**MSc in Data Science**

ITC 6103B1 – Applied Machine Learning

**Final Group Project**

**Instructor:**

Dr. E. Chatzimichali

**A logo with text on it

Description automatically generated**

**Submitted to:**

Deree – The American College of Greece

**Submitted by:**Kapsalis C. (275482)

Stavrogiannis C. (225765)

Tzoras K. (312632)

**Submission Date:**

06/04/2023

Table of Contents

[Table of Figures 4](#_Toc163318891)

[Regression Analysis 5](#_Toc163318892)

[Data Description & Scope of Analysis 5](#_Toc163318893)

[Data Exploration 5](#_Toc163318894)

[Data Pre-Processing 5](#_Toc163318895)

[Purpose 5](#_Toc163318896)

[Pre-Processing on the Features Matrix 5](#_Toc163318897)

[Fitting & Tuning Predictive Models 6](#_Toc163318898)

[Model Selection 6](#_Toc163318899)

[Model Performance 6](#_Toc163318900)

[Conclusions & Future Work 14](#_Toc163318901)

[Clustering 15](#_Toc163318902)

[Data Description & Analysis 15](#_Toc163318903)

[Data Exploration 15](#_Toc163318904)

[Data Preprocessing 17](#_Toc163318905)

[Dimensionality Reduction 19](#_Toc163318906)

[Clustering Algorithms 19](#_Toc163318907)

[Comparison of Clustering Algorithms 21](#_Toc163318908)

[Results and insights 21](#_Toc163318909)

[Conclusions & Future Work 25](#_Toc163318910)

[Classification 26](#_Toc163318911)

[Data Description & Scope of Analysis 26](#_Toc163318912)

[Purpose 26](#_Toc163318913)

[Data Exploration 26](#_Toc163318914)

[Purpose 26](#_Toc163318915)

[Key Findings 26](#_Toc163318916)

[Data Pre-Processing 31](#_Toc163318917)

[Purpose 31](#_Toc163318918)

[Description of Data Structures Used (separate X and y dataframes for the training and testing data) 32](#_Toc163318919)

[Pre-Processing on the Dependent Variable 32](#_Toc163318920)

[Pre-Processing on the Features Matrix 33](#_Toc163318921)

[Fitting & Tuning Predictive Models 34](#_Toc163318922)

[Model Selection 34](#_Toc163318923)

[Model Performance 35](#_Toc163318924)

[Explainable AI 36](#_Toc163318925)

[Purpose & Method Selection 36](#_Toc163318926)

[Conclusions & Future Work 37](#_Toc163318927)

[REFERENCES 38](#_Toc163318928)

[APPENDIX 40](#_Toc163318929)

[Regression 40](#_Toc163318930)

[Regression Analysis 40](#_Toc163318931)

[Data Exploration 40](#_Toc163318932)

[Data Description 41](#_Toc163318933)

[Data Pre-processing 41](#_Toc163318934)

[Clustering 43](#_Toc163318935)

[Davies Bouldin Index 43](#_Toc163318936)

[Calinski-Harabasz 43](#_Toc163318937)

[Brief explanation of algorithms and hyperparameters 43](#_Toc163318938)

[PCA 43](#_Toc163318939)

[OHE 43](#_Toc163318940)

[Classification 44](#_Toc163318941)

[Brief Description of Data Files Used 44](#_Toc163318942)

[Challenges 55](#_Toc163318943)

[Findings 55](#_Toc163318944)

# Table of Figures

[Figure 1: Top 10 Feature based on Cumulative Importance 9](#_Toc163318813)

[Figure 2: Average performance of top 10 models. 11](#_Toc163318814)

[Figure 3 : Average performance of top 10 models. 11](#_Toc163318815)

[Figure 4: Random Forest performance using default parameters and standard scaler. 13](#_Toc163318816)

[Figure 5: Gradient Boosting performance using default parameters and standard scaler. 13](#_Toc163318817)

[Figure 6: H. Gradient Boosting performance using default parameters and standard scaler. 14](#_Toc163318818)

[Figure 7: CatBoost performance using default parameters and standard scaler. 14](#_Toc163318819)

[Figure 8: Sum of spendings based on Channel, Region pair. 17](#_Toc163318820)

[Figure 9: Count of expenditures per category in Channel, Region pair. 18](#_Toc163318821)

[Figure 10: Sum of expenditures across all the dataset. 18](#_Toc163318822)

[Figure 11: Frequency in relation to Region. 19](#_Toc163318823)

[Figure 12: Sales based on Channel, region is considered as either Other region or Major Cities. 19](#_Toc163318824)

[Figure 13: Identification and mapping of numeric categorical values in region and channel. 20](#_Toc163318825)

[Figure 14: Replaced object values on the original dataset, to avoid biases. 20](#_Toc163318826)

[Figure 15:Results before scaling and OHE. 20](#_Toc163318827)

[Figure 16:Results after scaling and OHE. 21](#_Toc163318828)

[Figure 17: Cumulative explained variance. 21](#_Toc163318829)

[Figure 18: Finding optimal k with Elbow Method and Silhouette Score. 22](#_Toc163318830)

[Figure 19: Linkage method to find clusters in original dataset. 22](#_Toc163318831)

[Figure 20:Linkage method to find clusters in pca dataset. 22](#_Toc163318832)

[Figure 21: Top-3 results scores on all 3 metrics for then scaled original dataset. 23](#_Toc163318833)

[Figure 22:Top-3 results scores on all 3 metrics for then scaled PCA dataset. 23](#_Toc163318834)

[Figure 23: Importance of initial features for each PC. 24](#_Toc163318835)

[Figure 24: Average spending per PC and by Cluster. 24](#_Toc163318836)

[Figure 25: Average spending per Cluster and by initial feature. 25](#_Toc163318837)

[Figure 26: Average spending per original feature and by Cluster. 25](#_Toc163318838)

[Figure 27: Average spending per Cluster and by PC. 26](#_Toc163318839)

[Figure 28: Income Subgroups per Weekly Workload Type 29](#_Toc163318840)

[Figure 29: Frequency of Education Levels among US Adults 30](#_Toc163318841)

[Figure 30: Participation of Education-based segments to Income groups 31](#_Toc163318842)

[Figure 31: Frequency of Occupations among US Adults 32](#_Toc163318843)

[Figure 32: Occupational Segment Breakdown by Income Group 32](#_Toc163318844)

[Figure 33: ‘Workclass’ Segment Breakdown by Income Group 33](#_Toc163318845)

[Figure 34: Occupation Frequency among Long Week Workers 47](#_Toc163318846)

[Figure 35: Marital Status Relevance among Long Week Workers 48](#_Toc163318847)

[Figure 36: Frequency of the Sexes among US Adults 49](#_Toc163318848)

[Figure 37: Sexes Segment Breakdown by Income Group 49](#_Toc163318849)

[Figure 38: Race Segment Breakdown by Income Group 50](#_Toc163318850)

[Figure 39: Heatmap of Missing Values, Testing Set 53](#_Toc163318851)

[Figure 40: Distributions of Columns containing Missing Values 54](#_Toc163318852)

[Figure 41: Boxplot of selected Numeric Features after Standardization 55](#_Toc163318853)

[Figure 42: Mean Absolute SHAP Values per Feature 57](#_Toc163318854)

[Figure 43: SHAP Summary Plot 58](#_Toc163318855)

[Figure 44: SHAP Scores for Specific PC1 Values 59](#_Toc163318856)

[Figure 45: 5 Initial Features Contributing most to PC1's Values 60](#_Toc163318857)

[Figure 46: SHAP Scores for Specific PC2 Values 61](#_Toc163318858)

[Figure 47: 5 Initial Features Contributing most to PC2's Values 62](#_Toc163318859)

Regression Analysis

The following chapters summarize the processes we used to model property prices by utilizing an assortment of features and the conclusions derived by them. The restraints of the report necessitate that only the highlights of the analysis be presented, therefore the details are present in the relevant Appendix chapters.

For details regarding our perspective on regression analysis, please consult the relevant Appendix part.

Data Description & Scope of Analysis

The dataset used for our purpose consists of 1460 property samples of 79 features, and their respective sale prices. There is plethora of precedence of an analysis such as this in the literature, although the resulting models fail to entirely capture the dynamic effects applied by these features over different time frames (Foryś, 2022). More on data description on Appendix I.

Given the limitations set by the nature of this endeavor, as well as the accessibility of all the known and unknown causes of a property sale price, this analysis has the goal of finding an efficient way of producing the best performing models to describe the relationship between the features of this dataset and property sale prices.

Data Exploration

For details regarding our perspective on data exploration, please consult the relevant Appendix part.

