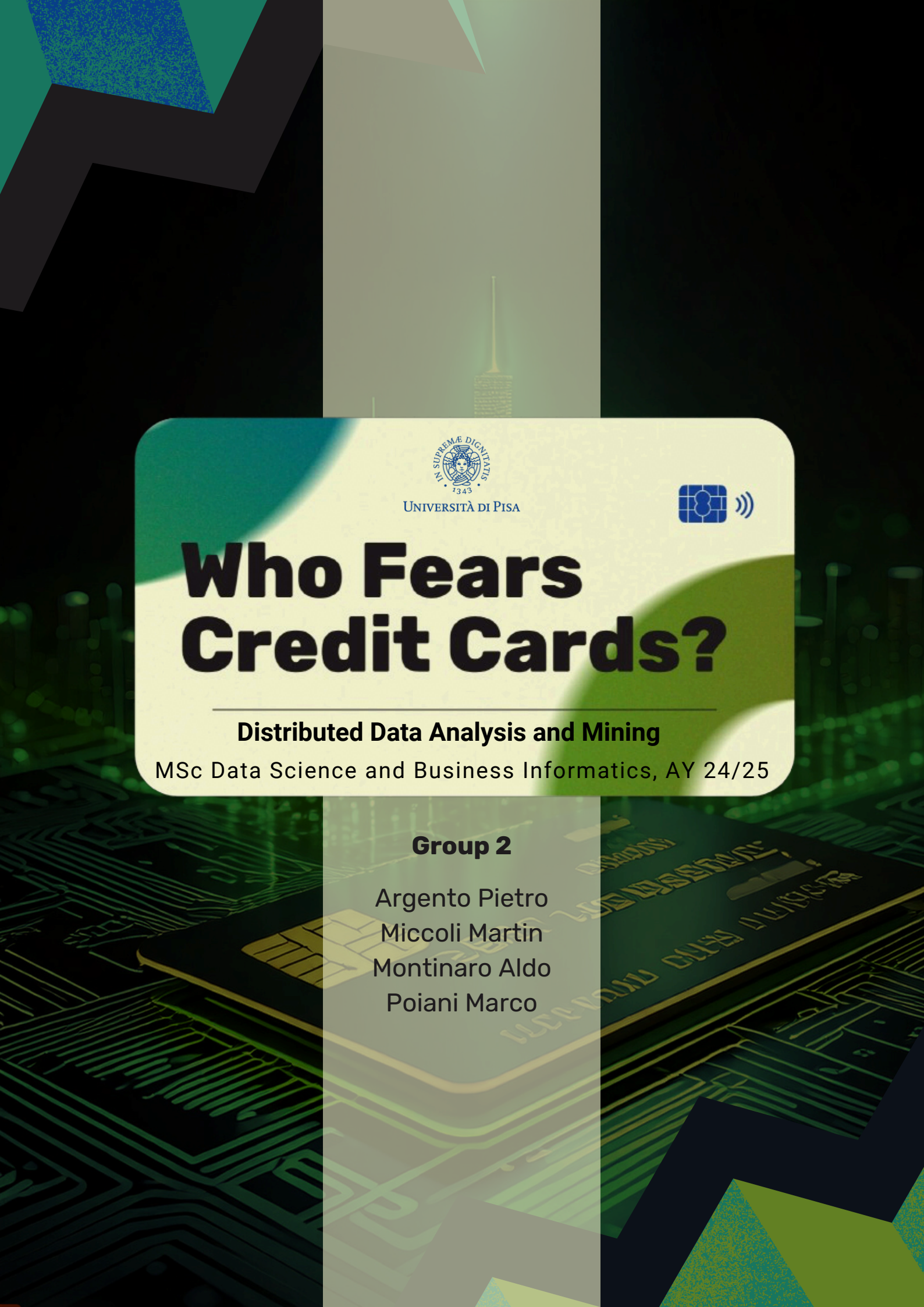


Università di Pisa

Who Fears Credit Cards?

Distributed Data Analysis and Mining

MSc Data Science and Business Informatics, AY 24/25

Group 2

Argento Pietro
Miccoli Martin
Montinaro Aldo
Poiani Marco

Introduction

The **Credit Card Transactions Dataset** that we used in our project contains **1296675** rows, and each one represents a *transaction*. Given that information about credit card transactions and cardholders is strongly protected for reasons concerning the privacy and safety of the users, this dataset is made up of **synthetic instances**. The dataset is generated by an algorithm called **Sparkov Data Generator** available on GitHub which aims at providing data pretty similar to the real ones. Another interesting information about the dataset is that each credit card has a relation **one-to-one with a cardholder** that in total are **983**. The initial dataset presents the **28 features** reported in the table below, and each of them is related either to a transaction or to the merchant or the cardholder.

