Fortune's Origin: Machine Learning Insights into Self-Made Versus Inherited Billionaire Wealth

Indrajit Ghosh
ighosh2@hawk.iit.edu
Computer Science
Illinois Institute of Technology

Rafail Mahammadli
rmahammadli@hawk.iit.edu
Computer Science
Illinois Institute of Technology

December 1, 2023

Abstract

This project presents an innovative application of machine learning (ML) in the field of economic analysis, focusing on distinguishing between self-made billionaires and those with inherited wealth. The central objective is to uncover and understand the patterns and dynamics behind global wealth distribution, particularly in the context of billionaires' wealth origins. By leveraging Machine Learning robust pattern recognition and predictive analytics capabilities, the study classifies billionaires into two distinct categories: self-made and inherited fortunes. The process begins with preprocessing the dataset by cleaning and finding different statistics about the dataset. We perform PCA analysis on the dataset and perform model fitting with the logistic regression. Subsequently, the project undertakes feature engineering to refine the model inputs and re-assess the model's performance; quasi-feature removal is performed. More iterations are then performed, along with more sophisticated classifiers, including Support Vector Machines (SVM) and Random Forest classifiers, to compare and evaluate their efficacy against the logistic regression baseline. Optimization using hyperparameter tuning is performed, ensuring the selected model is not only accurate but also generalizable The final model selection is based on a comparative analysis, primarily focusing on F1 scores, the confusion matrix, and the ROC curve, to identify the most effective classifier..This research contributes to a deeper, data-driven understanding of wealth creation and inheritance, highlighting the potential of ML in economic and social research.

1 Introduction

In an era of growing wealth disparities, this project addresses a pertinent question: how do billionaires accumulate their wealth? Distinguishing between self-made wealth and inheritance, we step beyond traditional analysis, utilizing machine learning (ML) as a practical tool for unraveling these wealth origins.
Our approach employs logistic regression, Support Vector Classification (SVC), and Random Forest classifiers. These ML techniques were chosen for their effectiveness in classification and their ability to handle complex data sets, making them ideal for our practical, data-driven investigation.
The project's purpose extends beyond theoretical analysis; it's about applying ML to a real-world issue, offering insights into global wealth distribution. We present our work in a structured manner, beginning with data preprocessing, followed by an exploration of various ML models, and culminating in a critical evaluation of our findings.
Through this project, we aim to bridge the gap between theoretical knowledge and practical applica-

tion, providing a nuanced understanding of wealth accumulation in today's economic landscape.

2 Dataset Analysis

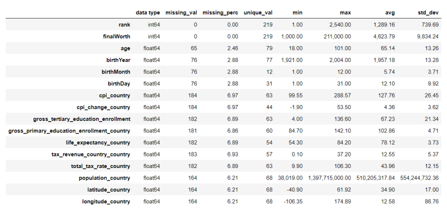
Our analysis is centered around the "Billionaires Statistics Dataset (2023)", encompassing detailed data on global billionaires. This dataset is integral for understanding the distribution and characteristics of the world's wealthiest individuals.

- **Dataset Composition:** The dataset comprises **2640 records**, each representing a billionaire, and is detailed across **35 features**. These features range from basic information like rank and net worth to more intricate economic indicators such as the GDP and CPI changes of the billionaires' respective countries.
- **Key Features:** Notable features include the billionaires' **rank**, their **estimated net worth**, and relevant economic metrics like **country GDP**, **CPI changes, gross primary education enrollment** of the **country and tax revenue system**, providing a multifaceted view of their financial status and economic environment.
- **Data Preprocessing:** Prior to analysis, we will perform essential data preprocessing. This includes cleaning the data, handling missing values, and standardizing the features to ensure a reliable foundation for our machine learning models.

3 Methodology

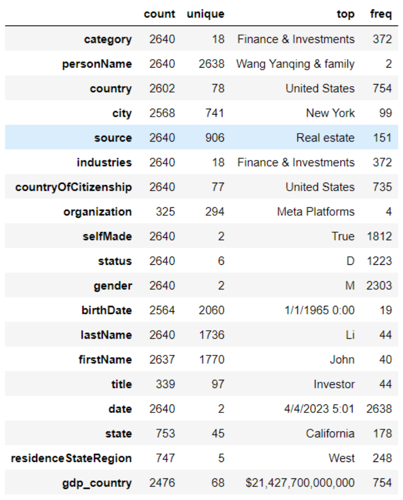
3.1 Dataset Preprocessing

- We found the initial statistics of the dataset to have an idea on the data cleaning activity we are required to perform. We generated a report to find the datatype, missing value, unique value, min, max,average, and standard deviation