Based on that understanding, we examined the dataset for duplicate samples, missing values and partitioned the features to numerical and categorical for further exploration. For the categorical features, we produced histograms and calculated the descriptive statistics of the sale price for each category. Numerical data exploration involved descriptive statistics, correlation matrices and line-charts with the dependent variable, as well as outlier detection and histograms.

After this initial exploration, every feature was checked per sample for the validity of its values. For example, checking for the existence of a property sample for which the house was built after it was sold.

For key findings from the data exploration, please consult the relevant Appendix part.

Data Pre-Processing

Purpose

Most of the algorithms used to model the relationship between sale price and the above features cannot handle missing values and require numerical data that fall on a similar scale. Part of the preprocessing includes separating the independent from the dependent variables as well as splitting the data to train and test sets.

The Pareto Principle (Newman, 2005), 80/20 rule, was used for the train-test split, due to the simplicity of implementation in dynamic feature selection, over the Scaling Law (Guyon, 1997), even though it is shown to perform better (Detective, 2020).

Pre-Processing on the Features Matrix

Most of the null values encountered were from categorical features, whose null value represented the absence of that feature in a sample. For example, null values on features like ‘BsmtQual’ or ‘BsmtCond’ represented properties without basements. Such null values where filled with zeros. One exception to this was the feature describing the garage built year, for which the null values were filled with the house built year for each sample. Even though the value is wrong, it was a quick way of filling those with a number that would not create any major bias.

Categorical features that objectively contribute to the actual value of a property were encoded manually. For details on the features and the scales used please examine Appendix. The remaining categories were subjected to One-Hot Encoding.

The scaling algorithms require that the columns of the data used to fit scaler are indentical to those being transformed afterwards. Since there was a difference in columns between the training and testing set, the missing columns were added. Adding the missing columns on the training set filled with zeros was not beneficial, considering that the algorithms training on that set would not ‘learn’ from features with zero variance. But since it was decided to keep us many features as possible for the upcoming feature selection, this was an inconsequential oversight.

Fitting & Tuning Predictive Models

Model Selection

The regression models involved in this analysis were the following:

* Linear
* Polynomial
* Lasso
* Ridge
* Random Forest
* Gradient Boosting
* Histogram Gradient Boosting
* CatBoost

Fitting and tuning can be split into three phases, based on the features used. Initially every feature was used on all these models. Based on the feature importance of all the models, the features were sorted and then the top 10, 30 and 50 features were used for all the models again. Both phases involved unscaled datasets, as well as datasets transformed by the Standard and the MinMax scaler of the sklearn library. Lastly, the Random Forest, the Gradient Boosting and the CatBoost models were run for every number of features on the list of features sorted by cumulative importance.

At the end of the analysis, a sequential Deep Neural Network was deployed to compare its performance with the rest of our results. The parameters of that network were for the most part arbitrary.

Model Performance

The top 10 features sorted by cumulative importance are presented on the bar-plot below:

A graph of blue bars with white text

Description automatically generated

Figure 1: Top 10 Feature based on Cumulative Importance

Below follow all the top 10 models based on their coefficient of determination.

Table 1: Top 10 models using all features.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Parameters** | **Scaler** | **Time** (s) | **r2** | **RMSE** | **MAE** |
| CatBoost | Default | Standard | 3.55 | 0.929 | 20559 | 14257 |
| CatBoost | Default | None | 4.02 | 0.929 | 20573 | 14270 |
| CatBoost | Default | MinMax | 3.68 | 0.929 | 20573 | 14270 |
| CatBoost | Optimized | None | 29.76 | 0.928 | 20625 | 14562 |
| CatBoost | Optimized | MinMax | 29.81 | 0.928 | 20625 | 14562 |
| Gradient Boosting | Default | Standard | 1.14 | 0.916 | 22376 | 15749 |
| Gradient Boosting | Default | None | 1.22 | 0.916 | 22377 | 15750 |
| Gradient Boosting | Default | MinMax | 1.21 | 0.916 | 22377 | 15750 |
| Gradient Boosting | Optimized | MinMax | 207.50 | 0.915 | 22418 | 15515 |
| Random Forest | Optimized | Standard | 221.75 | 0.913 | 22714 | 16024 |

Table 2: Top 10 models using the top 50 features based on cumulative importance.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Parameters** | **Scaler** | **Time** (s) | **r2** | **RMSE** | **MAE** |
| CatBoost | Default | Standard | 2.486 | 0.925 | 21089 | 14908 |
| CatBoost | Default | MinMax | 2.561 | 0.925 | 21087 | 14906 |
| CatBoost | Default | None | 2.786 | 0.925 | 21087 | 14907 |
| CatBoost | Optimized | MinMax | 21.764 | 0.922 | 21580 | 15164 |
| CatBoost | Optimized | None | 21.600 | 0.922 | 21580 | 15164 |
| Gradient Boosting | Optimized | Standard | 393.233 | 0.919 | 21975 | 15690 |
| Gradient Boosting | Default | Standard | 0.882 | 0.915 | 22462 | 15921 |
| Gradient Boosting | Default | MinMax | 1.075 | 0.915 | 22471 | 15933 |
| Gradient Boosting | Default | None | 0.883 | 0.915 | 22471 | 15933 |
| Gradient Boosting | Optimized | MinMax | 406.308 | 0.914 | 22531 | 15873 |

Table 3: Top 10 models using the top 30 features based on cumulative importance.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Parameters** | **Scaler** | **Time** (s) | **r2** | **RMSE** | **MAE** |
| CatBoost | Default | Standard | 2.565 | 0.925 | 21064 | 14932 |
| CatBoost | Default | MinMax | 2.542 | 0.925 | 21047 | 14910 |
| CatBoost | Default | None | 2.523 | 0.925 | 21062 | 14927 |
| CatBoost | Optimized | MinMax | 21.886 | 0.924 | 21197 | 14963 |
| CatBoost | Optimized | None | 22.587 | 0.924 | 21197 | 14963 |
| Gradient Boosting | Default | Standard | 0.717 | 0.916 | 22286 | 15778 |
| Gradient Boosting | Default | MinMax | 0.722 | 0.916 | 22288 | 15784 |
| Gradient Boosting | Default | None | 1.198 | 0.916 | 22288 | 15782 |
| Gradient Boosting | Optimized | MinMax | 442.291 | 0.915 | 22467 | 15885 |
| Gradient Boosting | Optimized | None | 437.456 | 0.915 | 22522 | 16014 |

Table 4: Top 10 models using the top 10 features based on cumulative importance.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Parameters** | **Scaler** | **Time** (s) | **r2** | **RMSE** | **MAE** |
| H. Gradient Boosting | Optimized | Standard | 6.237 | 0.898 | 24627 | 17864 |
| H. Gradient Boosting | Optimized | MinMax | 10.909 | 0.893 | 25150 | 18333 |
| Gradient Boosting | Optimized | MinMax | 239.261 | 0.892 | 25277 | 18513 |
| Gradient Boosting | Default | MinMax | 0.426 | 0.891 | 25429 | 18395 |
| H. Gradient Boosting | Default | Standard | 0.431 | 0.891 | 25431 | 18348 |
| Gradient Boosting | Default | None | 0.443 | 0.891 | 25437 | 18398 |
| Gradient Boosting | Optimized | None | 219.216 | 0.891 | 25388 | 18658 |
| Gradient Boosting | Default | Standard | 0.432 | 0.891 | 25435 | 18392 |
| H. Gradient Boosting | Default | None | 0.442 | 0.891 | 25431 | 18348 |
| H. Gradient Boosting | Default | MinMax | 0.364 | 0.891 | 25434 | 18337 |

All the performance results can be found on the submitted Jupyter notebook. Some of the important average performance plots can be found below, with the processing time being represented by the red line:

A green and red line graph

Description automatically generated

Figure 2: Average performance of top 10 models.

A graph with a red line

Description automatically generated

Figure 3 : Average performance of top 10 models.