Group	Feature	Description	Example	Type
Transaction	trans_num	Unique identifier for each transaction.	0b2a0b823cd6578578680df3065185b9	string
	trans_date_trans_time	Timestamp indicating when the transaction occurred.	01/01/19 00:00	timestamp
	unix_time	Unix timestamp representing the transaction time.	1546300818	int
	lat	Latitude coordinates of the transaction location.	300.853	double
	long	Longitude coordinates of the transaction location.	-817.818	double
	city_pop	Population of the city where the transaction occurred.	3458	int
	cc_num	Hashed or anonymized credit card number associated with the transaction.	2,03182E+14	bigint
	amt	Total amount of money involved in the transaction.	4.97	double
	category	Type or category of the transaction (e.g.	grocery	string
	is_fraud	Indicator (flag) showing whether the transaction is suspected or confirmed as fraudulent.	0	int
Cardholder	first	First name of the cardholder.	Jennifer	string
	last	Last name of the cardholder.	Boris	string
	gender	Gender of the cardholder.	F	string
	dob	Date of birth of the cardholder.	09/03/88	string
	street	Street address of the cardholder.	561 Perry Cove	string
	city	City of the cardholder's address.	Maynard Falls	string
	state	State or region of the cardholder's address.	NC	string
	zip	Zip or postal code of the cardholder's address.	28654	int
	job	Occupation of the cardholder.	Psychologist	string
Merchant	merchant	Name or identifier of the merchant or store where the transaction took place.	fraud_Rippin, Kub and Mann	string
	merch_lat	Latitude coordinates of the merchant's location.	300.123	double
	merch_long	Longitude coordinates of the merchant's location.	-6.248.135	double
	merch_zipcode	Zip or postal code of the merchant's location.	28705	int

Contents

1 Exploratory Data Analysis **1**
- 1.1 Data Distributions and correlations . . . 1
- 1.2 Geospatial Understanding . . . 2

2 Clustering **3**
- 2.1 Customers Segmentation . . . 3
  - 2.1.1 K-Means . . . 3
- 2.2 Transactions Clustering . . . 4
- 2.3 Geospatial Analysis . . . 5

3 Behavioural Analysis **7**
- 3.1 Customer Dimension . . . 7
- 3.2 Time Dimension . . . 7
- 3.3 Geospatial-based behavioural analysis . . . 8

4 Classification **9**
- 4.1 Classification with clustering information . . . 10
- 4.2 Main Classification Task . . . 10

1 Exploratory Data Analysis

In this phase, we analyzed our dataset to determine how to work with it and how to handle the features, given that they are both continuous and categorical. Among the categorical features, some have very high cardinality. The dataset does not contain duplicated rows, and only the feature merch_zipcode reports some missing values—more precisely, 15.1% of the values were missing. Table 1.1 shows some key statistics for numerical variables:

Summary	**amt**	**city_pop**	**is_fraud**
Count	1296675	1296675	1296675
Mean	70.3510	88824.44	0.0058
Std	160.3160	301956.36	0.0759
Min	1.0	23	0
Max	28948.9	2906700	1

Table 1.1: Summary statistics for amt, city_pop, and is_fraud

The **target variable** `is_fraud` is highly imbalanced, with only 7506 fraudulent transactions, representing approximately 0.58% of the total. Addressing this imbalance will be a priority in later stages.

1.1 Data Distributions and correlations

Figure 1.1 illustrates the distribution of transaction amounts. The distribution is positively skewed, with most transactions occurring below $500, while the highest amount for a single transaction is $28000 in correspondence of a purchase in the category travel

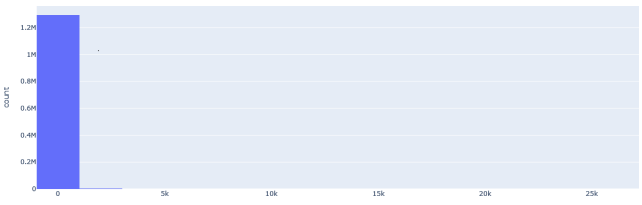


Figure 1.1: Histogram of Transaction Amounts

Successively, to investigate the relation among features we measure the correlation and plot it using a heatmap. The correlation matrix 1.2 reveals a strong positive correlation of 0.98 between merch_zipcode and zip, suggesting these variables may represent similar or overlapping geographic information. The target variable "is_fraud" shows weak correlations with all the variables, with a peak of 0.22 with the feature "amt". This indicates that further exploration and feature engineering may be needed, especially for classification tasks. After visualizing correlations among features, we dropped the variable "Unnamed: 0" which was an index and unix_time which was useless and even incorrect, because by inspecting the value there was an offset of seven years between the date in which the transaction occurred and the Unix time. In parallel, we added some external features, like the average household income by state in 2019 and 2020 (data have been retrieved from the official site of FRED). Finally, we used the Python library `holidays` to determine, based on the date of the transaction, if a transaction happened during the holidays or not. Additional features have been created, removed and processed during the experiments and we explained the reasons in the context in which changes occurred.

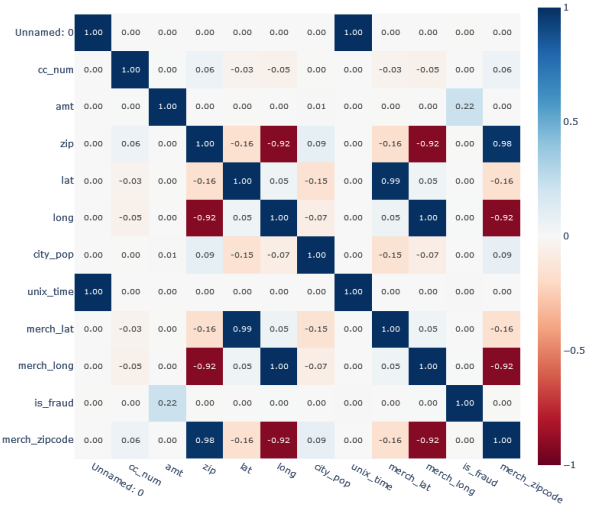


Figure 1.2: Heatmap of Correlations Among Numerical Variables

1.2 Geospatial Understanding

The map below 1.3 presents a geographic visualization of a sample of the transactions in the dataset, highlighting their distribution across the US. The blue dots represent legitimate transactions, while the red dots indicate fraudulent purchases. A concentration of transactions is observed in densely populated urban areas, such as the East Coast and California. Fraudulent transactions appear to be sparsely distributed but show peaks in metropolitan regions. This distribution suggests a potential relationship between demographic density and transactional activity, in addition to a higher presence of fraudulent transactions in the eastern part of the US.

