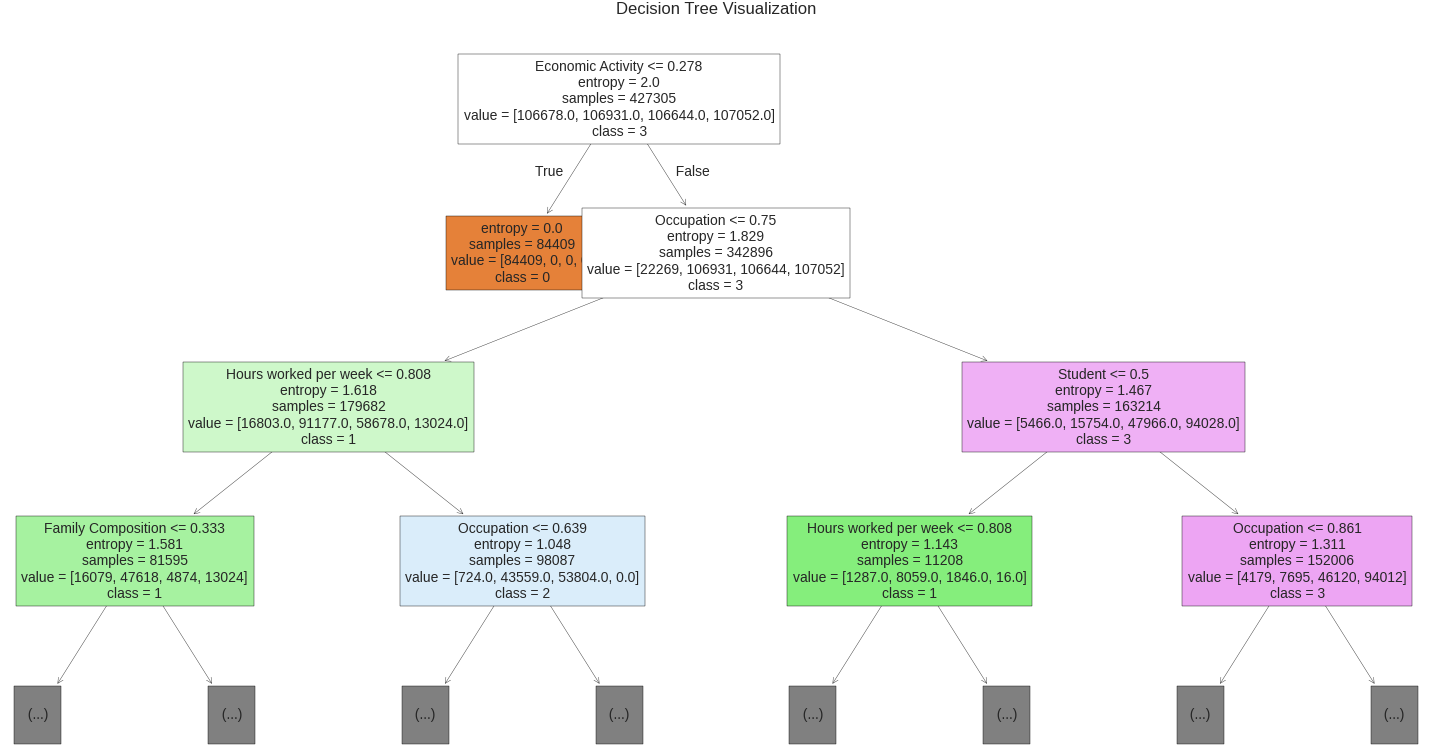


Figure - Decision Tree Visualization

Interpretation: The Decision Tree performed decent with an accuracy of 0.806. However, as shown it performs differently between classes. For class 0 it has managed a high precision, recall and f1-scores, but for class 2 it has struggled with performance which shows that it has overfitted to certain features.

