Wealth Forecast: Exploring Billionaires and Socio-Economic Factors

M. Guo[1], X. Wang[1], and S. Thopey Ganesh[1]

College of Computing and Informatics, Drexel University, Pennsylvania, USA

***Abstract*— This paper takes a deep dive into understanding how socio-economic factors play a role in shaping the financial success of individuals, focusing specifically on wealth distribution among billionaires. We're digging into a dataset that's packed with details about businesses, demographics, and economic indicators to unravel patterns, correlations, and predictive models. We are using a mix of machine learning models such as Logistic Regression, Naive Bayes, and Random Forest. We carefully select a curated set of features, including birth year, gender, country of residence, associated industries, and key economic indicators such as the Consumer Price Index (CPI), CPI change, Gross Domestic Product (GDP), tertiary education enrollment, primary education enrollment, population, and life expectancy, to ensure the robustness of our analysis. These characteristics, taken together, form a complete lens through which we hope to uncover patterns and connections driving individual financial performance. Our approach is designed to look at the big picture, considering all aspects intentionally. We're keeping our options open to explore more aspects that could add depth to what we're learning. Our goal is to share detailed insights into how social and economic factors are connected to how wealth is distributed. We hope this research helps make smarter decisions in economic policies and personal finances, giving a strong base for thoughtful planning.**

Keywords*— *Machine Learning, Classification, Logistic Regression, Naïve Bayes, Random Forest

I. Related Work

Finance, economics, and data science are just a few of the fields that have shown interest in understanding and forecasting wealth categories. Accurately classifying people into various wealth categories has important ramifications for targeted policy, marketing campaigns, and socioeconomic research. We discuss the earlier research in this part that served as the basis for the forecast and classification of wealth categories [1].

Previous exploratory data analysis (EDA) on billionaire datasets has provided important insights into the distribution of wealth across nations and sectors [2]. Fundamental inquiries, such as which nations produce the greatest number of billionaires and which sectors—particularly banking, investment, and manufacturing—produce the greatest number of people with extraordinary wealth have been the focus of research.

The mismatch between billionaires' ultimate financial worth and personal characteristics has been a recurring difficulty, despite these insightful observations. The complex elements impacting wealth growth are difficult for traditional economic models that focus on income, education, and employment to fully reflect. What emerges is frequently a partial knowledge of the ways in which personal traits influence financial achievement.

In order to forecast wealth categories, we use machine learning modes like Logistic Regression, Naive Bayes and ensemble approaches like Random Forest in our study, building on the basis established by earlier research. With an emphasis on accuracy, interpretability, and robustness, we want to contribute to the changing field of wealth prediction by taking into account a wide range of data, including sociodemographic and economic variables.

II. Motivation

The motivation behind uncovering patterns, correlations, and contributing factors influencing individual financial success stems from a profound interest in comprehending the intricate dynamics that shape economic well-being. By delving into the factors that contribute to an individual's financial success, we aim to reveal insights that extend beyond personal finance and resonate with broader societal and economic contexts.

Understanding the implications for wealth distribution is crucial for promoting fairness and equity within a society. By identifying the factors that contribute to financial success, we gain a deeper understanding of the forces that drive wealth accumulation or disparity. This knowledge can then inform economic policies aimed at fostering a more inclusive and just distribution of resources.

The ultimate goal is to utilize the insights gained from this analysis for future policy formulation and decision-making. Policymakers armed with a nuanced understanding of the factors influencing financial success can craft more targeted and effective interventions. Whether it's designing educational programs, implementing social policies, or shaping economic strategies, informed decision-making becomes a powerful tool for creating positive societal change.

On an individual level, the findings empower people to make informed choices about their lives. Armed with knowledge about the factors contributing to financial success, individuals can tailor their decisions to optimize their personal growth and economic well-being. This might include choosing the right education and

career path, making informed investment decisions, or adapting lifestyle choices to align with the identified patterns of success.