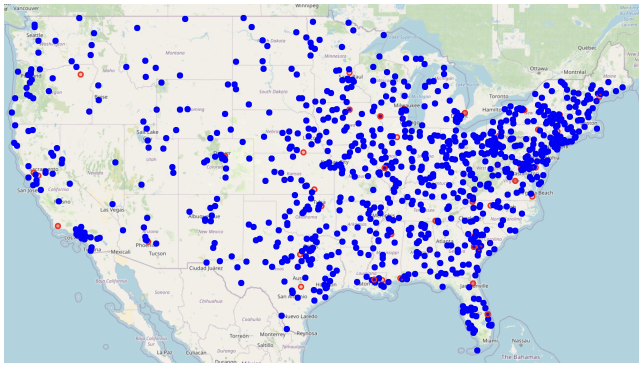


Figure 1.3: Distribution Map - Transactions

2 Clustering

During the project, we employed various clustering techniques, each focusing on a distinct level of granularity within the dataset. Initially, clustering was used for customer segmentation to identify distinct customer groups. Next, we applied clustering to the entire dataset to identify meaningful clusters of transactions, and finally, we leveraged clustering primarily on geospatial data to uncover location-based patterns.

2.1 Customers Segmentation

We began by engineering additional features to enhance the model's ability to generate meaningful clusters. To achieve this, we grouped the data by card number, which serves as an identifier for cardholders. Since there were only **983 unique customers**, this significantly reduced the dataset's size. We added the following features: the number of transactions made during holidays, the maximum spending for a single transaction by each cardholder, the number of online transactions, and the count of transactions exceeding the 95th percentile of the transaction amount calculated across the entire dataset. Additionally, recognizing the importance of demographic features in customer segmentation, we used the categorical feature `job`, which had high cardinality, to extract education-related information. Using **DistilBERT**, we generated a binary feature, `Education`, with values "High Education" and "Low Education" to represent the educational level. After completing the preparation phase, we evaluated the correlation between features and examined the presence of outliers. These steps were critical, as high correlations and the presence of outliers could negatively impact the quality of our clusters. The most correlated features were those regarding how much each cardholder spent in every category over the entire period, variables that we created to understand the spending habits of the cardholders. We removed these features and curiously noticed a strong positive correlation between the number of online transactions and the number of transactions during holidays. Next, we used Isolation Forest to detect outliers, identifying 10% of the dataset (99 instances) as potential anomalies. We selected this algorithm due to its reliability; however, upon inspecting the flagged instances, we found no significant statistical anomalies. The mean score for inliers was 0.45, compared to 0.57 for outliers, with a maximum outlier score of 0.77. Given that removing cardholders could compromise the integrity of our analysis, we decided to retain all instances in the dataset and eventually remove them if there were issues with the clustering.

2.1.1 K-Means

At this point, we decided to proceed with K-Means for performing the segmentation. We had demographic and behavioural features for each cardholder: **Demographic**: age, gender, state, zip, lat, long, the population of the city, education and Income 2019; **Behavioural**: average number of daily transactions, the average amount spent per day, number of transactions above the 0.95 percentile, the highest amount spent, number of online transactions and number of transactions during the holidays. Two crucial parts of the K-Means algorithm are both the selection of the right K and the initialization which is not deterministic. By exploiting the SSE and the Elbow Method we chose 6 as the best number of clusters, a choice confirmed also by the Silhouette method. We initialized the algorithm several times and at each iteration, we checked the contained instances and 6 revealed itself as a good number of clusters.

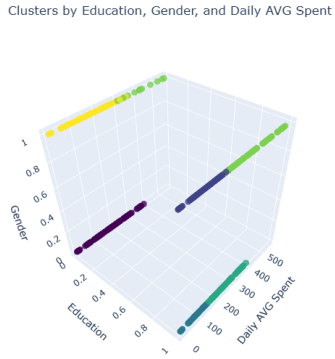


Figure 2.1: 6 clusters by Education, Gender, and Daily Average Spent.

Successively, by doing some plots, we discovered that three significant features used by the algorithm for performing the clusters were: *gender*, *education* and *daily average spent*, as depicted in figure: 2.1. Our concerns regarded the fact that our best clustering came up using a feature derived by a transformer, so we performed again the clustering following the same path but this time we excluded the artificial variable "education", as a result, we saw that the clusters were quite similar and a set of features that played a pivotal role were those related to the geographical information. Finally, we tested if also by using the dataset with the whole features, those already included plus those removed because of the high correlation, the clusters were still significant, also this time the results were clear and the groups well-separated. This means that fewer features were enough to capture the similarities relevant to this task, but at the same time, the correlation among features wasn't a relevant problem. To conclude this part about customer segmentation, we exploited the clustering with the education dimension. Every customer belongs to one of the 6 clusters, that given the composition have been renamed:

- **Low-educated men** (cluster 0): all the components are men that have in common a lower level of education. (171 instances)
- **Low-spending Highly-educated women** (cluster 1): all components are low-spending women with a higher education. (211 instances)
- **Low-spending Highly-educated men** (cluster 2): all components are low-spending men with a higher education. (171 instances)
- **High-spending and High-educated men** (cluster 3): all components are high-spending men highly educated. (144 instances)
- **High-spending women** (cluster 4): all components are high-spending women and education doesn't count. (147 instances)
- **Low-educated and low-spending women** (cluster 5): all components are low-educated and low-spending women. (139 instances)

2.2 Transactions Clustering

During this analysis, we employed K-Means to segment transactions, but also this time before applying the model we did a significant pre-processing phase. We started by exploiting Isolation Forest to detect anomalies in the data, ensuring that outliers were handled separately to improve the quality of the clustering results. The stages to do so are the following:

- The `features` vector was used as input for the Isolation Forest algorithm.
- Points exceeding a predefined threshold were flagged as anomalies and excluded from the subsequent clustering analysis.

