**Analyzing the Impact of Financial Stability and Family Structure on Elder Health and Longevity: A Machine Learning Approach Using the RAND HRS Longitudinal File 2020**

Yuzhen Zhou, Zeyuan Pang

**Part 1. Executive Summary**

The primary goal of our research is to analyze the impact of financial stability and family structure on the health outcomes. This investigation is crucial as nations globally grapple with the ramifications of an aging demographic, including heightened demands on healthcare systems, pension sustainability, and the overall welfare of the elderly population. Our study aims to provide evidence-based insights to guide policymaking in healthcare, social security, and family support initiatives, thereby facilitating more effective strategies to aid older adults.

For this analysis, we utilize the RAND HRS Longitudinal File 2020, which encompasses 15 waves of interview data collected over two decades. This comprehensive dataset is invaluable for research on health, family dynamics, retirement planning, employment history, and includes imputations for income, assets, and healthcare spending.

By examining the connections between economic status, family structure, and the health of the elderly, we aim to recommend targeted interventions that could improve life quality, reduce medical costs, and promote the sustainability of aging populations.

This project is of particular importance as it confronts a pressing challenge faced by East Asia: its rapidly aging population. This demographic transformation poses significant challenges for the social and economic progress of these countries, necessitating innovative approaches to ensure the well-being of the elderly and their families in a changing societal landscape.

**Part 2. Data Exploration and Preprocessing**

RAND HRS Longitudinal File 2020 is a huge dataset. It took weeks to read and understand the document and organize the data from different waves into a single dataset.

**1.1Data exploration**

After reading the document, several variables of interest were selected for the initial analysis. The response variables are

SHLT: Self-rated Health Level

COGTOT:Cognitive Level

MSTOT:Mental Status Level

The input variables are

'BMI':Body Mass Index

'INHPFN':Total Number of Helpers Ever Helped

'HHHRES':Number of People in Household

'HCHILD':Number of Children

'LIVSIB':Number of Living Siblings

'HAIRA':Individual Retirement Account Assets

'HATOTB':Total Asset Amount

'IEARN':Individual Income

'HITOT':Total Household Income

'PRPCNT':Current Receiving Pension Income

'INHPE':Any employee of institution ever helped

'HINPOV':whether an individual is considered part of the impoverished population and does not live in an institution

'HINPOVA':whether an individual is considered part of the impoverished population and lives in an institution

'PENINC':Current receiving pension income

'HIGOV':Covered by government health insurance plan

'RETMON':Retirement Status Based on Retirement Month

'SLFEMP':Self-Employment Status

In the figure 1 and 2 we can see the description of the features



figure 1 feature descrption1



figure 2 feature descrption2

And in the figure 3 shows the distribution with continues features and in the figure 4 shows the distribution with the categorical features.