Scaler performance information is presented below, based on average performance of all models:

Table 5: Performance without Scaler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of Features** | **R2** | **RMSE** | **MAE** | **Proccessing Time (s)** |
| 10 | 0.876 | 27134.167 | 20065.333 | 53.318 |
| 30 | 0.896 | 24706.167 | 17766.667 | 105.235 |
| 50 | 0.902 | 24073.667 | 17146.167 | 108.502 |
| 182 | 0.889 | 25558.700 | 18289.900 | 92.386 |

Table 6: Performance with MinMax Scaler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of Features** | **R2** | **RMSE** | **MAE** | **Proccessing Time (s)** |
| 10 | 0.877 | 27003.000 | 19928.333 | 58.193 |
| 30 | 0.893 | 25054.000 | 17998.500 | 105.825 |
| 50 | 0.900 | 24246.667 | 17254.833 | 105.293 |
| 182 | 0.887 | 25616.700 | 18499.200 | 81.440 |

Table 7: Performance with Standard Scaler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of Features** | **R2** | **RMSE** | **MAE** | **Proccessing Time (s)** |
| 10 | 0.716 | 35949.500 | 28172.833 | 63.459 |
| 30 | 0.743 | 34040.833 | 26379.500 | 89.701 |
| 50 | 0.752 | 33084.333 | 25476.000 | 101.799 |
| 182 | 0.795 | 31251.200 | 23504.000 | 68.957 |

Phase 3 results with default parameters and standard Scaler:

Table 8: Model performance with default parameters and standard scaler

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Number of Features** | **R2** | **Proccessing Time** (s) |
| Random Forest | 98 | 0.910 | 4.913 |
| Gradient Boosting | 33 | 0.920 | 0.845 |
| H. Gradient Boosting | 64 | 0.912 | 0.573 |
| CatBoost | 62 | 0.933 | 0.671 |

Below we can observe the performance of these models for each number of features, with the red line representing the processing time and the green the coefficient of determination.

A graph of a number of features

Description automatically generated

Figure 4: Random Forest performance using default parameters and standard scaler.

A graph of a number of features

Description automatically generated

Figure 5: Gradient Boosting performance using default parameters and standard scaler.

A graph showing the number of features

Description automatically generated

Figure 6: H. Gradient Boosting performance using default parameters and standard scaler.

A graph showing the number of features

Description automatically generated

Figure 7: CatBoost performance using default parameters and standard scaler.

Performance results from the DNN using all 182 features:

Table 9: Sequential DNN performance using all features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scaler** | **Time** (s) | **r2** | **RMSE** | **MAE** |
| MinMax | 9.532 | 0.905 | 23721 | 17424 |
| Standard | 10.708 | 0.839 | 30875 | 20552 |
| None | 11.052 | 0.797 | 34729 | 23589 |

Conclusions & Future Work

The results of this analysis assist on visualizing the fact that features, scalers, models and their parameters are all interdependent. The best performance was achieved by using the default parameters of CatBoost on the top 62 standardized features, achieving 0.933 coefficient of determination in 0.6 seconds. The fact that many tuned models didn’t increase in performance compared to the respective defualt ones, means that the tuning ranges and the number of iterations require further examination.

Achieving anything closer to perfect explainability of property sale price would require more features, for example inflation indexes, considering the complexity of modeling a stochastic process such as this, that involves biases beyond the scope of the dataset used.

At certain points, this analysis forced our hardware to its limit, while for most models the number of iterations used to find the optimal parameters was set low enough so that results are retrieved in a relative short time. This indicates that increasing the complexity of our models would require an abnormal system that is not currently available to us.

Due to these hardware limitations, our analysis lacks in several aspects. First we would like to try and develop the tuning algorithms to include feature selection for the specific model and scaler. Secondly, we would like to include in this algorithm the polynomial transformation of the scaled data, for several degrees of which the algorithm selects the most appropriate based on accuracy. Thirdly, we would like to implement in that tuning dynamic splitting of training and testing sets based on percentages from the Scaling Law. Adding this much of complexity in our algorithms would require an immense upgrade of hardware, in terms of memory and processing speed.

In closing, even implementing such level of complexity which would certainly increase the explainability of property sale prices by the resulting model, there would still be a portion of variations that cannot be captured by it and on the other hand increase the cost of the model, in terms of time and money, by several orders of magnitude.

# Clustering

## Data Description & Analysis

Consumer behavior analysis is pivotal in understanding market dynamics and forming effective marketing strategies. In this report, we delve into clustering consumer spending data to uncover insights into purchasing patterns and customer segmentation. The dataset includes detailed information on consumer purchases across product categories, regions, and channels. It consists of 8 features, including 6 numerical features related to product prices and purchase locations in specific regions, with a total of 440 samples. Region and Channel features are categorical, with three distinct types for Region and two for Channel. The dataset focuses on consumer behavior in Portugal, emphasizing purchases through Horeca or Retail channels in regions such as Porto, Lisbon, or other areas. Our main goal is to identify the most suitable clustering algorithm for grouping similar consumers based on spending behavior.

## Data Exploration

A graph of different colored squares

Description automatically generated

Figure 8: Sum of spendings based on Channel, Region pair.

A graph of different colored squares

Description automatically generated

Figure 9: Count of expenditures per category in Channel, Region pair.

A pie chart with different colored sections

Description automatically generated

Figure 10: Sum of expenditures across all the dataset.

It's important to note that although the total expenditure differs across product categories and Channel-Region pairs, the count (number of sales) remains consistent. This suggests that the percentage of sales for each product is uniform across all Channel-Region pairs, with each product count representing 1/6 of the total count of sales in each Channel-Region pair. This uniform distribution implies balanced representation of product purchases within each market segment.

A graph with a blue rectangle

Description automatically generated

Figure 11: Frequency in relation to Region.

A comparison of a bar graph

Description automatically generated with medium confidence

Figure 12: Sales based on Channel, region is considered as either Other region or Major Cities.

## Data Preprocessing

The preprocessing stage encompasses several crucial steps aimed at cleansing and transforming raw data to prepare it for clustering algorithms. Initially, we conducted checks for null values, which returned zero instances of missing data. Similarly, duplicate values were examined, and none were found.

We converted numerical and categorical values in the Channel and Region features to object type and categorical values to facilitate later application of one-hot encoding (OHE). This adjustment aims to transform categorical variables into numerical values suitable for clustering algorithms, mitigating biases introduced by the initial approach's mix of categorical and numerical values (Singh, 2023). Initially, we replaced numerical labels with categorical labels for the "Region" and "Channel" features to handle categorical variables effectively. Subsequently, we applied one-hot encoding to convert these variables into a suitable format for clustering algorithms. Scaling was then performed after one-hot encoding to ensure feature scaling consistency. Additionally, we included two additional columns that hold the total sum of expenditures for each sample and a summation of food-related products, providing further insights into consumer spending behavior (Sharma, 2024).

A black text on a white background

Description automatically generated

Figure 13: Identification and mapping of numeric categorical values in region and channel.

A screen shot of a computer code

Description automatically generated

Figure 14: Replaced object values on the original dataset, to avoid biases.

A graph with numbers and lines

Description automatically generated with medium confidence

Figure 15:Results before scaling and OHE.

A graph with different colored squares

Description automatically generated with medium confidence

Figure 16:Results after scaling and OHE.

## Dimensionality Reduction

One of the primary challenges in analyzing consumer behavior data is dealing with high-dimensional datasets. Although our dataset initially comprised only 8 features, we still applied PCA to address potential issues with dimensionality. After scaling the data, we proceeded to apply PCA for dimensionality reduction (IBM, n.d.).

Most of the subsequent actions and steps were applied to both the scaled initial dataset and the dataset after applying PCA. We chose to retain 6 Principal Components (PCs) since they collectively explain approximately 95% of the variance in our original dataset.

It's worth noting that the first two PCs explain approximately 57% of the dataset's variance. As a result, any clustering analysis based on the comparison between these two components will provide a good overview of the dataset's structure and patterns (Appendix PCA).

A graph with a line going up

Description automatically generated

Figure 17: Cumulative explained variance.

Clustering Algorithms

With preprocessed data, we applied three popular clustering algorithms: KMeans, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), and Agglomerative Clustering. The choice depends on factors like data nature, desired number of clusters, result interpretability, and hyperparameter settings (Appendix Algorithm Hyperparameters).

A graph of a graph of a graph

Description automatically generated with medium confidence

Figure 18: Finding optimal k with Elbow Method and Silhouette Score.

A diagram of a diagram

Description automatically generated

Figure 19: Linkage method to find clusters in original dataset.

A diagram of a city

Description automatically generated

Figure 20:Linkage method to find clusters in pca dataset.

While clustering results cannot be solely evaluated based on metrics, we've utilized multiple metrics including Silhouette score, Davies-Bouldin Index, and Calinski-Harabasz Index. After determining the best algorithm and parameters, applied to both the initial scaled dataset and the PC-derived dataset, we'll proceed with result analysis. This step is crucial to form clusters and derive insights into distinguishing features and attributes among different groups, guiding strategic business decisions.

## Comparison of Clustering Algorithms

To determine the optimal clustering algorithm, we compared KMeans, DBSCAN, and Agglomerative Clustering using evaluation metrics such as the Davies-Bouldin Index (Linkedin, 2023) and the Calinski-Harabasz Index. These metrics offer insights into cluster quality and coherence, aiding in assessing algorithm effectiveness. Based on our analysis, KMeans proved most effective for this dataset, with DBSCAN performing least effectively. Additionally, we achieved higher Calinski-Harabasz scores, indicating better clustering separation (Ribeiro, 2024) (Appendix DBI, CHI).