As the second step, we reduced the number of features, leaving just the most informative for our goal: Transaction amount (`amt`), Customer age (`age`), Daily transaction count (`daily_tran`) and Average daily spending (`daily_avg_spent`). Once the anomalies were removed, the optimal number of clusters (K) was determined using two metrics:

- **SSE:** The sum of squared errors was computed for values of k ranging from 2 to 10.
- **Silhouette Score:** The silhouette coefficient was computed for each k, providing an additional perspective to confirm the optimal choice.

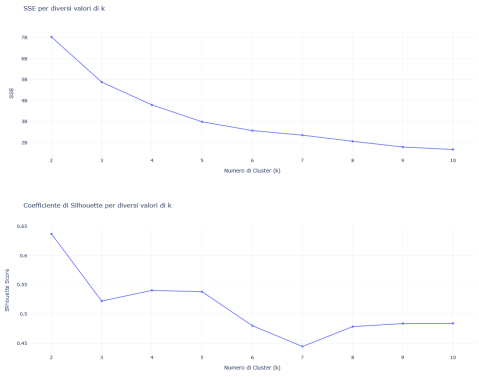


Figure 2.2: SSE and Silhouette score plots (elbow method)

After inspecting the plot we chose 5 as the optimal value for k. After several trials of understanding the cluster both visually and statistically considering only features related to the transactions we introduced also some features about customers, otherwise, the clusters appeared meaningless. This may indicate that clusters based solely on the transactions 2.3 don't provide significant results.

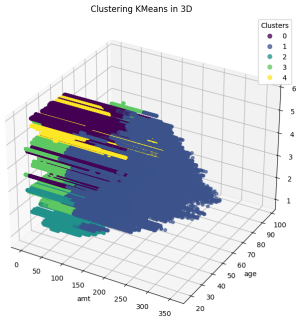


Figure 2.3: Cluster - amt, age, and daily tran

2.3 Geospatial Analysis

Since the dataset contains information related to transaction coordinates, we decided to develop an analysis based on the geographic characteristics of the records. We used the file with training data, containing 1296675 records related to credit card transactions, which we processed so that it could be used for the specific geospatial analysis: we dropped unnecessary columns like `first` and `last`, `year` or `job` in order to work only with useful spatial features like `merch_lat` (and user latitude) and `merch_long` (and user longitude) and some other interesting features like `age`. We then created an "online" binary column, containing 0 for offline transactions and 1 for online; we relied on the `category` feature in which some categories contain the suffix "_net" and can thus be associated with online payments. This choice is dictated by the fact that in order to carry out spatial analyses we need to focus on physical movements of the user. From here on, we will use only the filtered dataset containing 1090393 offline transactions for analysis.
First, we added, for each transaction, the `distance_from_home` (for home we took into account lat and long of the user) by calculating it using Haversine's formula (geodesic distance). We then printed the distribution of distances from home to choose a distance threshold above which to consider transactions as "not usual"; based on the plot depicted below, we filtered out transactions with distances greater than 80 miles, obtaining a 520307 records. To prepare the data for clustering, we converted the categorical variables ('merchant', 'category' and 'cc_num') into numerical representation with StringIndexer and then combined all features into a feature vector with VectorAssembler. To this we apply a StandardScaler to normalize its values. In order to use KMeans in an optimized way, we tested different values of k (from 2 to 11) by printing the corresponding SSE, so that we could choose the optimal k value through the elbow method: in our

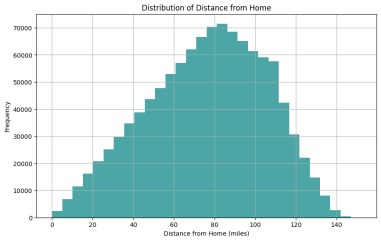
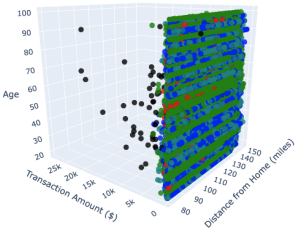


Figure 2.4: Graph of the distribution of the derived column `distance_from_home`.

case it results in $k = 4$. As an evaluation metric, we used the Silhouette Score, which was found to be 0.5: thus, we can assume that the clusters are approximately separated but not perfectly, probably caused by the presence of outliers.



In the figure we have highlighted the fraudulent transactions in red. We can initially see a clear separation between transactions with very high amounts (black cluster) and all other records; we can see that in general fraudulent transactions are still part of groups of transactions with small amounts. High transactions themselves are not distinguished by age or distance from home, which is also true for low amount transactions. We can therefore conclude that distance from home alone may not be a sufficient flag for classification.
For a more in-depth geographic study, we isolated only the coordinates as input for the kmeans ('lat', 'long', 'merch_lat', 'merch_long', 'prev_lat', 'prev_long', 'distance_from_home'). Again we started from the elbow method and identified as k optimal again $k = 4$.