Figure - Code Output: Random Forest Classification Report

A screenshot of a computer screen

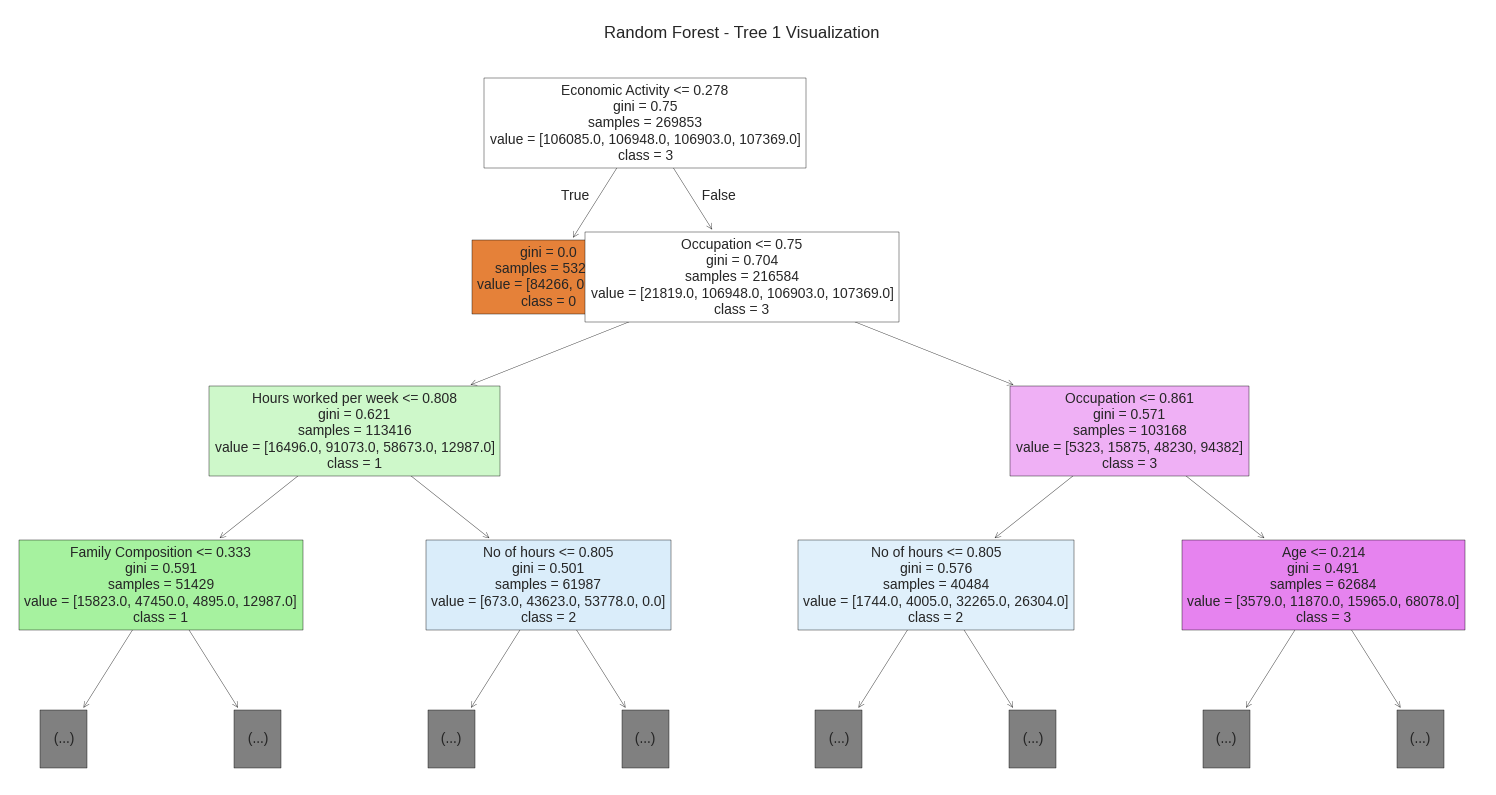
Description automatically generated

Random Forest Classifier:

Random Forest algorithm is an ensemble method that implements multiple decision trees to make predictions. It does not have the overfitting issue because it uses the predictions of many singular trees which allows it to be better and more precise than a single decision tree.

* Accuracy: 0.845
* Precision:
  + Class 0 (0.95)
  + Class 2 (0.80)
* Recall:
  + Class 3 (0.86)
  + Class 2 (0.76)
* F1-Score:
  + Class 0 (0.94)
  + Class 2 (0.78)

Figure - Code Output: Random Forest Visualization



Interpretation: Random Forest performed much better than Decision tree with an accuracy of 0.845 and shows a very good performance between all the classes especially classes 0 and 3. It shows to balance precision and recall better than Decision Tree which leads to better predictions.