In essence, this motivation encapsulates a holistic approach to understanding the intricate interplay of factors that determine financial success. By doing so, we not only empower individuals to make better-informed decisions but also equip policymakers with the tools needed to create a more equitable and prosperous society.

III. Dataset

This dataset, which was obtained from Kaggle, contains detailed information on businesses, industries, and personal profiles of billionaires worldwide. It consists of 35 features, primarily categorized into aspects such as wealth distribution, business sectors, and demographics.

The data includes a breakdown of billionaires across various business sectors, helping us to understand industry-specific dynamics. Additionally, demographic information within the dataset reveals trends within the billionaire community, including age, gender, nationality, and more. Understanding these demographic trends is essential for crafting policies that address potential disparities and ensure equal opportunities for financial success across diverse groups.

This dataset provides a solid foundation for delving into the complexities of global financial success. With its diverse set of 35 features and relevance to the year 2023, which is notably recent, it possesses substantial analytical power. The features only have a small number of missing values, with low bias, therefore we chose this dataset for our analysis.

IV. Data Preprocessing

To prepare our data for model building, we conducted a thorough preprocessing phase on the data frame. Initially, we eliminated irrelevant columns that exhibited little correlation with the target variable—final worth. The selected features for our model included 'finalWorth,' 'birthYear,' 'gender,' 'country,' 'industries,' 'cpi_country,' 'cpi_change_country,' 'gdp_country,' 'gross_tertiary_education_enrollment,' 'gross_primary_education_enrollment_country,' 'population_country,' and 'life_expectancy_country.'

Subsequently, we employed LabelEncoder to transform categorical columns ('country,' 'industries,' and 'gender') into numerical labels. To align with the requirements of machine learning algorithms, we converted the data types of these columns to float. It ensures that the model can interpret and learn from these features.

Our attention then turned to the 'gdp_country' column, where we aimed to clean and convert it to the Decimal type. This involved handling cases of non-numeric or missing values. Leveraging NumPy's np.nan, pandas' pd.isnull, and the Decimal class from the decimal module ensured precision with decimal numbers. Regular expressions were applied to eliminate non-numeric characters, such as '$,' before the conversion process.

Next, we calculated the GDP per capita which is a common practice in economic analysis and modeling. It provides a more meaningful representation of a country's economic output by considering the average income per person. GDP (Gross Domestic Product) measures the total economic output of a country. However, larger populations tend to have higher total GDP even if the average income per person is not high. GDP per capita standardizes this measure by dividing the total GDP by the population, providing a per-person measure. To add this new column, we first converted the 'gdp_country' and 'population_country' columns to numeric type. Then create a new column 'gdp_per_capita' by dividing the 'gdp_country' column by the 'population_country' column.

Moving forward, we calculated quartiles (percentiles) for the 'finalWorth' column in our data frame. Understanding the distribution of values through quartiles—specifically, the first quartile (Q1), median (Q2), and third quartile (Q3)—informed our decision to categorize the 'finalWorth' column into four distinct categories. These categories were defined as follows: below the 25th percentile (labeled as 0), between 25th and median (labeled as 1), median to 75th percentile (labeled as 2), and above the 75th percentile (labeled as 3). Consequently, the 'finalWorth' column was transformed into labels, containing only values 0, 1, 2, and 3. We decided to use percentiles to assign labels to the 'finalWorth' column because they are less sensitive to extreme values or outliers. By dividing the data into quartiles, the impact of extremely high or low values is mitigated, providing a more robust representation of the overall distribution. The use of percentiles makes the categorization adaptable to changes in the dataset. If new data introduces shifts in the distribution of 'finalWorth,' the percentiles will automatically adjust to maintain the defined quartiles.

It is important to note the presence of NaN values in our dataset. We will delve into this in greater detail when discussing the modeling phase.