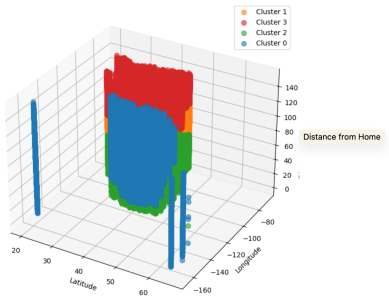
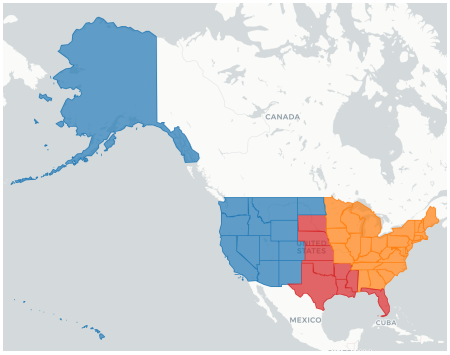


Figure 2.5: Spatial clustering results

With a Silhouette score of 0.47 we identified the 4 clusters in the Fig. 2.5: we noticed from the Latitude and Longitude dimensions that there are some transactions belonging to cluster 0 that occur in very different locations from the rest of the points (by observing the map we noted that they probably refer to separate U.S. states such as Alaska). Looking at the rest of the points we notice that the clusters group transactions with low distance from home (green), medium distance from home (orange), and other geolocated transactions farther away (red); the blue cluster invests all three bands.
We then carried on the analysis to assign clusters of points to U.S. states based on their geographic locations taken from the official shapefile (link). The analysis begins by loading the boundaries of U.S. states which serves as the foundational layer for the spatial operations. The dataset of clustered points is converted into a GeoDataFrame where each point is defined by its latitude and longitude coordinates, and the dataset is projected into the same coordinate reference system (CRS) as the U.S. states shapefile (`EPSG:4326`). The key spatial operation performed is a **spatial join** that associates each point with the state it falls within; once the points are spatially linked to states, the analysis aggregates the number of points belonging to each cluster within each state. This information is further processed to identify the dominant cluster in each state, defined as the cluster with the highest count of points.



We defined a color mapping for the states based on the majority cluster in each state, represented in a `folium` map. The states are added to the map with their geometries filled with colors corresponding to the dominant cluster, and the map is centered on the average latitude and longitude of all points.
In the EDA phase, we saw that most fraudulent transactions are recorded in the eastern states, and from this visualization we note that the same group of states contains mid-range (orange) transactions; we might therefore think of it as a reconfirmation that fraudulent transactions involve mid-range amounts at medium distances from home.

3 Behavioural Analysis

3.1 Customer Dimension

For the first analysis about the spending behaviour we exploited the clusters derived in the previous section. During these analyses we used again the entire dataset with 1296675 instances but since each transaction has a customer, we assigned the cluster to which the cardholder belongs to the transaction. At this point by aggregating data we calculated the percentage spending per category by cluster. Our idea was to discover the most and the less profitable segment for each category.

Figure 3.1: Percentage spending per category by cluster

Figure 3.1 clearly illustrates that certain clusters of people made more purchases in specific categories compared to other clusters. This insight can be strategically valuable for companies in the field. For example, we observe that, in general, a higher percentage of total spending by men is allocated to the "travel" category, while women tend to spend a larger percentage of their total expenditure on the "grocery_pos"category. An extended version of this graph, shows that another interesting category is the "health_fitness", where the percentage of the total spent of each cluster is the same except for the cluster High-spend and High-educated men. Secondly, since our clusters didn't capture well the age dimension, we created three group of people based on the age, "Under 30", "30-60" and "Over 60", because also this dimension could lead some additional information about our customers. Indeed, as we can see in picture 3.2 every segment has a different spending behaviour, for instance the segment "Under 30" is the one that allocates a higher percentage of the total amount to the "entertainment" category and also the one that has a lower percentage of spent for "grocery_pos".

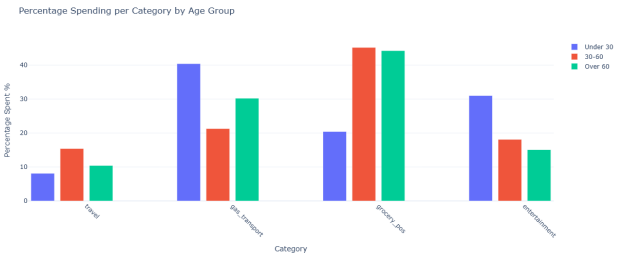


Figure 3.2: Percentage spending per category by age group

3.2 Time Dimension

After analyzing the customer data, we shifted our focus to the time dimension to determine if certain categories experience higher sales at specific times of the day. In Figure 3.3, we show the number of transactions involving the 'Personal Care' category by hour. The chart reveals that the peak occurs at 7 p.m., while the lowest point is at 2 p.m. This type of analysis could be valuable for businesses looking to optimize product promotions.

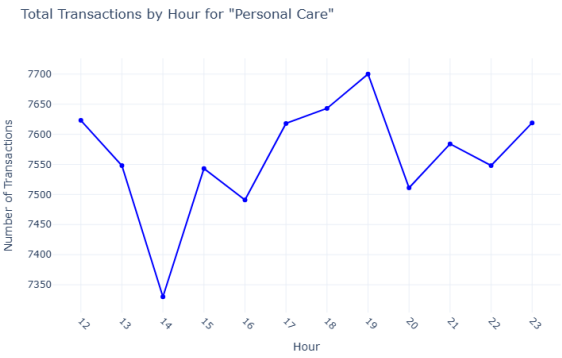
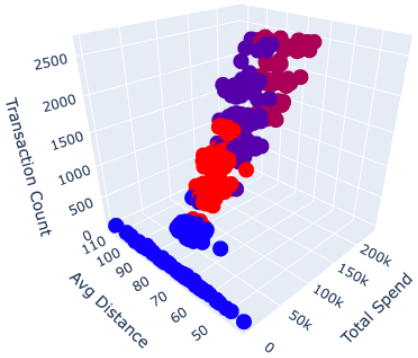


Figure 3.3: Total transactions by hour for "Personal Care"

3.3 Geospatial-based behavioural analysis

The primary goal of this part of the study was to identify distinct user segments based on purchasing habits and geographical information of transactions. Through the use of KMeans (with $k = 4$ identified by the Elbow method) each user was assigned to one of the 4 clusters with silhouette scores of 0.718 (indicating a good degree of cohesion and separation among the clusters).