A screenshot of a computer screen

Description automatically generated

Figure 21: Top-3 results scores on all 3 metrics for then scaled original dataset.

A screenshot of a graph

Description automatically generated

Figure 22:Top-3 results scores on all 3 metrics for then scaled PCA dataset.

## Results and insights

The clustering results unveil distinct consumer segments based on spending behavior, offering valuable insights into consumer preferences and purchasing habits. Each cluster displays unique characteristics, aiding market segmentation and targeted marketing strategies. Notably, Cluster 3 prioritizes spending on "fresh" items, while Cluster 4 emphasizes "Detergents\_Paper". Clusters 1, 3, and 4 exhibit relatively poor spending habits, while Cluster 0 demonstrates the highest spending, followed by Cluster 2. Cluster 1 allocates its spending primarily towards "fresh" products and all clusters tend to spend more through the Horeca Channel.

Analyzing our Principal Components (PCs) sheds light on the importance of original features and their influence on newly formed clusters from the PCA dataset. Specifically, examining the significance of the HORECA\_Channel feature in PC2 and PC6 (0.369949-0.497564, respectively, Figure 14) and comparing this with the results in Figure 15, we deduce that Cluster 2 primarily spends through the Retail channel despite its higher spending. This observation is reinforced by the analysis of Cluster 2 in PC6. Thus, the significance of the HORECA\_Channel feature in PC2 and PC6 implies that Cluster 2 predominantly utilizes the Retail channel for spending.

A table with numbers and letters

Description automatically generated

Figure 23: Importance of initial features for each PC.

A graph with different colored lines

Description automatically generated

Figure 24: Average spending per PC and by Cluster.

Cluster 1 spends significantly in PC1, focusing on Detergents\_Paper, Milk, and Groceries, as indicated by the importance of these features in PC1 (Figure 14). Their preference for Retail channels suggests an inclination towards purchasing everyday household essentials and groceries from retail outlets. This insight can inform targeted marketing strategies and product offerings for Cluster 1 consumers.

A graph with different colored lines

Description automatically generated

Figure 25: Average spending per Cluster and by initial feature.

A graph with colorful bars

Description automatically generated with medium confidence

Figure 26: Average spending per original feature and by Cluster.

Cluster 3 highly relates (has high values for PC3) PC3 is highly influenced by:

A group of black text

Description automatically generated

Cluster 3 operates mainly in Lisbon and Porto, with less activity in rural areas, suggesting it likely comprises individuals residing in major cities of Portugal, as supported by Figures 14 and 15. Similar inferences apply to Cluster 4. Additionally, considering information from PC4, Cluster 4 operates in major cities like Cluster 3 but predominantly focuses on Porto rather than Lisbon. This highlights the importance of considering geographic location and regional preferences in segmenting consumer behavior.

A group of black text

Description automatically generated

Delicatessen emerges as the most important feature for PC5. Consequently, it's evident that not all clusters allocate spending towards Delicatessen. This observation underscores the importance of considering feature importance and its impact on different clusters when analyzing consumer behavior and segmentation.

A graph of different colored lines

Description automatically generated

Figure 27: Average spending per Cluster and by PC.

The previous conclusion holds true for only 3 out of 5 clusters. Cluster 3 doesn't spend on delicatessen, while Cluster 4 spends the most (Figure 18). These clusters counterbalance each other. Based on the insignificant values in the other 3 clusters, the original assumption that PC3 and the Delicatessen feature (most important for PC3) are unimportant for all clusters is incorrect.

Conclusions & Future Work

In conclusion, clustering analysis offers valuable insights into consumer behavior and segmentation, enabling businesses to better understand their target market. KMeans emerged as the preferred clustering algorithm for this dataset, especially when applied to the PCA dataset. Additionally, actionable insights were derived from the final analysis, aided by creating a mapping of the importance of the original features to the final Principal Components (PCs). Furthermore, comparing the clusters resulted from both datasets provided valuable information. However, a notable approach missing is focusing on segmentation based on the region and deriving more insights on consumer behavior by analyzing the topology of the shops. Such analysis and refinement of clustering techniques could enhance our understanding of consumer behavior and drive business growth in an increasingly competitive market landscape.

# Classification

## Data Description & Scope of Analysis

### **Purpose**

The dataset[[1]](#footnote-1) we worked on comes from the Adult Population Census carried out in the United States in 1994. It contains information on the socio-economic characteristics of various individuals in the United States. For each entry, we also have an indication on whether they made more or less than $50,000 that year.

The goal of our analysis was to build predictive models that will allow us to predict whether a person makes more than $50,000 a year or not, given some specific socio-economic attributes of theirs. Since the available data used to train the models on are labeled and our goal is to predict the value of a discrete variable, this is a case of supervised learning, and, in particular, of binary classification.

The dataset files that we needed to use in our analysis were ‘adults.data’, ‘adults.test’, and ‘adults.names’. For a more detail description of these files, consult **Appendix I**.

## Data Exploration

### **Purpose**

In recent years, the issue of economic inequality and the factors contributing to it has been in the spotlight. Racial, ethnic, gender and other types of discrimination have become a common topic of discussion among analysts worldwide. Experts are striving to highlight systematic biases present in society, the educational system, and the workspace, barring individuals from reaching their full potential. This is the lens under which we approached our analysis; one of the main points of focus was acquiring some intuition into which socio-economic parameters better explained the income group individuals belonged to.

### **Key Findings**

Assuming that in a fair economy the amount of pay someone receives should be linked to how much effort they put in their current job position and their professional and academic development, we start by inspecting the variables ‘hours-per-week’, ‘education’, and ‘occupation’.

We saw that those working more than the national standard were indeed among the high earners. For more observations on this subgroup, consult **Appendix II**.

A red and black rectangles

Description automatically generated

**Figure 28:** Income Subgroups per Weekly Workload Type

When it comes to education, we see that in the American society the most common group of adults was high school graduates.

A graph of different colored bars

Description automatically generated

**Figure 29:** Frequency of Education Levels among US Adults

As we expected, highly educated individuals were among the high earners in a more systematic way, with those holding a PhD or those having attended Professional School (preparing them to pursue careers in Law or Medicine) were high earners at approximately 75%, followed by those having a Masters and then a Bachelor’s degree.   
  
A graph of different colored bars

Description automatically generated with medium confidence

**Figure 30:** Participation of Education-based segments to Income groups

Among specific occupations, the most common ones where professors, craftspeople and business executives, with specialty professionals (like doctors and lawyers) and executives being high earners in a more systematic way than the rest.

A graph of different colored bars

Description automatically generated

**Figure 31:** Frequency of Occupations among US Adults  
  
A graph of red and black bars

Description automatically generated

**Figure 32:** Occupational Segment Breakdown by Income Group

Other than the previous professions, however, the private sector also contained some of the least lucrative occupations (like Handlers-cleaners), so the groups being more systematically among high earners were self employed people (typically entrepreneurs or external contractors to corporations), followed by the various levels of government.

A graph of red and black bars

Description automatically generated

**Figure 33:** ‘Workclass’ Segment Breakdown by Income Group

However, when we turned our attention to social factors, it was made clear from the training set that the probability for someone to be a high earner is tightly linked to parameters in and out of their control but irrelevant to their work efforts; sex, ethnicity, race, and marital status were among factors that led to significant inconsistencies in income group participation among various population subgroups, as analyzed in **appendix III.**

## Data Pre-Processing

### **Purpose**

We performed data pre-processing with the goal of improving the quality of our data and making it more suitable for the specific data mining task.

### **Description of Data Structures Used (separate X and y dataframes for the training and testing data)**

Holdout validation was already performed in the given data utilizing MLC++ GenCVFiles, and we checked that it was stratified.

**Table 10:** First 5 Rows of the 'X\_test' Dataframe

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **workclass** | **education** | **marital-status** | **occupation** | **relationship** | **race** | **sex** | **capital-gain** | **capital-loss** | **native-country** | **week\_type** | **age** | **workclass** |
| 25 | Private | high | Never-married | Machine-op-inspct | Own-child | Black | Male | 0 | 0 | United-States | national\_standard | 25 | Private |
| 38 | Private | HS-grad | Married-civ-spouse | Farming-fishing | Husband | White | Male | 0 | 0 | United-States | long\_week | 38 | Private |
| 28 | Local-gov | Assoc-acdm | Married-civ-spouse | Protective-serv | Husband | White | Male | 0 | 0 | United-States | national\_standard | 28 | Local-gov |
| 44 | Private | Some-college | Married-civ-spouse | Machine-op-inspct | Husband | Black | Male | 7688 | 0 | United-States | national\_standard | 44 | Private |
| 18 | NaN | Some-college | Never-married | NaN | Own-child | White | Female | 0 | 0 | United-States | national\_standard | 18 | NaN |

### **Pre-Processing on the Dependent Variable**

An inspection of the total values of ‘target’ showed that it is stored in string values; thus, label encoding (using sklearn’s “LabelEncoder” class) was necessary to make sure our models’ input was strictly numeric.