Figure - Code Output: KNN Classification Report

A screenshot of a graph

Description automatically generated

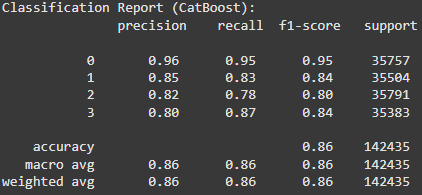
K-Nearest Neighbors (KNN):

KNN classifies data based on the majority class between their nearest neighbors. One downside of it is that it is a distance based model and it is mainly ineffective when it comes to high-dimensional data but serves as a decent baseline for classification tasks.

* Accuracy: 0.781
* Precision:
  + Class 0 (0.92)
  + Class 2 (0.69)
* Recall:
* Class 0 (0.94)
* Class 2 (0.67)
* F1-Score:
* Class 0 (0.93)
* Class 2 (0.68)

Interpretation: Between the four classifiers used, KNN has the lowest accuracy of 0.781. The only class it performed well in was class 0, for the others especially class 2 the scores were very low. This shows that KNN is not a model suitable for this dataset.

Figure - Code Output: CatBoost Classification Report



CatBoost Classifier:

CatBoost is an excellent algorithm that works particularly well with categorical data. It has mechanisms for handling categorical data that are built in which puts it in favour of use over regular tree models like the ones talked about earlier.

* Accuracy: 0.855
* Precision:
  + Class 0 (0.96)
  + Class 2 (0.81)
* F1-Score:
  + Class 0 (0.95)
  + Class 2 (0.80)

Interpretation: CatBoost has performed the best between all the algorithms. It had the highest score (0.855) and very good scores between all the classes. The model did particularly well for class 0. It seems to balance the trade off between precision and recall which is why it is the most effective classifier for the dataset.

**Comparison of Classifiers:**

* Best Model: As mentioned, the CatBoost model performed the best, it outperformed the other models in the classification scores and in the accuracy because it is great for handling categorical variables.
* Second Best: Random Forest was not too far behind the CatBoost model it has done well in most classes and it has a high accuracy. However, catboost did better in predicting class 2.
* Weakest Models: KNN and Decision tree both struggled in performing well with this dataset. They have scored lower accuracies and really struggled with some classes especially class 2.

The comparison clearly highlights that CatBoost is the most suitable algorithm for classifying social grades in this dataset, while Random Forest is a good alternative.

1. Task 3: Regression

In this task, the goal was to predict the "No of Hours" worked per week. Three regression algorithms were applied:

* Linear Regression
* Random Forest Regression
* CatBoost Regression

Mean Squared Error (MSE)**,** R²**,** and Adjusted R² were used to evaluate the performance of these algorithms.

Figure - Code Output: Scores of Linear Regression





Linear Regression:

Linear Regression is the simplest regression algorithm, it works by assuming a linear relationship between the features and the target. It is quite efficient but can struggle when the relationship between the feature and the target are non-linear.

* Mean Squared Error (MSE): 0.405
* R²: 0.676
* Adjusted R²: N/A

Interpretation: It has performed decent. An R² of 0.676 shows that almost 68% of the variance in the target is explained, but a MSE of 0.405 shows there is a lot of error in prediction. It serves as a decent baseline and its performance can be upgraded using more advanced techniques.

Figure - Code Output: Random Forest Regression Scores







Random Forest Regression:

Random Forest Regression is an ensemble learning method that works in the same was as the Random Forest Classifier, making multiple decision trees and taking the average of their predictions. It is able to capture non linear relationships and is overall more accurate than Linear Regression.

* Mean Squared Error (MSE): 0.215
* R²: 0.828
* Adjusted R²: 0.828

Interpretation: With a MSE of 0.215 and an R² score of 0.828, the model explains 83% of the variance in the target and a decent error value. The Adjusted R² is the same as the R² because it is quite a simple model but it confirms how well it fitted.

Figure - Code Ouptut: CatBoost Regression Scores







CatBoost Regression:

CatBoost is a gradient algorithm that is made to handle categorical features and complex relationships (non-linear.)

It is a good pick because it handles large datasets well and often outperforms traditional models.