For the same average distance from home, we noticed different ranges of user behavior. In red a first band with number of transactions and relative amount in the average range; at the top the purple and burgundy clusters instead indicate behavior bands with high spending and high number of transactions, with the purple band containing more variety by also investing part of the red cluster. The main insight we can see is that the users just described, associated with medium-high number and medium-high amount all refer to the same range of average distance; instead we note that for sporadic transactions (low count) and very low spending the average distance presents a much more varied distribution.
At this point we have included total_spend, avg_spend, transaction_count, avg_distance, and avg_hour in a VectorAssembler so that they can then be scaled with `MinMaxScaler`. We then grouped the data by cluster label and averaged the scaled features. In order to better visualize and compare the resulting clusters, we used radar charts that contained the scaled features, the result is in Fig. 3.4:

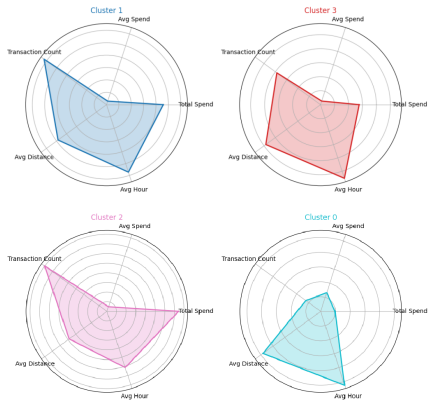


Figure 3.4: Radar plot of customer segmentation.

- Cluster 0: Occasional Travelers: includes users who have moderate total and average spending combined with a moderate number of transactions. Their transactions often occur at locations farther from home, indicating occasional travel or non-local spending patterns. These cardholders might represent individuals who use their credit cards for travel-related expenses or sporadic high-value purchases during trips.
- Cluster 1: Frequent Small Spenders: characterized by users who make a high number of transactions but with relatively low average and total spending. Their transactions tend to occur at nearby locations, as indicated by their moderate average distance from home. These cardholders might represent regular local shoppers or individuals frequently engaging in small-value purchases. Their activity pattern suggests high engagement with their credit cards for everyday spending.
- Cluster 2: Local Low-Activity Users: these users exhibit low total and average spending, along with a low number of transactions. Their transactions occur very close to home, suggesting a localized spending pattern. These cardholders might represent individuals who use their credit cards sparingly, perhaps for essential or occasional purchases rather than frequent or diverse spending.
- Cluster 3: High-Value, Moderate Activity: consists of users who have higher total spending with a moderate number of transactions. Their spending is spread across locations at a moderate average distance from home. They may represent affluent cardholders who make fewer, but higher-value transactions. They focus on quality over quantity in their spending habits.

4 Classification

In this section, we explain the classification phase in detail. In the following table, we report all the features that we used for training our models. We retained all the features that could help our models classify the transactions, therefore we included features concerning the merchant, the cardholder and the transactions.

amt	gender	lat
long	city_pop	merch_lat
merch_long	Income_2019	Income_2020
age	year	month
day	hour	minute
second	is_holiday	daily_tran
daily_avg_spent	entertainment	food_dining
gas_transport	grocery_net	grocery_pos
health_fitness	home	kids_pets
misc_net	misc_pos	personal_care
shopping_net	shopping_pos	travel
online	cat_enc	st_enc

Figure 4.1: Features selected for the classification task.

The features from "entertainment" up to "travel" concern the total amount spent in that category by the cardholder, while "cat_enc" and "st_enc" respectively stand for category encoded and state encoded. We encoded them to allow our models to exploit this information that could be crucial. To do so we used the **frequency encoding** technique, a suitable way for transforming categorical features with high cardinality into numerical ones, without increasing the dimension of the dataset as one-hot encoding would do. To the original strategy, we added a smoothing parameter "alpha" to mitigate the impact that more represented classes could have. We avoided one-hot encoding because it would add 62 features and despite the sparsity, they might compromise the ability of the model.

We saw in the EDA phase that the dataset used for this study is highly unbalanced: 99.4% of the transactions belonged to the non-fraudulent class, while only 0.6% were fraudulent. In order to train the different types of models, which we will see later, we had to find appropriate solutions to resolve the imbalance so that the minority class (i.e., is_fraud=1) emerges since the main interest of the task is the correct detection of fraudulent transactions. To address this, we applied various oversampling and undersampling techniques. Since most libraries we found were not compatible with PySpark DataFrames, we implemented custom functions. The simplest approach involved oversampling the minority class (fraudulent transactions) with replacement to achieve the desired class ratio while leaving the majority class untouched. Next, we implemented a function that applied SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic samples for the minority class. Models trained on the SMOTE-balanced dataset outperformed those trained on the dataset-balanced using the simpler oversampling method. Lastly, we compared the results of our PySpark-based approaches with models trained on datasets converted to Pandas and balanced using ADASYN and SMOTE, leveraging libraries compatible with Pandas. While we hypothesized that these libraries might yield better results, the best-performing model was trained using the SMOTE function implemented in PySpark.
In order to properly evaluate (and compare) the behavior of the models with respect to the minority class, we decided not to dwell on the use of accuracy but evaluated multiple metrics across the board, specifically:

- `Recall` (sensitivity): in this context missing positive instances cost is high so we want to maximize its value.