	data type	missing_val	missing_perc	unique_val	min	max	avg	std_dev
rank	int64	0	0.00	219	1.00	2,540.00	1,289.16	739.69
finalWorth	int64	0	0.00	219	1,000.00	211,000.00	4,623.79	9,834.24
age	float64	65	2.46	79	18.00	101.00	65.14	13.26
birthYear	float64	76	2.88	77	1,921.00	2,004.00	1,957.18	13.28
birthMonth	float64	76	2.88	12	1.00	12.00	5.74	3.71
birthDay	float64	76	2.88	31	1.00	31.00	12.10	9.92
cpi_country	float64	184	6.97	63	99.55	288.57	127.76	26.45
cpi_change_country	float64	184	6.97	44	-1.90	53.50	4.36	3.62
gross_tertiary_education_enrollment	float64	182	6.89	63	4.00	136.60	67.23	21.34
gross_primary_education_enrollment_country	float64	181	6.86	60	84.70	142.10	102.86	4.71
life_expectancy_country	float64	182	6.89	54	54.30	84.20	78.12	3.73
tax_revenue_country_country	float64	183	6.93	57	0.10	37.20	12.55	5.37
total_tax_rate_country	float64	182	6.89	63	9.90	106.30	43.96	12.15
population_country	float64	164	6.21	68	38,019.00	1,397,715,000.00	510,205,317.84	554,244,732.36
latitude_country	float64	164	6.21	68	-40.90	61.92	34.90	17.00
longitude_country	float64	164	6.21	68	-106.35	174.89	12.58	86.76

Figure 1: Statistics of the Dataset



	count	unique	top	freq
category	2640	18	Finance & Investments	372
personName	2640	2638	Wang Yanqing & family	2
country	2602	78	United States	754
city	2568	741	New York	99
source	2640	906	Real estate	151
industries	2640	18	Finance & Investments	372
countryOfCitizenship	2640	77	United States	735
organization	325	294	Meta Platforms	4
selfMade	2640	2	True	1812
status	2640	6	D	1223
gender	2640	2	M	2303
birthDate	2564	2060	1/1/1965 0:00	19
lastName	2640	1736	Li	44
firstName	2637	1770	John	40
title	339	97	Investor	44
date	2640	2	4/4/2023 5:01	2638
state	753	45	California	178
residenceStateRegion	747	5	West	248
gdp_country	2476	68	$21,427,700,000,000	754

Figure 2: Metrics on object and boolean datatypes

Observations:

- **Date Feature:** The 'Date' field contains only two distinct values, making it less informative for our analysis. Hence, it will be dropped.
- **GDP Country Conversion:**The 'gdp_country' feature, currently in object format with dollar signs, needs to be converted into a numerical format for better analysis.
- **Birth Date Conversion:** The 'birth date' field, currently an object type, will be converted to a timestamp format to facilitate age-related analyses.
- **Country Data Redundancy:** There is redundancy between the 'Country' and 'Country

of Citizenship' fields. We will drop 'Country of Citizenship' to eliminate this duplication.

- **Irrelevant Data Removal:** Several features, including 'date', 'personName', 'city', 'source', 'industries', 'countryOfCitizenship', 'organization', 'lastName', 'firstName', and 'title', 'residenceStateRegion' are deemed irrelevant for our analysis and will be removed to streamline the dataset.

- **Updating datatype:** A few datatypes were updated from objects type like GDP of country, birthdate, birth year, birth month, and birthday to float and integer type

Data Cleaning

- **Feature Removal:** The dataset underwent an initial cleaning phase where all identified irrelevant features were removed. This step streamlined the dataset, focusing on the most pertinent data for analysis, thereby reducing the feature size from **35 to 24**.

- **Duplicate Removal:** Further processing included the elimination of 3 duplicate records, ensuring each data point in the dataset was unique and representative.

- **Handling Missing Values:** The dataset contained **4228** missing values. To address this, a combination of methods was employed:

  - **Fill Forward Method:** This approach was used to fill in missing data by carrying forward the previous data point's value.

  - **Insertion with Mode Value:** For some fields, missing values were filled using the mode value of the respective feature, ensuring a statistically relevant and consistent approach to data imputation

- **Outlier removal using Z-score:** The outliers of the dataset having z-scores greater than 3 were removed specifically for the columns having integer and floating point data. This step increased the quality of the data and the resultant dataset was reduced from **2637** to **2395** records.

3.2 Plotting Observations

Successfuly completion of data preprocessing, we started exploring further into the relationships of the data using various graphs and plots.