V. methods

V-A. Logistic regression

In our study, we used logistic regression to do the multiclass classification. The choice of the gradient descent optimization algorithm was made due to its effectiveness in iteratively refining model parameters. Logistic regression is recognized as a fundamental technique adept at

handling binary and multiclass classification tasks. It has the potential to achieve optimal predictive accuracy.

Formulars that are being used in this section include

$$\text{sigmoid}(z) = \frac{1}{1+e^{-z}}$$

and

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\theta(x^{(i)}))_k + (1-y_k^{(i)})\log(1-(h_\theta(x^{(i)}))_k) \quad (1)$$

The implementation of the sigmoid function played a crucial role in introducing non-linearity to the model. It enables the logistic regression model to capture intricate relationships within the data. The sigmoid function's transformation of input values into a range between 0 and 1 is well-suited for modeling complex, non-linear patterns, which is useful in our approach to multiclass classification.

Our logistic regression model also contains the cost function, which is a quantitative measure of the disparity between predicted and actual outcomes. The gradient descent optimization algorithm is used because of its iterative parameter adjustment mechanism. This choice allows the model to converge towards optimal parameter values, minimizing the cost and significantly enhancing predictive capabilities. In scenarios such as high-dimensional feature spaces, such as multiclass classification, gradient descent's efficiency in navigating the parameter space makes it a suitable optimization method.

In addition to these considerations, we also addressed missing values during data preprocessing by replacing them with the median of the respective columns. This decision was made to enhance the robustness of the logistic regression model by ensuring that it is trained on a complete and representative dataset. By replacing missing values with column medians, it mitigates potential biases and contributes to the overall reliability of the model's predictions.

Further steps in the preprocessing pipeline, such as standardizing features and introducing a bias term, were implemented to promote model generalization. Standardizing features ensures that all variables contribute uniformly to the model, preventing any particular feature from dominating the learning process. The addition of a bias term provides the model with flexibility in capturing variations in the data, contributing to its adaptability across diverse scenarios.

The transformation of the target variable into one-hot encoding was motivated by the multiclass nature of the problem. This facilitates the logistic regression model's training for multiclass classification, ensuring that the model can effectively classify instances across multiple categories.

The integration of non-linearity, iterative parameter optimization, and data handling collectively highlighted the efficacy of our model in addressing classification challenges.

V-B. Naïve Bayes

We chose Naïve Bayes to predict the target value for its computational simplicity and efficiency. It is also less sensitive to missing values in the dataset because it will use other attributes instead [3]. To accommodate Naïve Bayes Algorithms, we converted each feature with continuous value into binary ones like 0 and 1 except for gender since it is already in binary. The core of the Naïve Bayes approach is the computation of the posterior probability, expressed as $P(y|x) = \frac{P(y)P(x|y)}{P(x)}$ This formula assumes that features are conditionally independent. Drawing from Professor Burlick's 2023 lecture, the posterior can also be represented as $P(y=i|\mathbf{x}) = \frac{P(\hat{y}=i)\prod_{j=1}^{D} P(x_j|y=i)}{\rho}$, where $\rho = \sum_{k=1}^{K}\left(P(y=k)\prod_{j=1}^{D} P(x_j|y=k)\right)$ [4]. The idea is that we divide the probability by ρ so they sum up to one.

In our implementation, the class prior and generative likelihood were computed following the above-mentioned formulas. We used dictionaries to store the generative likelihood for each class and their corresponding features. Although this approach slightly increases space complexity, it significantly enhances the efficiency of posterior computation, as dictionary look-ups in Python are very efficient with constant complexity.

V-C. Random Forest

The Random Forest's efficacy is built on the two major sources of randomness introduced during training: feature selection and bootstrapped sampling. In contrast to a standard decision tree, which evaluates every feature at each split, Random Forest selects a subset of features at random at each decision point. This nature of unpredictability prevents individual trees from becoming overly specialized and potentially boosts variety among them.