We also noticed significant class imbalances present in our dataset.

A blue and orange rectangular shapes

Description automatically generated

Ignoring them can introduce biases and significantly harm the generalization capabilities of our models. We dealt with them by implementing various measures, which are explained in Appendix IV.

## Pre-Processing on the Features Matrix

For more detailed explanations of the steps taken, consult appendix V

#### **Checking for Duplicate Values**

We dropped rows corresponding to the exact same individuals to prevent data redundancy and biases.

#### **Tracing Missing Values and Putting Down the Imputation Strategy**

We decided to implement random imputation so as to respect the distribution of the initially known values.

#### **Cleaning Categorical Columns**

Redundant space characters in string values were trimmed.

#### **Feature Engineering**

##### Feature Selection

We dropped the ‘fnlwgt’ and ‘education-num’ columns because we deemed their underlying information as unnecessary or even contrasting to our analysis’ needs.

#### Aggregating information

The information of the ‘education’ column was aggregated regarding individuals that have not graduated from high school, so as to make graph interpretations easier.

#### **Encoding Categorical Variables**

One-hot encoding was implemented, utilizing panda’s ‘get\_dummies’ method.

#### **Sampling Techniques**

We implemented the SMOTE method to eliminate imbalances in the training set. We chose this method because it synthesizes new values based on the KNN algorithm, preventing us from incurring the cost of random over-sampling and random under-sampling the training set bears, which could replicate biases or increase noise in our data.

#### **Feature Scaling**

Scikit-learn’s standard scaler was used to standardize all columns.

#### **Dimensionality Reduction**

Significant correlation was noticed between all features in the pre-processed training set, together with high sparsity. The method of Principal Component Analysis was chosen to eliminate high dimensionality.

A blue and orange dots

Description automatically generated

The first and second principal components do not do a great job separating between the two classes due to the relatively low proportion of the dataset’s variance that they manage to explain (approx. 11.50%), but it seems that in the high-dimensional plane our data might actually not divert much from the case of linear separability.

## Fitting & Tuning Predictive Models

### **Model Selection**

Apart from the required models (KNN, Random Forest, and Support Vector Machines). we also opted for training a logistic regression model and a voting ensemble. For the reasoning behind this selection, consult appendix IV.

### **Model Performance**

**Table 11:** Performance of Benchmark Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 (macro)** |
| Benchmark KNN | 0.815372 | 0.74 | 0.74 | 0.740394 |
| Benchmark RF | 0.815618 | 0.74 | 0.73 | 0.737196 |
| Benchmark Linear SVM | 0.841054 | 0.78 | 0.79 | 0.785139 |
| Benchmark RBF SVM | 0.830794 | 0.77 | 0.78 | 0.77282 |
| Benchmark Logistic Regression | 0.840686 | 0.78 | 0.79 | 0.783573 |
| Benchmark Voting Ensemble | 0.837368 | 0.77 | 0.78 | 0.777299 |

**Table 12:** Performance of Tuned Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 (macro)** |
| Optimal KNN | 0.813222 | 0.74 | 0.74 | 0.738556 |
| Optimal RF | 0.829196 | 0.76 | 0.76 | 0.762954 |
| Optimal Linear SVM | 0.839273 | 0.78 | 0.79 | 0.783698 |
| Optimal Non-Linear SVM | 0.830487 | 0.76 | 0.78 | 0.771549 |
| Optimal LogReg | 0.841054 | 0.78 | 0.79 | 0.78374 |
| Optimal Voting Ensemble | 0.838412 | 0.78 | 0.79 | 0.780691 |

**Note:** Slight drops in performance compared to the benchmark models are due to our computational resources, which did not allow for exhaustive search.

The leading performance of linear models (SVM and logistic regression) support our conjecture of linear separability.

The KNN and Random Forest models perform poorly in this setting, possibly due to the ‘curse of dimensionality’[[2]](#footnote-2) in our training set (Kouiroukidis & Evangelidis, 2011). These models had such shortfalls that the ensemble’s performance was lower than some of its base learners.

**Table 13:** Parameters of Tuned Models

|  |  |
| --- | --- |
| Model | Parameter Values |
| Optimal KNN | 'metric': 'manhattan',  'n\_neighbors': 10,  'weights': 'distance' |
| Optimal RF | 'n\_estimators': 40,  'min\_samples\_split': 5,  'min\_samples\_leaf': 1,  'max\_features': 'sqrt',  'max\_depth': 11,  'criterion': 'gini'} |
| Optimal Linear SVM | 'C': 0.01,  'class\_weight': 'balanced' |
| Optimal Non-Linear SVM | 'kernel': 'rbf',  'gamma': 'auto',  'class\_weight': None,  'C': 10 |
| Optimal LogReg | ‘C': 29.763514416313132,  'class\_weight': 'balanced',  'penalty': 'l1',  'solver': 'liblinear' |
| Optimal Voting Ensemble | 'weights': None,   'rf\_\_n\_estimators': 20, 'rf\_\_min\_samples\_split': 5, 'rf\_\_min\_samples\_leaf': 3, 'rf\_\_max\_depth': 49,  'rf\_\_criterion': 'entropy'  'rbf\_svm\_\_gamma': 0.01, 'rbf\_svm\_\_class\_weight': None, 'rbf\_svm\_\_C': 10,   'log\_reg\_\_solver': 'liblinear', 'log\_reg\_\_penalty': 'l2', 'log\_reg\_\_class\_weight': None, 'log\_reg\_\_C': 4.281332398719396,   'linear\_svm\_\_class\_weight': None, 'linear\_svm\_\_C': 0.1,   'knn\_\_weights': 'distance', 'knn\_\_n\_neighbors': 10 |

## Explainable AI

### **Purpose & Method Selection**

To uncover the biases in the American society which are implicit in the given dataset, we need to be able to estimate each feature’s marginal contribution toward the extraction of the signal[[3]](#footnote-3). We chose python’s ‘shap’ library to implement this analysis.

We identify the following features of the initial dataset as most contributing to someone being a low-earner: being, unmarried, female, not belonging to any family or being the child in one (and thus young).

Similarly for being a high earner: being married to someone in the armed forces or someone that lives far away, typically due to professional reasons (U.S. Census Bureau, n.d.), or having been married but widowed, increases an individual’s changes to become a high earner.

It should be noted that we faced various challenges in order to retrieve the aforementioned results. For a detailed overview of them and of method selection, as well as the calculations implemented to get the results above, consult **appendix 3**.

All in all, in the charts above we saw no variables linked to personal effort. Marital status, sex and relationships to other people may raise barriers or offer opportunities to people’s financial prospects in a more systematic way than professional and academic progress does.

## Conclusions & Future Work

So, there are clearly biases and systematic discrimination implied from the 1994 US Census data. However, there are state-level and race- and ethnicity- specific cultural and social norms that may influence an individual’s career choices and educational paths, as well as how much (in terms of hours spent) someone is eager to work. So there are various data that we could collect and include in our research so as to be able to isolate only external effects to someone’s income. Namely, these could be the state someone is based in, and some racial and ethnic controls for the aforementioned factors.

# REFERENCES

Asgard, R. (2014). Iranian Americans: Immigration and Assimilation. Public Affairs Alliance of Iranian Americans. <https://paaia.org/wp-content/uploads/2017/11/iranian-americans-immigration-and-assimilation.pdf>

Chen, X., Wang, Q., Liu, Z., & Han, Z. (2024). A novel approach for dimensionality reduction of high-dimensional stochastic dynamical systems using symbolic regression. *Mechanical Systems and Signal Processing*, *214*, 111373. <https://doi.org/10.1016/j.ymssp.2024.111373>

Darity, W., Guilkey, D. K., & Winfrey, W. (1996). Explaining Differences in Economic Performance among Racial and Ethnic Groups in the USA: The Data Examined. *The American Journal of Economics and Sociology, 55(4)*, 411–425. <http://www.jstor.org/stable/3487615>

Detective, T. D. (2020, January 31). *Finally: Why we use an 80/20 split for training and Test Data Plus an alternative method (oh yes...)*. Medium. https://towardsdatascience.com/finally-why-we-use-an-80-20-split-for-training-and-test-data-plus-an-alternative-method-oh-yes-edc77e96295d

Foryś, I. (2022). Machine learning in house price analysis: Regression Models versus neural networks. *Procedia Computer Science*, *207*, 435–445. https://doi.org/10.1016/j.procs.2022.09.078

Guyon, I. M. (1997). A Scaling Law for the Validation-Set Training-Set Size Ratio. *Mathematics, Computer Science*.