The below donut chart gives an idea of the target distribution between the self-made and the inherited wealth billionaires. We can see that around **70.4** percent are self-made while **29.6** percent inherited their wealth from some sources

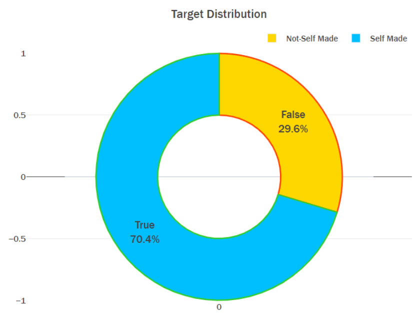


Figure 3: Chart represeting the class balance percentage between self-made vs non self-made billionaires

The below plots of the probablity density histogram and boxplots shows the various data distributions between self-made and non self made billionaires

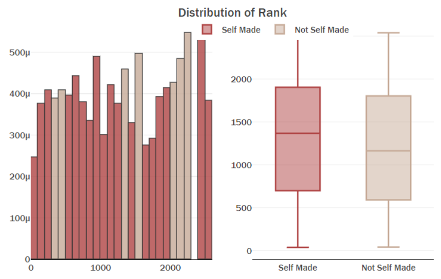


Figure 4: Distribution of Rank between self-made and not self-made

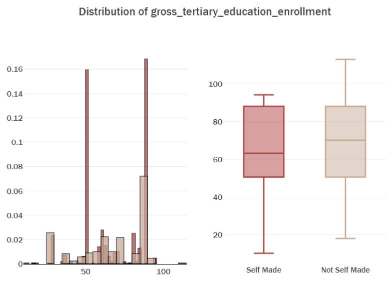


Figure 5: Gross tertiary education enrollment distribution between self-made and not self-made billionaires

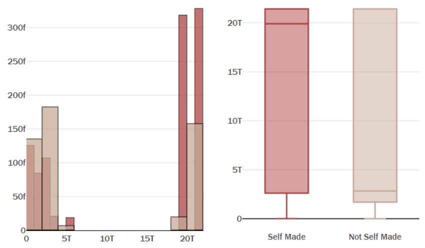


Figure 6: GDP of Country between self-made and not self-made billionaires

The figure shown below is a correlation matrix that shows the feature correlation with the target variable i.e. Self Made vs the selected features in the dataset

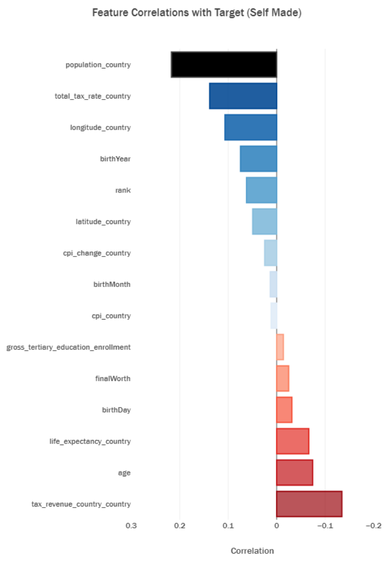


Figure 7: Correlation Graph with target as Self Made Billionaries

Observations: Based on the investigations of the various plots and the correlation matrix, we dropped the non-correlated and irrelevant features like category, country, state, birthDate, birthday , birth month, birthYear. This reduced out final set of features from **24 to 17** .

3.3 PCA results

We implemented Principal component analysis to lower high-dimensional data in lower data dimension space. We chose number of principal components to 3 to reduce dimensionality of data. Then we checked explained ratio of variance of PC1,PC2,PC3 to see each principal component hold variance. Principal component 1 holds 27.2% of the information while the principal component 2 holds only 18% of the information and 3rd PC holds 11% of the information . Also, the other point to note is that while projecting thirty five-dimensional data to a three-dimensional data, 35% information was lost.