It is essential to provide randomization into feature selection. Each tree in the Random Forest examines a random selection of features while reaching a split decision, as opposed to taking into account all features. This makes the model more resilient and less prone to noise by ensuring that the ensemble captures a more thorough knowledge of the relationships within the data.

It is typical for real-world datasets to have missing values. We use proper assessment and management of missing values as part of our preprocessing method. For example, we use the pandas `to_numeric` function to convert data to numeric format in the `gdp_country` column; non-convertible items are replaced with `NaN`.

In our implementation, key hyperparameters of the Random Forest Classifier are carefully chosen to balance model complexity and performance. The `n_estimators` parameter defines the number of trees in the forest. A moderate value is selected to prevent excessive computational costs while maintaining sufficient diversity among trees.

The maximum depth of any decision tree is controlled by the `max_depth` parameter. To avoid overfitting and guarantee that the model performs well when applied to new data, a shallow depth is used.

`random_state` argument ensures there is reproducibility, which fixes the random seed to provide consistent outcomes over several runs.

Using various training data subsets, several decision trees are built as part of the Random Forest training process. A random subset of traits is taken into consideration for splitting at each node of a tree, creating variation among trees. This procedure increases the overall robustness of the model by preventing some characteristics from having an excessive impact on individual trees.

VI. Results and Evaluation

VI-A. Evaluation on Logistic regression

The training accuracy is approximately 30.91%, and the testing accuracy is around 31.93%. Both accuracies are relatively low, suggesting that the logistic regression model is struggling to capture the underlying patterns in the data. It's essential to investigate potential reasons for this low accuracy, such as the complexity of the data, feature engineering, or model choice. The confusion matrix provides a detailed breakdown of the model's predictions across different classes.

- **Class 0:** The model correctly predicted 95 instances of class 0, but there were also false positives (15 instances predicted as class 0 that were not).
- **Class 1:** The model predicted 20 instances correctly as class 1, but there were also false positives and false negatives.
- **Class 2:** Similar observations can be made for class 2, where correct predictions coexist with false positives and false negatives.
- **Class 3:** The model performed relatively better in predicting class 3, with a higher number of true positives.

Precision (0.31): The precision indicates that, on average, 31% of instances predicted as positive are true positives. The relatively low precision suggests that the model may be making a significant number of false positive predictions.

Recall (0.31): The recall of 31% indicates that the model captures only 31% of all instances belonging to the positive class. The low recall suggests that the model is missing a substantial number of positive instances.

F1 Score (0.28): The F1 score, being the harmonic mean of precision and recall, provides a balanced measure. A score of 0.28 suggests that the model has a trade-off between precision and recall, and improvements in both metrics are needed for a more robust model.

VI-B. Evaluation on Naïve Bayes

Similar to Logistic Regression, the validation accuracy for Naïve Bayes is around 31.3%, precision is at 26.8%, recall is 31.3% and F-measure is at 24.8%. While the performance is less than we aimed for, the consistent poor performances across our algorithms could stem from our dataset.

For Naïve Bayes, we did convert all continuous features into binary ones. While this step aligns with the algorithm's requirements, it inevitably simplifies the dataset, potentially overlooking nuanced relationships within the data.

Lastly, the assumption of conditional independence among features, a fundamental aspect of Naïve Bayes, might not hold entirely true in our dataset. For instance, there could be an underlying correlation between a country's GDP and its gross tertiary education enrollment. Such interdependencies between features could undermine the effectiveness of the Naïve Bayes model.

VI-C. Evaluation on Random Forest

The model can predict on both the training and testing datasets after training. The individual forecasts made by each decision tree in the forest are combined to create the predictions. The performance of the model is then assessed using criteria like accuracy and confusion matrix.

The capacity of Random Forest to offer insights regarding feature relevance is one of its main advantages. We may learn a great deal about the characteristics that are important in forecasting wealth categories by examining how each feature contributes to the model's decision-making process. To comprehend the underlying patterns in the data, interpretability is essential.