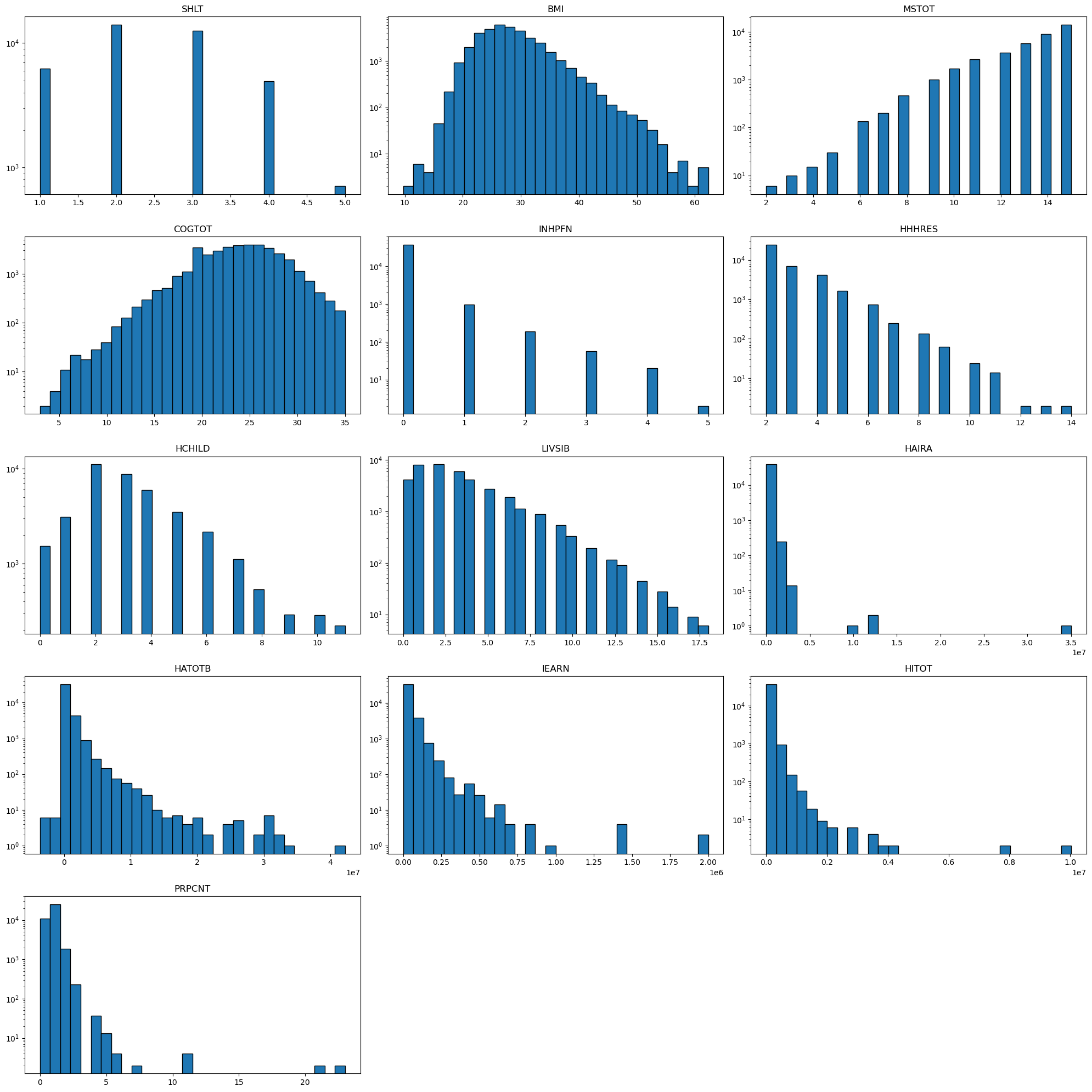


figure 3 continues feature hist

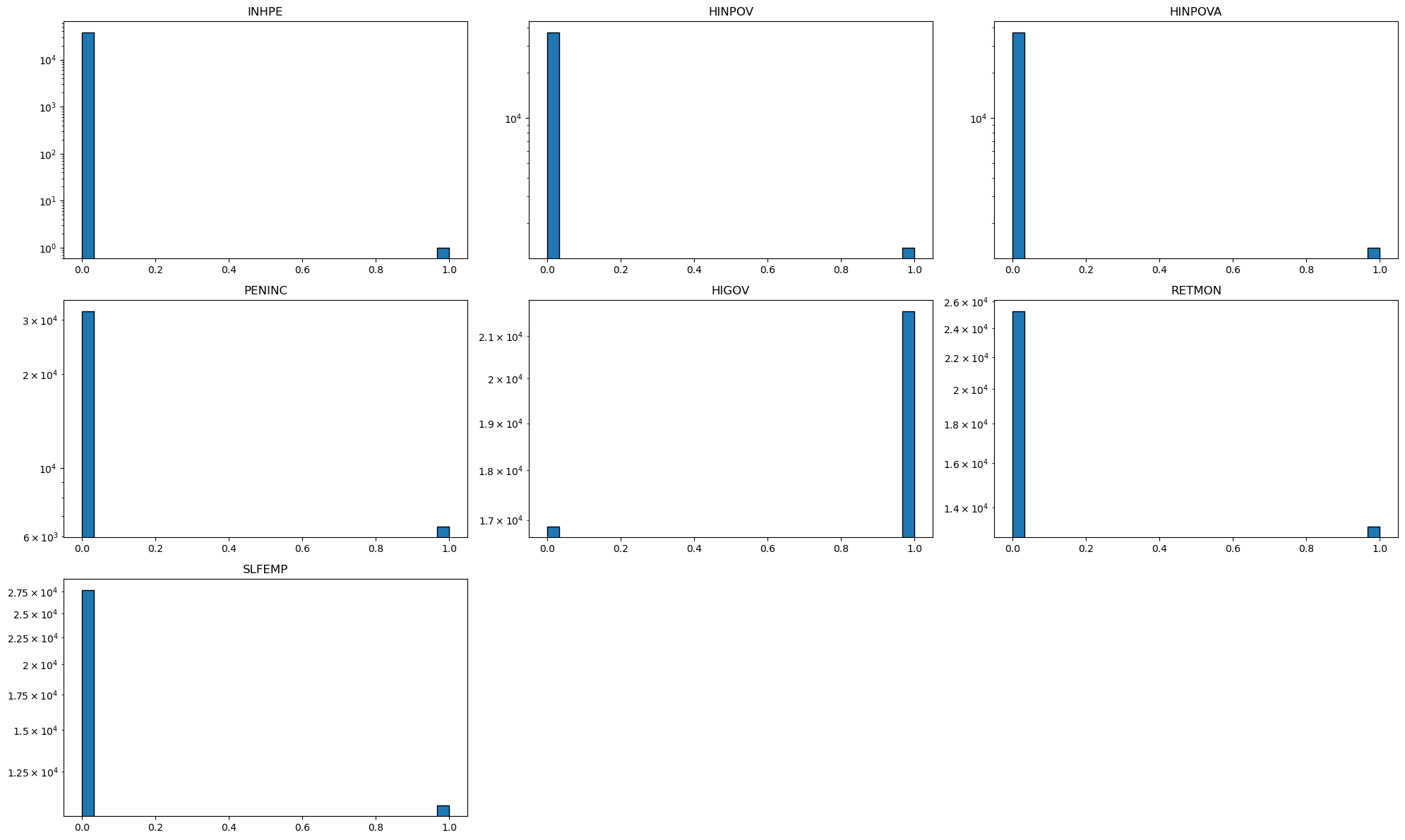


figure 4 categorical feature hist

**2.2 Data Preprocessing**

**2.2.1 Feature selection**

We overall have 12 csv file and they all are large dataset.So first we need to analyze the feature with over 50% data we just remove that.Then we need to select and extract the feature which relate to the finacial conditions we want and combine all the data in one csv file.After that we need to drop the N/A raw.Now, we get the dataset we need to analyze.

**2.2.2 Outliers**

We choose the isolate forest to address the outliers.Isolation Forest is an efficient and specialized algorithm for anomaly detection, leveraging a tree-based approach that excels in identifying outliers with minimal assumptions about data distribution. Its efficiency stems from its ability to isolate anomalies by randomly selecting features and split values, thus quickly partitioning the data space. Unlike distance or density-based methods, Isolation Forest does not require predefined parameters like outlier proportion or neighborhood size, making it broadly applicable and easy to use. It performs well with large datasets, especially in low-dimensional spaces, and is robust across various data distributions without stringent data distribution assumptions.

**2.2.3 Group the data by categorical feature**

These dataset has several categorical features which is interesting we can group them to analyze them as group with other continues features.So we divided the analyze to the two parts.The first part we use all the features to make analyze.And the second part we just create all the group(figure5) with five categorical features to analyze the data.