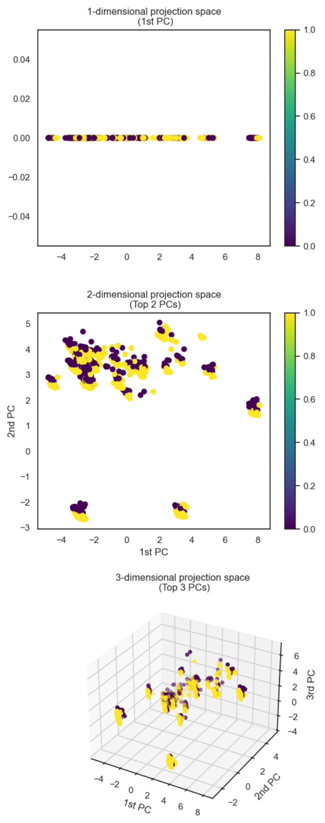


Figure 8: Pca results in projection space

We observe that 2 class points are stacked to each other. Fig 8 illustrates that there is no pattern in 1d,2d, or 3d projection space. The proximity of the points indicates that they share similar feature values and do not exhibit clear distinctions that would enable a straightforward classification based on those features. In such cases, it may be challenging to establish a reliable separation boundary between the classes, and traditional classification approaches might struggle to effectively distinguish between them in the given two-dimensional space.

We decided to remove some columns from the data to detect patterns. We observe in Fig9 that in the 2-dimensional projection space, we see a similar trend that self-made change across the x-axis. Most of its variation occurs in the projection on the first principal component. We can observe that, some combination of the features: gdpCountry, total-taxrateCountry, rank, finalworth, and population-country leads to a consistent change in self-made. Now, the explained ratio of the variance of PC1, PC2, and PC3 is changed also. Principal component 1 holds 34.5% of the information while principal component 2 holds only 23.4% of the information and 3rd PC holds 19.5% of the information. 23% information was lost.

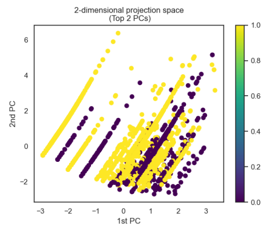


Figure 9: PCA, 2D Projection space on selected columns

3.4 Feature Selection and Data Transformation

We further performed the following steps for the process of feature selection

- **Rank Grouping:** To simplify the 'rank' feature in the dataset, ranks were categorized into 5 distinct ranges. These ranges were defined as intervals of 500, resulting in labels from 1 to 5. For example, a rank of 38 falls under label 1. This transformation was executed using the 'rank' column, with specified bins [0, 500, 1000, 1500, 2000, 2600] and corresponding labels [1, 2, 3, 4, 5].
- **One-Hot Encoding:** Applied to status and gender columns in the dataset, with the first category dropped to prevent multicollinearity, effectively transforming categorical variables into a binary format for model compatibility.
- **Polynomial Feature Creation:** Utilized Polynomial Features for interaction-only terms

without bias, expanding the feature set by generating interaction terms between existing variables, resulting in a new, enriched dataset having **210 features**.

- **Removing Quasi-Constant Features:** Out of the **210** features, quasi-constant features with variance below 0.25 were identified and removed using the VarianceThreshold function. This approach streamlined the dataset by excluding features that offer minimal variability and therefore limited predictive value for the machine learning models. The final features are reduced to **189**.
- **Data Scaling:** Feature scaling was performed using the StandardScaler function. This process standardized the features in the dataset by scaling them to have a mean of zero and a standard deviation of one, ensuring that all features now have the same scale. This standardization is crucial for models that are sensitive to the scale of input data, like many machine learning algorithms, as it ensures that no feature dominates others due to its scale.

3.5 Model Evaluation and Testing

Cross-validation Strategy :

- A 5-fold cross-validation (KFold) was set up with shuffling, and a random state of 42 to ensure reproducibility.
- The F1 score, which balances precision and recall, was chosen as the scoring metric.

Model Selection and Testing Strategy: Three models that were selected were Logistic Regression, Support Vector Classifier(SVC), and Random Forest Classifier for our classification problem and tests were performed using 5-fold cross-validation.
Logistic Regression Results:

- Cross-validation F1 scores: [0.87569061, 0.83844011, 0.82576867, 0.864, 0.84722222]
- Average F1 Score: 0.8502243218040787

Random Forest Classifier Results:

- Cross-validation F1 scores: [0.84813754, 0.82708934, 0.78593272, 0.85714286, 0.81779053]
- Average F1 Score: 0.8272185965388583

SVC Results:

- Cross-validation F1 scores: [0.86909582, 0.85479452, 0.83168317, 0.86410256, 0.85054348]
- Average F1 Score: 0.8540439095384895

3.6 Final Model Selection:

Analyzing the parameters of our dataset, it was found that

- The dataset has non-linear relationships
- The dataset has imbalanced data
- Tolerance to outliers
- Complexity of the dataset
- Robustness to Overfitting
- Feature Interaction

It was decided to select the Random Forest Classifier as the final model

3.7 Performance Tuning

Hyperparameter Tuning: Hyperparameter tuning was performed on the Random Forest Classifier using GridSearchCV, a technique that systematically works through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance.
Hyperparameter Grid Defined: A variety of hyperparameters are considered for tuning:

- **n_estimators**: Number of trees in the forest (50, 100, 200).