Validation Accuracy: 32.45% Precision: 30.492% Recall: 32.45% F1 Score: 30.37% Confusion Matrix:

$$\begin{bmatrix} 126 & 25 & 33 & 61 \\ 76 & 23 & 48 & 52 \\ 68 & 25 & 39 & 73 \\ 74 & 22 & 32 & 95 \end{bmatrix}$$

VII. CONCLUSION

VII-A. Accuracy:

- Logistic Regression: Training Accuracy (30.91%), Testing Accuracy (31.93%)
- Naïve Bayes: Validation Accuracy (31.3%)
- Random Forest: Validation Accuracy (32.45%)

Random Forest shows a slightly higher accuracy compared to Logistic Regression and Naïve Bayes.

VII-B. Precision:

- Logistic Regression: 31%
- Naïve Bayes: 26.8%
- Random Forest: 30.492%

Logistic Regression and Random Forest have similar precision, with Naïve Bayes having a slightly lower precision.

VII-C. Recall:

- Logistic Regression: 31%
- Naïve Bayes: 31.3%
- Random Forest: 32.45%

Random Forest has the highest recall, followed closely by Naïve Bayes and then Logistic Regression.

VII-D. F1 Score:

- Logistic Regression: 0.28
- Naïve Bayes: 24.8%
- Random Forest: 30.37%

Random Forest has the highest F1 score, indicating a better balance between precision and recall. Logistic Regression follows, and Naïve Bayes has the lowest F1 score.

Random Forest generally outperforms Logistic Regression and Naïve Bayes in terms of accuracy, precision, recall, and F1 score. However, it's important to note that the performance of each model depends on the specific characteristics and requirements of the dataset. Further tuning and exploration of features may be needed to enhance the models' overall effectiveness.

In addition, the feature importance analysis reveals that *"birthYear"* has the highest significance with an importance score of 0.220586, suggesting a substantial impact on the model's predictions. Following closely is the *country* feature with an importance score of 0.152099, indicating its noteworthy contribution to the model. Additionally, *"gross_primary_education_enrollment_country"* holds considerable importance of 0.103132. However, the information provided about the *"industries"* feature seems to be incomplete, and further details are needed to assess its importance in the model.

VII-E. future works and Extension

Improving Feature Selection A key area for development is the refinement of our feature selection process. This entails a more rigorous analysis of the dataset to identify features that are most predictive of the target variable. We aim to explore advanced feature engineering techniques, such as Principal Component Analysis (PCA) or other dimensionality reduction methods, to extract more informative and less redundant features.

Reformulating Research Questions Another avenue to explore is the evaluation of our research questions. Our current results suggest that the questions we are attempting to answer may not align optimally with the strengths of the methodologies employed. In future studies, we intend to reassess our research objectives, tailoring them to better suit the capabilities of our analytical approaches. This might involve focusing on different aspects of the dataset or redefining the scope to more closely match the capabilities of our models.

Explore regression models We may want to explore regression models as an alternative, as converting features to categories can result in a loss of information.

REFERENCES

[1] C. Freund, "The origins of the superrich: The billionaire characteristics database," *SSRN Electronic Journal*, pp. 16–1, 2016. (available at: *https://doi.org/10.2139/ssrn.2731353*).

[2] Shivansh R, "Exploratory analysis - billionaires dataset," Accessed in 2023. (available at: *https://www.kaggle.com/code/shivanshr12/exploratory-analysis-billionaires-dataset*).

[3] G. Webb, "Naïve bayes," *Encyclopedia of Machine Learning*, C. Sammut and G. Webb, Eds. Boston, MA: Springer, 2011.

[4] M. Burlick, "Probabilistic/statistical classification," PowerPoint slides, Drexel University, 2023, retrieved from BBlearn.