* Mean Squared Error (MSE): 0.198
* R²: 0.842
* Adjusted R²: 0.842

Interpretation: It has managed the lowest MSE score of 0.198 and the highest R² of 0.842, which shows that the model was able to explain 84% of the variance in the target. The adjusted R² showed that the model has fitted very well to the data with no overfitting.

**Comparison of Regression Models:**

* Best Model: CatBoost Regression is the best model between them for predicting the number of hours worked per week as it has scored the lowest MSE and the highest R².
* Second Best: Random Forest Regression is not too far behind, as it has done well with a R² of 0.828, it excels in capturing complex relationships.
* Weakest Model: Linear Regression serves as a useful baseline but it is worse than the others in performance, with a lower R² and higher MSE compared to the other models.

The results suggest that CatBoost is the most effective model for this regression task, with Random Forest as a good alternative.

1. Task 4: Association Rule Mining

To discover relationships between attributes or patterns between different variables, association rule mining is widely used for large datasets. Two algorithms were applied to do it:

* Apriori
* FP-Growth

Figure - Code Output: Top 5 rules from Apriori

A screen shot of a computer

Description automatically generated

Apriori Algorithm:

Apriori algorithm may be the most popular algorithms for association rule mining. It operates by scanning the dataset and then marking frequent itemsets, then getting association rules from them. It works using a bottom-up approach which means those frequent individual items are extended to larger itemsets.

Here are the top 5 rules discovered using the Apriori algorithm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Antecedent | Consequent | Support | Confidence | Lift |
| {Yes, Male, UK, Part-time: 15 or less} | {AB} | 0.285714 | 1.0 | 3.5 |
| {White, Single} | {Very good health, 1} | 0.285714 | 1.0 | 3.5 |
| {AB, UK} | {Male, Part-time: 15 or less} | 0.285714 | 1.0 | 3.5 |
| {Male, AB} | {UK, Part-time: 15 or less} | 0.285714 | 1.0 | 3.5 |
| {Part-time: 15 or less} | |  | | --- | | {AB, Male, UK} |  |  | | --- | |  | | 0.285714 | 1.0 | 3.5 |

Interpretation:

* Having a support score of 0.2857 means that these specific rules are present in about 29% of the dataset which means that they are quite frequent.
* Having a confidence score of 1.0 shows just full confidence in the association, which means that anytime the antecedent happens, the consequent will surely happen.
* Having a lift of 3.5 means that, when the antecedent is present, the chance of the consequent happening is 3.5 times higher which shows strong relationships between them.

Figure - Code Output: Top 5 rules from FP-Growth

A screenshot of a computer

Description automatically generated

FP-Growth Algorithm:

FP-Growth is another method for mining for frequents. However, it does not explicitly generate candidate itemsets. But it uses instead a tree structure called the DP-tree to mine efficiently.

Top 5 rules using the FP-Growth algorithm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Antecedent | Consequent | Support | Confidence | Lift |
| {Very good health, 1} | {UK, Single} | 0.285714 | 1.0 | 3.5 |
| {Male, UK, Part-time: 15 or less} | {AB} | 0.285714 | 1.0 | 3.5 |
| {AB, Male} | {UK, Part-time: 15 or less} | 0.285714 | 1.0 | 3.5 |
| {UK, White, 1} | {Single} | 0.285714 | 1.0 | 3.5 |
| {Male, Part-time: 15 or less} | {AB, UK} | 0.285714 | 1.0 | 3.5 |

Interpretation:

* FP-Growth support score is 0.2857, confidence of 1.0 and lift of 3.5. The rules indicate strong relationships between features such as health, sex, and employment status (e.g., "Male, UK, Part-time: 15 or less" → "AB").
* The patterns are very consistent and reliable which shows that the specific combinations of demographic and employment features are very tightly linked with social grades or conditions.

**Comparison of Apriori and FP-Growth:**

Both Apriori and GP-Growth have outputted very similar results for the support, confidence, and lift which means that the patters that have been found are reliable. In general, FP-Growth shows to be more efficient than Apriori especially when the dataset is large because it doesn't need to generate candidate itemsets explicity.

Both algorithms reveal interesting insights into the relationships between various demographic and employment features, which can be used to understand how different characteristics are related to specific social grades.