- **max_depth**: Maximum depth of the tree (None, 10, 20, 30).
- **min_samples_split**: Minimum number of samples required to split an internal node (2, 5, 10).
- **min_samples_leaf**: Minimum number of samples required to be at a leaf node (1, 2, 4).
- **max_features**: Number of features to consider when looking for the best split ('sqrt', 'log2').
- **criterion**: Function to measure the quality of a split ('gini', 'entropy').
- **class_weight**: Weights associated with classes (None, 'balanced').

GridSearchCV Execution: This tool was used to search over the defined parameter grid using 5-fold cross-validation and find the best set of parameters.
Best Model Identification: The best model and its best parameters are extracted from the GridSearchCV results.
Model Retraining and Evaluation:

- Then the best model is retrained on the entire training dataset.
- The F1-score, a balance between precision and recall, is calculated for both the training and testing datasets, providing a measure of the model's final accuracy.

3.8 Final Observations

The optimal hyperparameters for the Random Forest classifier, determined through tuning, are entropy criterion, a maximum depth of 10, log2 for maximum features, a minimum of 10 samples required to split a node, 50 estimators, and a random state set to 25 for reproducibility. It was observed an increase in the F1 score from 82.7 % to nearly 87 %
Confusion Matrix: The confusion matrix of the final tuned classifier was generated to depict the performance of the model which demonstrates a low count of False positives.

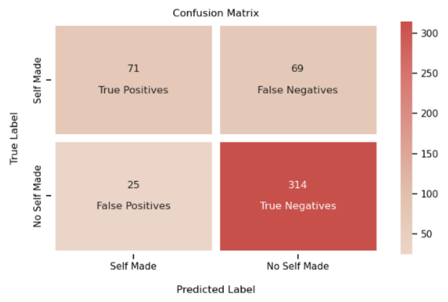


Figure 10: Confusion Matrix between Self Made and Non Self-Made Billionaire

ROC Curve: The AUC value of 0.84 indicates a good predictive performance, with the model being able to distinguish between the classes well. The curve stays well above the diagonal line (which represents a random chance), suggesting the model has a significantly better than random chance at classifying the positive class correctly. The ROC curve approaches the top-left corner of the graph, which is desirable as it implies a higher true positive rate and a lower false positive rate. In summary, the Random Forest Classifier model exhibits a strong ability to classify the positive class, with a low rate of false positive errors, as indicated by the AUC of 0.84

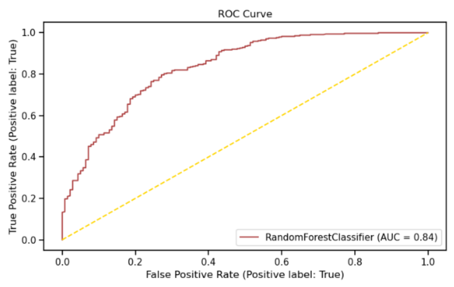


Figure 11: ROC curve for the Random Forest Classifier Model

4 GitHub Repository

The training dataset, code repository, and results of the project can be found on the below GitHub repository – https://github.com/ighosh2/Classification-Model-for-BDS

5 Conclusion

In conclusion, our project leveraged the Forbes Billionaires 2023 dataset to explore various machine learning techniques to identify the most effective classifier. We successfully built a classifier model for the task of identifying self-made billionaires from non self-made. Through meticulous model fine-tuning, we successfully enhanced the model's efficacy from an initial accuracy of 82.7% to approximately 88%. This improvement underscores the value of hyperparameter optimization in machine learning workflows. The increase in performance reflects a substantial step towards achieving a robust predictive model that can provide insightful analyses into the characteristics and patterns within the dataset of the world's billionaires.

6 References

1. Kaggle.com

2. Domingos, P. (2012). "A Few Useful Things to Know About Machine Learning." Communications of the ACM, 55(10), 78-87.

3. Vapnik, V. N. (1995). "Support Vector Machines for Classification and Regression." Neural Networks for Signal Processing V, 1995. Proceedings of the 1995 IEEE Workshop, 55-60.

4. Bellman, R. (1961). "The Curse of Dimensionality." American Mathematical Monthly, 58(8), 681-693.

5. Abdi, H., Williams, L. J., Valentin, D. (2013). "On the Use of Principal Components Analysis in Data Preprocessing." Computational and Mathematical Methods in Medicine, 2013, 1-9.