- `G-measure`: geometric mean between precision and recall which better captures the behaviour of the model on the minority class.

- `Jaccard Index`: measures the similarity between the predicted and actual sets dividing the size of their intersection by the size of their union; penalizes both FP and FN encouraging the model to minimize both.

We then created a function capable of returning all the metrics based on the predictions of the model used. First, we defined a VectorAssembler to assemble the input features and a MinMaxScaler to apply to them. They will be used in each training pipeline before the chosen model.

4.1 Classification with clustering information

In this section, we present a classification experiment conducted using the dataset employed for clustering. This dataset includes information about the cluster to which each transaction belongs. To prepare for classification, we joined the dataset containing all transactions with the dataset containing customer information. Since each transaction is associated with a specific cardholder, this allowed us to assign a cluster label to every transaction. Before training the classification models, we addressed class imbalance by oversampling the minority class. Using the oversampling function discussed earlier, we created two less imbalanced datasets: one with a minority-to-majority class ratio of 1:5 (1 minority instance for every 5 majority instances) and another with a ratio of 1:7. We then trained two classification models: logistic regression and random forest. Hyperparameter optimization was performed using random search combined with cross-fold validation. Among these models, the **Random Forest** trained on the dataset with a **1:7 oversampling ratio** achieved the best performance on the validation set.

- Precision: 0.929 | Recall: 0.775
- F1-score: 0.845 | G-measure: 0.849
- Jaccard Index: 0.732
- ROC-AUC: 0.992

Following this experiment, we returned to analyze the original dataset introduced earlier in the study.

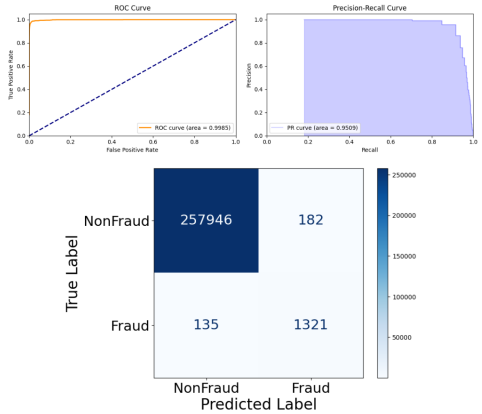
4.2 Main Classification Task

We started with a dummy classifier applied on the unbalanced dataset to evaluate the baseline performance: we obtained an accuracy of 99.4% (which is equal to the presence of the majority class); of course, the other metrics evaluated perform poorly with jaccard and g-measure at 0 and ROC-AUC at 0.5. We then trained a decision tree on the same data to understand its behavior compared to the dummy classifier: performance is slightly better but with big room for improvement. At this point we used a hybrid technique with Random Undersampling and ADASYN to be able to undersample (50:1) the majority class and then subsequent oversampling (based on point density) the minority class: thus we get 302894 non-fraud and 304161 fraud. The dummy classifier results dropped with accuracy of 0.005 and same ROC-AUC of 0.5. The models tested on the balanced dataset are (in order): Decision Tree, Logistic Regressor, Random Forest, XGBoost.

Model	Precision	Recall	F1 Score	Accuracy	G-Measure	Jaccard Index	ROC AUC
Dummy Classifier (no balancing)	0.0	0.0	0.0	0.994	0.0	0.0	0.5
Decision Tree Classifier (no balancing)	0.818	0.63	0.712	0.997	0.718	0.553	0.849
Dummy Classifier	0.006	1.0	0.011	0.006	0.075	0.006	0.5
Decision Tree Classifier	0.065	0.889	0.121	0.927	0.24	0.064	0.818
Logistic Regression	0.041	0.791	0.077	0.894	0.179	0.04	0.897
Random Forest	0.099	0.883	0.179	0.955	0.296	0.098	0.971
XGBoost (ADASYN)	0.471	0.955	0.631	0.994	0.631	0.461	0.998
XGBoost (SMOTE)	0.89	0.9	0.895	0.999	0.895	0.81	0.999
XGBoost (PCA)	0.67	0.704	0.687	0.996	0.687	0.523	0.97

Figure 4.2: Tested classification models compared; values refer to validation set.

We note that the results are markedly better, with g-measure at 0.89 and jaccard index at 0.81; thus, in general it is the combination **Random Undersampling, SMOTE and XGBoost** turns out to be the best for this study.



Next, we tested this optimal configuration but again changing the initial conditions: we apply dimensionality reduction with PCA of `pyspark.ml.feature` so as to understand whether the results could be further improved by using only the most informative

features. To choose the optimal component value (k), we initially set k equal to the total number of features (36). From this we fitt ed the PCA on the balanced dataset with SMOTE and extracted the explained variance, which we then plotted (Fig. 4.3).