IBM. (n.d.). *What is Principal Component Analysis (PCA)?* https://www.ibm.com/topics/principal-component-analysis

Jurafsky, D., & Martin, J.H. (2023). Logistic regression. In Speech and Language Processing (Draft version, pp. 81-104). <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>

Kouiroukidis, Nikolaos & Evangelidis, Georgios. (2011). The Effects of Dimensionality Curse in High Dimensional kNN Search. Proceedings - 2011 Panhellenic Conference on Informatics, PCI 2011. 41 - 45. 10.1109/PCI.2011.45. <https://www.researchgate.net/publication/224265352_The_Effects_of_Dimensionality_Curse_in_High_Dimensional_kNN_Search>

Linkedin. (2023, March 30). *What are some challenges and limitations of cluster analysis using Davies-Bouldin index?* www.linkedin.com. <https://www.linkedin.com/advice/0/what-some-challenges-limitations-cluster-analysis>

Lu, L., Zeng, Z., Jiang, Y., Zhu, Y., & Hu, P. (2024). Weak collocation regression method: Fast reveal hidden stochastic dynamics from high-dimensional aggregate data. *Journal of Computational Physics*, *502*, 112799. <https://doi.org/10.1016/j.jcp.2024.112799>

Markovitch, T. [MarkovML] (pseudonym). (2023, October 26). LIME vs SHAP: A Comparative Analysis of Interpretability Tools. MarkovML. https://www.markovml.com/blog/lime-vs-shap

Newman, M. (2005). Power laws, pareto distributions and Zipf’s law. *Contemporary Physics*, *46*(5), 323–351. https://doi.org/10.1080/00107510500052444

NIST. (2024). *Histogram Interpretation*. 1.3.3.14.6. histogram interpretation: Skewed (non-normal) right. <https://www.itl.nist.gov/div898/handbook/eda/section3/histogr6.htm>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12\*\*(Oct), 2825-2830. <https://scikit-learn.org/>

Ribeiro, D. (2024, January 31). Optimal clustering - data and beyond - medium. *Medium*. <https://medium.com/data-and-beyond/optimal-clustering-a816e70b0ad3>

Seabold, P. & Perlman, J. (2010). statsmodels: Econometric and statistical modeling with Python. <http://www.statsmodels.org>

SHAP team. (n.d.). SHAP: A Unified Approach to Explain the Output of Any Machine Learning Model (version not explicitly mentioned on the retrieved page). Retrieved April 1, 2024, from <https://shap.readthedocs.io/en/latest/>

Sharma, N. (2024, March 17). *How to do One Hot Encoding? Transform Your Categorical Data!* Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2023/12/how-to-do-one-hot-encoding/>

Singh, S. (2023, July 19). *Title: Label Encoding and One-Hot Encoding for Data Preprocessing*. https://www.linkedin.com/pulse/title-label-encoding-one-hot-data-preprocessing-shivani-singh

Soyer, E., & Hogarth, R. M. (2012). The illusion of predictability: How regression statistics mislead experts. *International Journal of Forecasting*, *28*(3), 695–711. https://doi.org/10.1016/j.ijforecast.2012.02.002

Tuychiev, B. (2023, December 6). *Understanding skewness and kurtosis and how to plot them*. DataCamp. https://www.datacamp.com/tutorial/understanding-skewness-and-kurtosis

U.S. Census Bureau. (n.d.). Subject Definitions. Retrieved April 1, 2024, from https://www.census.gov/programs-surveys/cps/technical-documentation/subject-definitions.html

USAHello. (n.d.). What are the US education Levels?. <https://usahello.org/education/children/grade-levels/>

# APPENDIX

Regression

Regression Analysis

A regression analysis involves a collection of statistical processes aiming to define a mathematical model. The processes are performed on variables, that belong on the same descriptive paradigm or schema of a study. The purpose of the analysis determines which variable becomes the dependent, i.e. the one whose variance is to be explained by the resulting model by utilizing the remaining, the indipendent variables.

Appointing a dependent variable designates it as the outcome caused by the independent variables. The first challenge arises at the selection of the appropriate independent variables. The field of study, the sytem analyzed for causes and effects, establishes the level of complexity required by the predicting model. Many cases involve stochastic dynamics, meaning that the effects are stochastic in part, or that the complex dimensionality of their causes is too high to be captured and expressed by a single mathematical model (Lu et al., 2024).

The advent of artificial intelligence and its application on complex data fitting methods has demonstrated the potential to derive mathematical expressions from them, but these expressions do not completely model the relationship between causes and effects (Chen et al., 2024). The regression model may explain a proportion of the effects of independent variables on the changes of the dependent, however that does not affirm the model’s capability to make accurate predictions outside the scope of the training dataset (Soyer & Hogarth, 2012).

Data Exploration

Before any transformative process ensues on a dataset, especially one of unknown origin, there needs to be an inspection for peculiarities in the data set that would ascertain the validity of it as well as potentially provide insights to some underlying mechanisms of the system under study.

*Key Findings*

There were no duplicate samples in the dataset. However, there were over 7.800 missing values over 19 features. 43 out of the 79 features were categorical and 5 out of those were features that over 97% of samples have the same values.

|  |  |
| --- | --- |
| **Feature** | **% of Primary Value** |
| Street | 99.59% |
| Utilities | 99.93% |
| Condition2 | 98.97% |
| RoofMatl | 98.29% |
| Heating | 97.81% |

Most categories and some numerical features produced positively skewed histograms, probably due to the lower bounds on the data (NIST, 2024), and overall leptokurtic distributions, indicating greater likelihood of samples with extreme feature values (Tuychiev, 2023). This was reinforced further by the outlier detection, that resulted in a considerable number of outliers for most features. The independent variables were positively correlated with the sale price, with very few exceptions, for example ‘BsmtHalfBath’ and ‘EnclosedPorch’. For detailed results please consult the submitted Jupyter notebook.

Among other things, the feature-by-feature analysis revealed that there were some problematic values for certain features. For example, there were samples whose veneer area was 0 but their veneer type was not none. Also, there was a sample of a property without a kitchen, whose kitchen quality was set as ‘Typical/Average’. Such samples were dealt with on the following stage, except for the sample with the missing kitchen, which was dropped from the dataset.

Data Description

Numeric Features:

'LotFrontage', 'Alley', 'MasVnrType', 'MasVnrArea', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Electrical', 'FireplaceQu', 'GarageType', 'GarageYrBlt', 'GarageFinish', 'GarageQual', 'GarageCond', 'PoolQC', 'Fence', 'MiscFeature'

Categorical Features:

'MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functional', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual', 'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature', 'SaleType', 'SaleCondition'

Data Pre-processing

Categorical Encoding:

Alley : 'Pave':5, 'Grvl':3, 'No Alley': 0

BsmtQual : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

BsmtCond : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

BsmtExposure : 'Gd':4, 'Av':3, 'Mn':2, 'No':1

BsmtFinType1 : 'GLQ':6, 'ALQ':5, 'BLQ':4, 'Rec':3, 'LwQ':2, 'Unf':1

BsmtFinType2 : 'GLQ':6, 'ALQ':5, 'BLQ':4, 'Rec':3, 'LwQ':2, 'Unf':1

FireplaceQu : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

GarageFinish : 'Fin':5, 'RFn':3, 'Unf':1

GarageQual : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

GarageCond : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

PoolQC : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

Street : 'Grvl':5, 'Pave':3

Utilities : 'AllPub':4, 'NoSewr':3, 'NoSeWa':2, 'ELO':1

LandSlope : 'Gtl':3, 'Mod':2, 'Sev':1

ExterQual : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

ExterCond : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

HeatingQC : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

CentralAir : 'Y':1, 'N':0

KitchenQual : 'Ex':5, 'Gd':4, 'TA':3, 'Fa':2, 'Po':1

Functional : 'Typ':8, 'Min1':7, 'Min2':6, 'Mod':5, 'Maj1':4, 'Maj2':3, 'Sev':2, 'Sal':1

PavedDrive : 'Y':2, 'P':1, 'N':0

LotShape : 'Reg':3, 'IR1':2, 'IR2':1, 'IR3':0

LandContour : 'Lvl':3, 'Bnk':2, 'HLS':1, 'Low':0

BldgType : '1Fam':5, '2FmCon':4, '2fmCon':4, 'Duplx':3, 'Duplex':3, 'TwnhsE':2, 'Twnhs':1, 'TwnhsI':1

Foundation : 'PConc':6, 'Stone':5, 'BrkTil':4, 'CBlock':3, 'Wood':2, 'Slab':1

Heating : 'GasW':6, 'GasA':5, 'OthW':4, 'Wall':3, 'Grav':2, 'Floor':1

Electrical : 'SBrkr':5, 'FuseA':4, 'FuseF':3, 'FuseP':2, 'Mix':1

RoofMatl : 'ClyTile':7, 'Metal':6, 'WdShngl':5, 'WdShake':4, 'CompShg':3, 'Tar&Grv':2, 'Roll':1, 'Membran':0

MasVnrType : 'Stone':5, 'BrkFace':4, 'BrkCmn':3, 'CBlock':2, 'None':1

Fence : 'GdPrv':5, 'GdWo':4, 'MnPrv':3, 'MnWw':2, 'No Fence':1

## Clustering

Davies Bouldin Index

It is calculated as the average similarity measure of each cluster with the cluster most similar to it. In this context, similarity is defined as the ratio between inter-cluster and intra-cluster distances. As such, this index ranks well-separated clusters with less dispersion as having a better score.