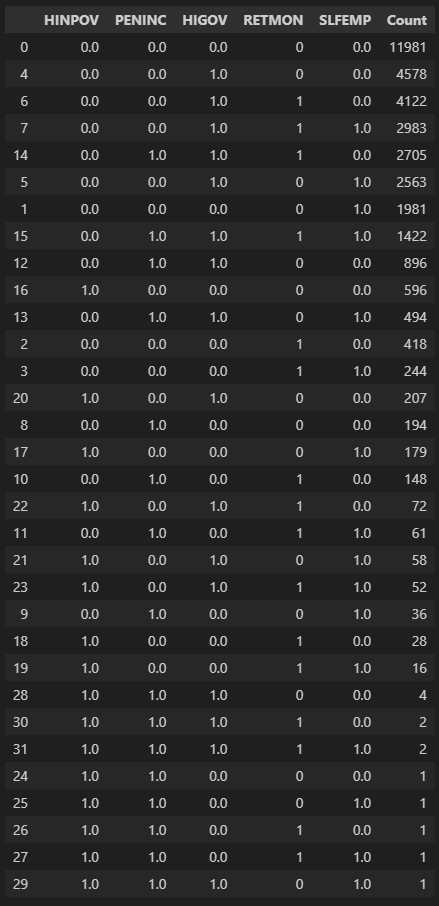


figure 5 grouped data

**Part 3. Model Updates**

**3.1 Model**

## We try different regression model and find that the Random forest regressors is the best model to fit our data.Random Forest Regressors stand out for their ability to combine the predictions of multiple decision trees to enhance overall prediction accuracy and robustness, particularly for regression tasks. This ensemble method mitigates the risk of overfitting associated with individual trees by training each tree on a random subset of the data and using a random subset of features at each split. It excels in handling high-dimensional data without the need for dimensionality reduction, as it can handle numerical and categorical data effectively. Random Forests are robust to outliers and feature a mechanism for assessing feature importance, which aids in model interpretability. Their ability to be parallelized makes them computationally efficient, though their performance is sensitive to the tuning of parameters such as the number of trees and their depth. This blend of features makes Random Forest Regressors a versatile and powerful tool for regression analysis, offering a good balance between complexity and performance.

We analyze the data for two part as mentioned before, the first part we use all raw data to make the prediction and fit the model,this part is just looking how the model performed and overview the data.

The second part is most important part ,we use the grouped data to analyze the data include fitting the model one group by one and finding the importance of each feature.Of course test the MSE and R-squared value to rate the model.

**3.1.1 Original Data**

## In the original Data part,we plot the KDE of Residuals to rate the model performance, we also calculate the mean R-squared and MSE as part of model performance for the raw data and standization data.