1. Task 5: Clustering

Clustering is an unsupervised learning task, it works by gathering and grouping data into separate clusters depending on their similarities. Three clustering algorithms were applied in this task to cluster the data in the dataset:

* KMeans
* Agglomerative Clustering
* KMedoids

The metric used to assess the performance of these models is the Silhouette Score, this score assesses the similarity between each data is to the cluster it is in and compares it with other clusters. A high silhouette score means that clusters and separated well and are not too similar and vice versa.

Figure - Code Output: KMeans Performance Scores and Visualizations





A graph of a line graph

Description automatically generated A yellow and purple blobs

Description automatically generated

KMeans Clustering:

KMeans is a very popular clustering algorithm, it splits the dataset into a predefined amount of clusters and then starts decreasing the difference within each cluster. KMeans relies on the placement of centroids and can perform worse with differently shaped clusters.

* Silhouette Score (Pre-PCA): 0.191
* Silhouette Score (Post-PCA): 0.450

Interpretation: The algorithm had a low score of 0.191 which shows that the clusters were not very different. However when Principal Component Analysis was applied, the score improved to a 0.450 which shows that reduced dimensionality has aided in segmenting the clusters better.

Figure - Code Output: Agglomerative Clustering Performance Scores and Visualizations





A graph with a purple line

Description automatically generated A purple and yellow dots

Description automatically generated

Agglomerative Clustering:

Agglomerative Clustering is a hierarchical clustering method that works with distance, it starts with each data point as its own cluster and then merges the clusters. If the clusters are not spherical or have a different shape then it mostly performs better than KMeans.

* Silhouette Score (Pre-PCA): 0.553
* Silhouette Score (Post-PCA): -0.040

Interpretation: The algorithm has had a decent score of 0.553 which shows the clusters have been separated well. However, after PCA was applies the score plummeted to -0.04, which clearly shows that dimensionality damaged the performance of it as important information was lost during the PCA application.

Figure - Code Output: KMedoids Performance Scores and Visualizations





A graph of a line with dots

Description automatically generated A yellow and purple dots

Description automatically generated

KMedoids Clustering:

KMedoids is similar to KMeans but uses data points as medoids (the centers) instead of the centroids. It is overall less efficient than KMeans but can work well with noise and outliers.

* Silhouette Score (Pre-PCA): 0.051
* Silhouette Score (Post-PCA): 0.023

Interpretation: The silhouette score was extremely poor for the KMedoids both before (0.051) and after PCA (0.023). This shows that it was not effective at all in clustering this dataset mostly because noise is present and the feature space is quite complex.

1. Conclusion

This report applied multiple various machine learning techniques to the 2011 UK Census dataset. There were multiple tasks and goals; predict, classify, find patters, cluster and segment people based on similarity in their features:

* 1. Descriptive Analytics:

For the first task, it is aimed at exploring the key statistics of the dataset such as the mean, standard deviation, and range of important features such as age, region, sex, and social grade. These statistics allowed for a good understanding of the structure of this dataset and opened the door for deeper analyses.

* 1. Classification:

For the classification, four algorithms were used: Decision Tree, Random Forest, KNN, and CatBoost. The target was to predict the Approximated Social Grade. From the analysis of results, CatBoost has performed the best between all of them, it has achieved the highest accuracy and done well between all the classes. Random Forest was not too far behind, Decision Tree and KNN were the worst especially in prediciting class 2 (social grade C).

* 1. Regression:

For regression, three algorithms were applied: Linear Regression, Random Forest Regression, and CatBoost Regression. The aim was to predict the No of hours worked per week based on different attributes. Again, CatBoost was the most effective model between them as it has achieved the lowest MSE and the highest R² score. Random Forest also did well, but the Linear Regression algorithm was bad in comparison.

* 1. Association Rule Mining:

For this task two algorithms were used: Apriori and FP-Growth, these helped unveil some strong association rules between the features. They have both found rules with great confidence and a lift of 3.5 which shows the clear relationships in features like employment type, sex, and social grade.

* 1. Clustering:

KMeans, Agglomerative Clustering, and KMedoids are the algorithms used to group and cluster people based on similarity in their features. The Agglomerative Clustering algorithm did the best before applying PCA, after applying it its performance worsened. KMeans showed a good improvement in score after PCA. KMedoids really struggled before and after PCA.