### Calinski-Harabasz

Displays how similar an object is to its own cluster compared to other clusters (Cohesion). This cohesion is estimated based on the distances from the data points in a cluster to its cluster centroid, while separation is based on the distance of the cluster centroids from the global centroid.

### Brief explanation of algorithms and hyperparameters

KMeans is a centroid-based clustering algorithm that partitions the data into k clusters by minimizing the within-cluster variance. DBSCAN, on the other hand, is a density-based algorithm that groups together closely packed points based on a specified minimum number of points (minPts) within a specified radius (eps). Agglomerative Clustering is a hierarchical clustering algorithm that iteratively merges the closest pairs of clusters until only a single cluster remains.

### PCA

PCA is a powerful technique that reduces the number of features in the dataset while preserving most of the variance. By transforming the original features into a new set of orthogonal variables called principal components, PCA allows us to represent the data in a lower-dimensional space without losing significant information. This reduction in dimensionality not only improves the efficiency of clustering algorithms but also helps mitigate the curse of dimensionality, thereby leading to more accurate and robust clustering results.

### OHE

OHE expands each categorical feature into multiple binary features, with each representing a distinct category. This transformation ensures that the clustering algorithm can effectively capture the underlying patterns in the data.

## Classification

Appendix I

### **Brief Description of Data Files Used**

To briefly describe the data we worked on, we proceed to show a glimpse of the corresponding dataframe structures we created after reading those text files:

**adults.data:** each record corresponds to an individual in a certain state of the US, and 14 attributes are displayed, as shown in the following table. The ‘target’ column comprises our target variable. This file will be used train our models.

**Table 14:** First 5 rows of 'adult.data' in DataFrame format

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **workclass** | **fnlwgt** | **education** | **education-num** | **marital-status** | **occupation** | **relationship** | **race** | **sex** | **capital-gain** | **capital-loss** | **hours-per-week** | **native-country** | **target** |
| 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |
| 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| 28 | Private | 338409 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Wife | Black | Female | 0 | 0 | 40 | Cuba | <=50K |

**adults.test:** contains relevant information as ‘adults.data’, but it will be used in validating the predictive models we train.

**Table 15:** First 5 rows of 'adult.test' in DataFrame format

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **workclass** | **fnlwgt** | **education** | **education-num** | **marital-status** | **occupation** | **relationship** | **race** | **sex** | **capital-gain** | **capital-loss** | **hours-per-week** | **native-country** | **target** |
| 25 | Private | 226802 | 11th | 7 | Never-married | Machine-op-inspct | Own-child | Black | Male | 0 | 0 | 40 | United-States | <=50K. |
| 38 | Private | 89814 | HS-grad | 9 | Married-civ-spouse | Farming-fishing | Husband | White | Male | 0 | 0 | 50 | United-States | <=50K. |
| 28 | Local-gov | 336951 | Assoc-acdm | 12 | Married-civ-spouse | Protective-serv | Husband | White | Male | 0 | 0 | 40 | United-States | >50K. |
| 44 | Private | 160323 | Some-college | 10 | Married-civ-spouse | Machine-op-inspct | Husband | Black | Male | 7688 | 0 | 40 | United-States | >50K. |
| 18 | ? | 103497 | Some-college | 10 | Never-married | ? | Own-child | White | Female | 0 | 0 | 30 | United-States | <=50K. |

**adults.names:** contains a short documentation of the given dataset.

Appendix II

We found out that individuals working more hours than the national standard on a weekly basis were among the high earners relatively more often than the others, even though one could expect them to be people working harder because their lower social and academic standing makes it hard to make ends meet. We further looked at their case and found out that they only made for 12.58% of the training set’s samples, and were mainly occupied at very lucrative positions, most of them being business executives, professors, and salespeople. Furthermore, we believed those individuals would be singles rushing toward getting the most out of their careers while leaving their social lives behind, but we observed that married individuals were the most frequent among all marital status segments, even though singles followed closely.

A graph of different colored bars

Description automatically generated

**Figure 34:** Occupation Frequency among Long Week Workers

A pie chart with text and a couple of colored circles

Description automatically generated

**Figure 35:** Marital Status Relevance among Long Week Workers

Appendix III

We began our consideration of social factors’ influence on income upon finding out that men were not just more commonly included in the studied sample; they were also among high earners at about double the relevant rate of women.

A graph of a person and person

Description automatically generated

**Figure 36:** Frequency of the Sexes among US Adults  
  
  
A red and black rectangles

Description automatically generated

**Figure 37:** Sexes Segment Breakdown by Income Group

On the frontier of ethnicity, we were thrilled to see that Iranians made for the most financially successful population group (according to our training set) back in 1994, followed by French-born people. Following the 1979 Islamic Revolution, the influx of Iranians in the US made it clear that they were putting a lot of effort into their academic and professional development, leading them to impressive economic achievements (Asgard, 2014). As for the rest of the mentioned Asian groups, their achievements were already documented through MIT studies based on the results of the 1980 and 1990 US Census, and they are still some of the most successful ethnic groups in the country (Darity, Guilkey, & Winfrey, 1996).

|  |  |
| --- | --- |
| **Native Country** | **High Earners (%)** |
| Guatemala | 4.84% |
| Columbia | 3.39% |
| Dominican-Republic | 2.86% |
| Outlying-US | 0.00% |
| Holand-Netherlands | 0.00% |

**Table 16:** Top 5 High Earner Countries **Table 17:** Top 5 High Earner Countries

|  |  |
| --- | --- |
| **Native Country** | **High Earners (%)** |
| Iran | 41.86% |
| France | 41.38% |
| India | 40.00% |
| Taiwan | 39.22% |
| Japan | 38.71% |

The previous also paints the image of racial income imbalances in the country in 1994.

A red and black rectangles

Description automatically generated

**Figure 38:** Race Segment Breakdown by Income Group

Last, we observed that married (no matter the type of marriage) people where more likely to be high earners. This gives us a potential hint at the fact that the social and/or economic standing of someone’s spouse may influence their own.

**Table 18:** Percentage of High Earners per Marital Status Segment

|  |  |
| --- | --- |
| **Marital Status** | **Percentage of High Earners** |
| Divorced | 10.43% |
| Married-AF-spouse | 43.48% |
| Married-civ-spouse | 44.69% |
| Married-spouse-absent | 8.13% |
| Never-married | 4.60% |
| Separated | 6.44% |
| Widowed | 8.56% |

Appendix IV

The steps we took to make sure that class imbalances are fully addressed (according to our capacities) were the following:

* Stratified holdout validation: we have checked and ensured that the same class imbalance situation holds in the ‘target’ column of the training and test sets, to make sure that we train and test our models under the same imbalances which resemble our only estimate of the actual situation in the US in 1994
* Stratified k-fold cross-validation: the same imbalances as above need to be present in each iteration to achieve the previous
* To compare different proposed estimators, we considered the F1 score instead of accuracy. The F1 score is sensitive to both Type 1 and Type 2 prediction errors, giving a more balanced view of the model's generalization performance. Moreover, we considered macro average calculations of performance metrics, and not just the weighted averages, since applying frequency weightings to the results of the majority class would obscure those of the minority class. - However, we continued to fine-tune each model separately using accuracy as the point of reference, since ‘y\_test’ contains a given number of class A and class B cases, so the best model will also come with the best accuracy.
* Implement sampling techniques on our dataset so as to eliminate imbalances in the training set: A graph of a number of classes

  Description automatically generated with medium confidence
* When tuning specific classification models (SVMs, Logistic Regression, and Voting Ensemble), we also considered all legal values of the 'class\_weights' parameter in the parameters grid so that this built-in weighting facility offered by scikit-learn’s implementation is also considered. This factors in the ratio (i.e. relative frequency) of the class of each data sample so that it corrects to the particular imbalances in the dataset at hand.