Train: R-squared = 0.9569688647468897, MSE = 0.2834284996487357

Test: R-squared = 0.7038317228487762, MSE = 1.9510437640101217

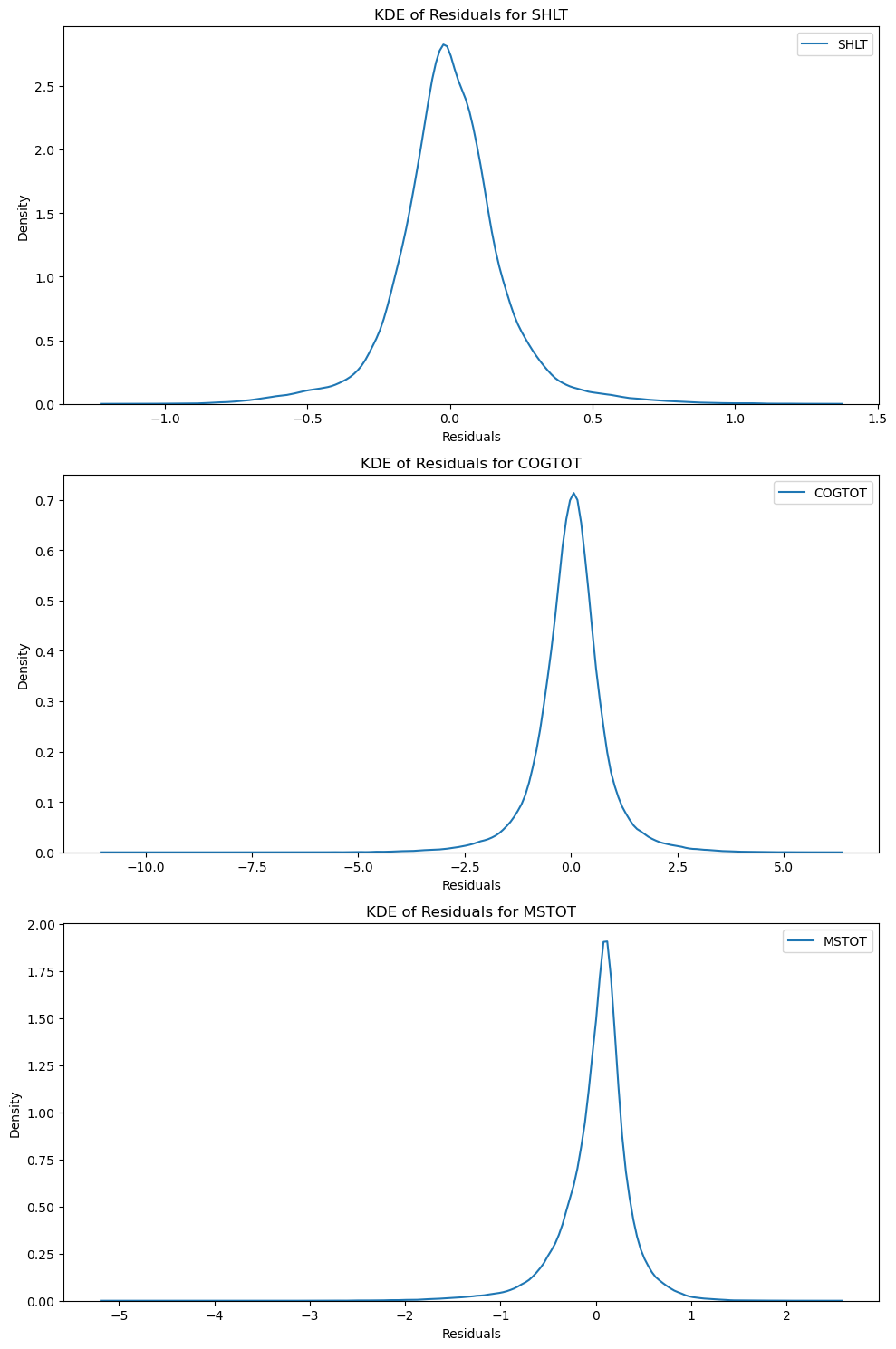


figure 6 KDE of Residuals

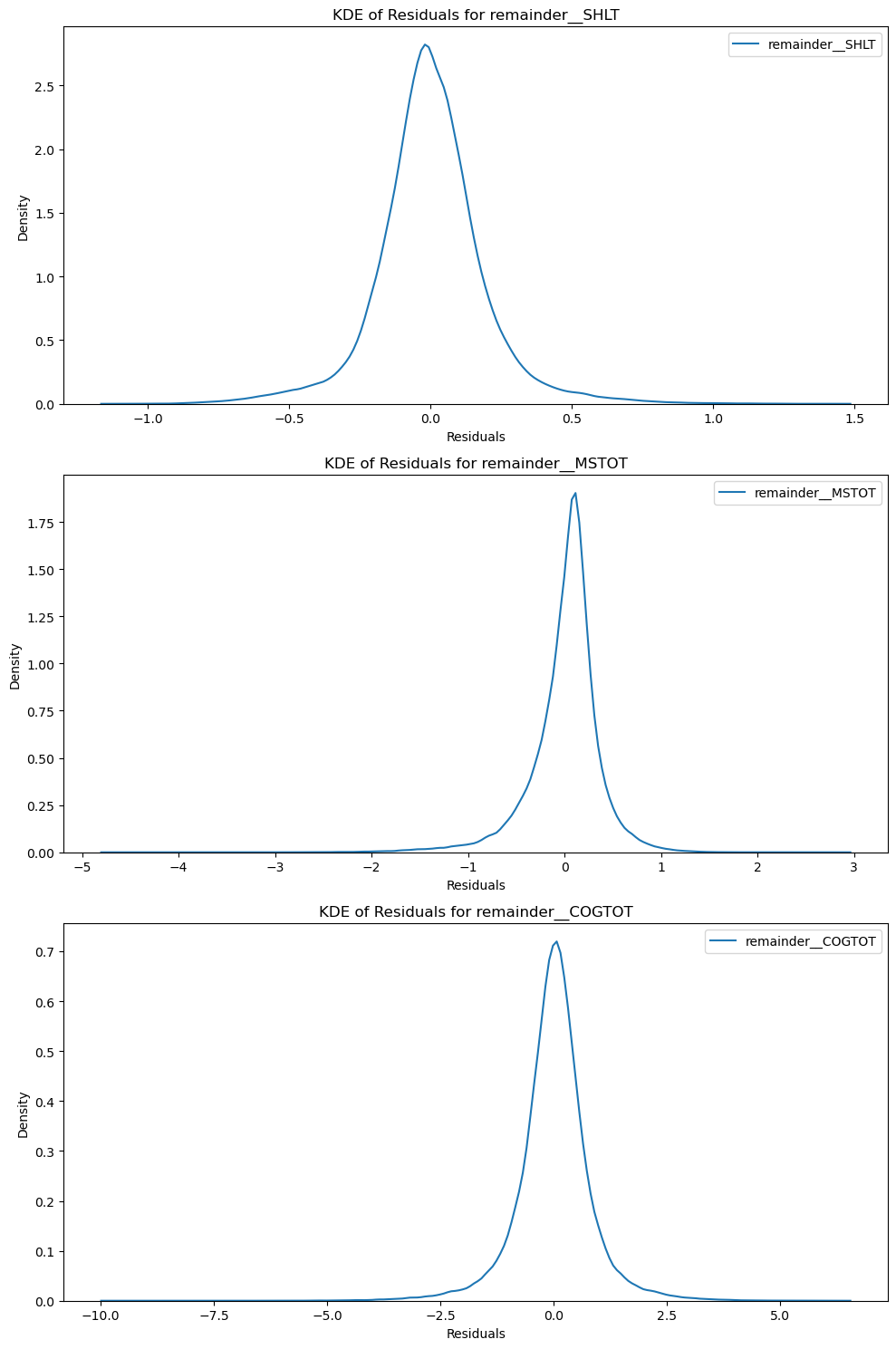


figure 7 KDE of Residuals without outlier

**3.1.2 Grouped Data**

## In the grouped data,we first create all the group and drop the group which less than 500,then we use the randomforest regression to fit the data. Table 1 to 3 is the performance table to each target value.

For target feature:SHLT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | Train R-squared | Train MSE | Test R-squared | Test MSE |
| (0.0, 0.0, 0.0, 0.0, 0.0) | 0.9559 | 0.043 | 0.6882 | 0.3037 |
| (0.0, 0.0, 0.0, 0.0, 1.0) | 0.9529 | 0.0452 | 0.6455 | 0.3416 |
| (0.0, 0.0, 1.0, 0.0, 0.0) | 0.9583 | 0.0382 | 0.7236 | 0.2529 |
| (0.0, 0.0, 1.0, 0.0, 1.0) | 0.9587 | 0.0397 | 0.6857 | 0.3025 |
| (0.0, 0.0, 1.0, 1.0, 0.0) | 0.9559 | 0.0379 | 0.681 | 0.274 |
| (0.0, 0.0, 1.0, 1.0, 1.0) | 0.9544 | 0.0423 | 0.6498 | 0.3246 |
| (0.0, 1.0, 1.0, 0.0, 0.0) | 0.9505 | 0.0428 | 0.6643 | 0.2911 |
| (0.0, 1.0, 1.0, 0.0, 1.0) | 0.9548 | 0.0405 | 0.6594 | 0.3054 |
| (0.0, 1.0, 1.0, 0.0, 1.0) | 0.9525 | 0.0367 | 0.7047 | 0.2285 |