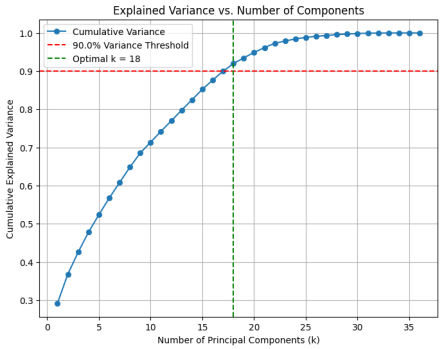
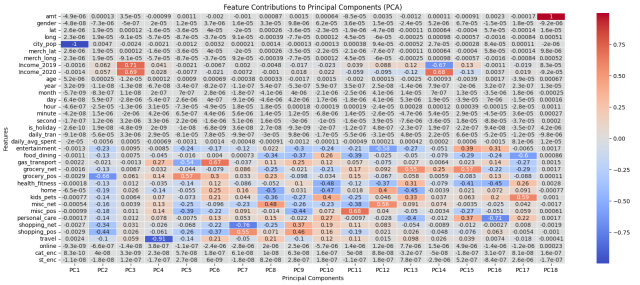


Figure 4.3: Cumulative Explained Variance versus number of PCs.

For our study, we decided to select components below the 90% threshold of explained variance, which is a common rule of thumb to retain enough components to explain a significant proportion of the total variance in the data, avoiding including unnecessary components that contribute minimally to the explained variance. By using the loading of the PCA model we created a heatmap showing the contribution of each feature to each of the principal components. The heatmap reveals the strength of feature contributions across the PCs and we can observe: in general we can see that the highest contributing features are the population of the city in which the transaction is recorded, the amount of the transaction, income information and the categories with shopping and grocery that stand out.



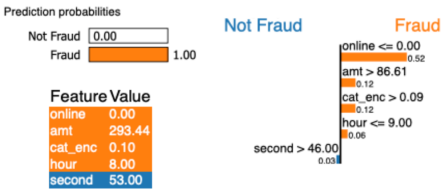
As we reported in Fig. 4.2, the XGBoost model trained on the observed PCs performed well but still worse than the model seen on the unreduced data.

Feature Importance and Local Explanations

To conclude the study, we decided to further analyze the behavior of the features within the best-performing model (XGBoost after SMOTE): we start with the feature importance of the model itself that we accessed via `.get_booster().get_score()`. The top 6 features according to the model are:

Feature	Importance (weight)
cat_enc	717
amt	642
hour	312
day	289
month	221
age	166

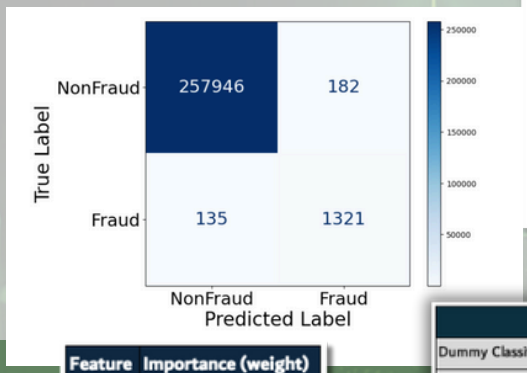
We see that `cat_enc` (the **category encoding** described at the beginning of the chapter) stands out, an indicator that the purchase category may already be a good indicator of the presence of fraud: in fact, during EDA we saw that most fraudulent transactions occur for gas transport or groceries transactions. This is followed by the **amount** of the transaction and some **temporal characteristics**, but also the **age** of the cardholder.
To get a different perspective on the behavior of the features, we then implemented `LimeTabularExplainer` which allowed us to study locally how each attribute contributes to the prediction on the validation dataset.



We considered an instance with prediction `is_fraud=1` and saw that locally the features seen in the model's feature importance are reconfirmed: the binary feature "online" also comes into play this time with a large contribution, highlighting how locally that feature is also taken into account.

Conclusions and Insights

The **clustering** phase provided a multidimensional understanding of transactional and demographic patterns, revealing critical insights into customer behaviors, such as high-spending tendencies among groups and spatial patterns correlating with fraudulent activity. **Geospatial analysis** revealed that fraudulent transactions often occur within mid-range distances from home, emphasizing the interplay between location and spending patterns. The **classification** phase, which included balancing techniques, achieved high performance (ROC-AUC 0.999, PR 0.89/0.9), using key predictors such as transaction categories and cardholder demographics. Together, these findings could improve **fraud detection** systems but also offer business strategies to enhance customer engagement and retention.



Feature	Importance (weight)
cat_enc	717
amt	642
hour	312
day	289
month	221
age	166

Model	Precision	Recall	F1 Score	Accuracy	G-Measure	Jaccard Index	ROC AUC
Dummy Classifier (no balancing)	0.0	0.0	0.0	0.994	0.0	0.0	0.5
Decision Tree Classifier (no balancing)	0.818	0.63	0.712	0.997	0.718	0.553	0.849
Dummy Classifier	0.006	1.0	0.011	0.006	0.075	0.006	0.5
Decision Tree Classifier	0.065	0.889	0.121	0.927	0.24	0.064	0.818
Logistic Regression	0.041	0.791	0.077	0.894	0.179	0.04	0.897
Random Forest	0.099	0.883	0.179	0.955	0.296	0.098	0.971
XGBoost (ADASYN)	0.471	0.955	0.631	0.994	0.631	0.461	0.998
XGBoost (SMOTE)	0.89	0.9	0.895	0.999	0.895	0.81	0.999
XGBoost (PCA)	0.67	0.704	0.687	0.996	0.687	0.523	0.97

GROUP 2


Pietro Argento
600362


Martin Miccoli
672873


Aldo Montinaro
673236


Marco Poiani
677880

References:

- Sparkov Data Generator: https://github.com/namebrandon/Sparkov_Data_Generation
- US states boundaries: https://public.opendatasoft.com/explore/dataset/us-state-boundaries/export/
- Income data: https://fred.stlouisfed.org/release/tables?eid=259515&rid=249