Appendix V

#### **Checking for Duplicate Values**

We considered two rows as referring to the exact same socio-economic group of people in a given state (since the grouping was performed on a state-by-state basis by the dataset’s contributors) when all their features have the same values, simultaneously. In this case, they would introduce redundancy to our data, as a particular group in a particular state only exists once. Thus, we dropped duplicate rows to prevent introducing biases:

|  |  |  |
| --- | --- | --- |
|  | **training set** | **testing set** |
| **Initial shape** | (32561, 15) | (16281, 15) |
| **Final shape** | (32537, 15) | (16276, 15) |

#### **Tracing Missing Values and Putting Down the Imputation Strategy**

Missing values were denoted with ‘ ?’ characters, so we replaced those with NaN values and then found the following:

**Table 19:** Tracing Missing Values in the Training Set

|  |  |  |
| --- | --- | --- |
| Missing values per column of the training set: | | Percentage of Total Observations |
| workclass | 1,836 | 5.64% |
| occupation | 1,843 | 5.66% |
| native-country | 582 | 1.79% |

Similar was the situation in the testing set: A yellow rectangular object with black and white lines

Description automatically generated with medium confidence

**Figure 39:** Heatmap of Missing Values, Testing Set

All variables containing missing values are categorical. We observed that the distribution of occupation is rather diverse, so imputing its NaNs with its most frequent value would distort the relative distribution of labels, shifting away from the actual situation. Thus, we decided to implement random imputation, filling in missing values in each of those columns with random non-missing labels of the training set (respecting their distribution in it).

A graph of different colored bars

Description automatically generated

**Figure 40:** Distributions of Columns containing Missing Values

#### **Cleaning Categorical Columns**

All categorical variables' values have a redundant leading space character; we deleted it as they could cause problems in visualizations or value comparisons.

#### **Feature Engineering**

##### Feature Selection

We try to predict whether someone makes more than $50,000/year in the US by analyzing their socio-economic attributes, irrespective to the state they are based in. According to the dataset's documentation, 'fnlwgt' estimates are constructed based on the socio-economic characteristics of each sample against the population of their state, and only that of the whole country. States may have vastly different attributes, so these estimates do not reflect how likely an entry is to appear across the whole country, while we also lack any information on the state each entry refers to. As such, we dropped it.

Furthermore, the information given by 'education' and 'education-num' seems redundant, and the way the two variables influence the training of classification models is different. From common experience, we know that what matters most is the educational certifications that a person holds. As for people with less than a high-school diploma, just knowing the highest level of school education they received is enough.

So we dropped 'education-num' variable and kept the categorical 'education' variable that we would later on perform one-hot-encoding upon. This will help prevent introducing any bias to the model, as we saw in the training set before that people with higher academic qualifications were high earners at a higher rate than the rest.

#### Aggregating information

We performed the aforementioned operation on the ‘education’ column by extracting the cardinal number of the last class attended by relevant individuals, whom we then grouped together based on the following segmentation (USAHello, n.d.):

* Elementary school – attended up to the 5th grade of school
* Junior high school - up to the 8th grade
* High school - up to the 12th grade

#### **Encoding Categorical Variables**

We opted for one hot encoding, which had the cost of increasing the dimensionality and sparsity of our dataset. We took this decision because we wanted to avoid introducing any bias to our forecasts other than that which may be uncovered by studying the relations between variables in our features set, and which it is one of our goals to investigate.

#### **Feature Scaling**

The scales of our numeric features are significantly different, which can influence predictive models to weigh them differently. To prevent this, we utilized scikit-learn’s StandardScaler() class to make all features have a mean value of 0 and standard deviation equal to 1:

A diagram of a training set

Description automatically generated

**Figure 41:** Boxplot of selected Numeric Features after Standardization

From the boxplot above we also see that there are multiple outliers in the showcased variables; however, we will not drop them, as they do not resemble wrong values, just very rare cases that do exist in the population.

#### **Dimensionality Reduction**

To quantify the correlation between the pre-processed features, we used the VIF factor as a point of reference, which estimates the marginal contribution of a feature to the variance of estimates. Using the widely accepted threshold of a VIF factor of 5, we concluded that 98 out of 101 features are highly correlated.

Also, we found out that 98 out of 101 columns in the training features set that are significantly correlated, so implementing a dimensionality reduction technique here is necessary. Also, 68 out of the 100 columns are very sparse, having more than 95% of their values equal to zero. Thus, we implemented Principal Component Analysis – PCA to implement dimensionality reduction and eliminate multicollinearity.

Appendix VI

We chose the logistic regression model because it performs very well on binary classification problems in high-dimensional spaces with independent and scaled data (<https://web.stanford.edu/~jurafsky/slp3/5.pdf> ) (Jurafsky & Martin, 2023). Moreover, it is a white-box model, so it is easier to compute the importance it assigns to features. However, it is also susceptible to overfitting, so regularization techniques need to be implemented (L1 and L2 regularization). Additionally, it assumes that the values of the features set are linearly related to those of the target variable.

As for the voting ensemble, we chose it to show that even when working with models that have very poor predictive capabilities, simultaneously considering all of them in association may lead to better generalization. This comes by eliminating the threat of overfitting and better benefiting from the advantages of all while weakening their disadvantages, as the ensemble is protected from misclassifications of some learners by the correct predictions of others.

Appendix VII

We used python’s ‘shap’ library which offers an efficient implementation of the SHAP approach in explainable AI. We chose this method over LIME because it offers both local (i.e. prediction-by-prediction) and global explanations, and its implementation in python offers computationally efficient classes dedicated to the various trained models. Also, it is generally preferred when dealing with more complex models, like the SVM and Voting Ensemble we used in our analysis (MarkovML, 2023).

### **Challenges**

The problem we faced was that the fully pre-processed training set was scaled, and PCA was implemented on it. As a result, we could not link the values of our final features to the values of the initial dataset’s features.

We tried to work around this issue by pinpointing the 5 initial features contributing the most to the values of each of the most important features (PCs) of the fully-preprocessed training set.

### **Findings**

The global shap values for the optimal logistic regression model have as follows:

A graph with numbers and a bar

Description automatically generated

**Figure 42:** Mean Absolute SHAP Values per Feature

Each of these values shows how much (on average) did each contribute to the model's predictions when we include it in the features set compared to when we do not.

To show an overall summary of the local explanations we got from shap (i.e. how much did each value of each feature contribute to the prediction's value on average (when we include it vs we do not)), we also constructed a summary plot:

A screen shot of a graph

Description automatically generated

**Figure 43:** SHAP Summary Plot

To get a more detailed view of the two most important features:

A blue line graph with numbers

Description automatically generated

**Figure 44:** SHAP Scores for Specific PC1 Values

So higher values of PC1 contributed toward lower shap values, pertaining to more negative impact to the predictions’ values (on average). Below we show the 5 features of the initial dataset that contributed the most to the values of PC1, together with the percentage of their values that influenced PC1 positively and negatively.

A screenshot of a graph

Description automatically generated

**Figure 45:** 5 Initial Features Contributing most to PC1's Values

We see that being an unmarried female that does not belong to any family or that is the child of a family (possibly pertaining to lower ages) in mid-1990s USA contributing to their being a low earner.

Now, for PC2, we see that high values are linked to positive influence on the predicted values, whereas low values are linked to significant negative influence to them:

A blue line with dots

Description automatically generated

**Figure 46:** SHAP Scores for Specific PC2 Values

Similarly as before we see that being married to someone in the armed forces or someone that lives far away, typically due to professional reasons (U.S. Census Bureau, n.d.), or having been married but widowed, increases an individual’s changes to become a high earner. On the other hand, being separated may lead to the opposite.

A red and black striped graph

Description automatically generated

**Figure 47:** 5 Initial Features Contributing most to PC2's Values

1. <https://archive.ics.uci.edu/dataset/2/adult> [↑](#footnote-ref-1)
2. When working with scaled data on which we have also implemented PCA, differences between features become extremely small, so the model finds it difficult to make clear distinctions. [↑](#footnote-ref-2)
3. For example, if two individuals have the same educational background, are working the same amount each week, and have the same occupation in the same country, then their income should not be significantly important. If it is, and we find that variables other than those linked with the aforementioned attributes are the most contributing to the predictions a model makes, it could be due to the existence of biases toward ethnic, racial or other types of minorities. [↑](#footnote-ref-3)