| (0.0, 1.0, 1.0, 1.0, 1.0) | 0.951 | 0.0423 | 0.691 | 0.2668 |
| (1.0, 0.0, 0.0, 0.0, 0.0) | 0.9263 | 0.0901 | 0.5216 | 0.581 |

## For target feature: COGOTO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | Train R-squared | Train MSE | Test R-squared | Test MSE |
| (0.0, 0.0, 0.0, 0.0, 0.0) | 0.9558 | 0.7004 | 0.6804 | 5.0599 |
| (0.0, 0.0, 0.0, 0.0, 1.0) | 0.9515 | 0.7414 | 0.6433 | 5.4349 |
| (0.0, 0.0, 1.0, 0.0, 0.0) | 0.9564 | 0.8072 | 0.7005 | 5.553 |
| (0.0, 0.0, 1.0, 0.0, 1.0) | 0.9559 | 0.8409 | 0.6695 | 6.3069 |
| (0.0, 0.0, 1.0, 1.0, 0.0) | 0.9579 | 0.7152 | 0.6982 | 5.1187 |
| (0.0, 0.0, 1.0, 1.0, 1.0) | 0.9542 | 0.791 | 0.6334 | 6.3539 |
| (0.0, 1.0, 1.0, 0.0, 0.0) | 0.9487 | 0.8315 | 0.6406 | 5.9037 |
| (0.0, 1.0, 1.0, 0.0, 1.0) | 0.9541 | 0.7571 | 0.655 | 5.6545 |
| (0.0, 1.0, 1.0, 1.0, 0.0) | 0.9523 | 0.6897 | 0.7118 | 4.1603 |
| (0.0, 1.0, 1.0, 1.0, 1.0) | 0.95 | 0.7474 | 0.6622 | 5.0523 |
| (1.0, 0.0, 0.0, 0.0, 0.0) | 0.9239 | 1.3952 | 0.5069 | 8.9787 |

## For target feature:MSTOTO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | Train R-squared | Train MSE | Test R-squared | Test MSE |
| (0.0, 0.0, 0.0, 0.0, 0.0) | 0.955 | 0.1532 | 0.6808 | 1.0848 |
| (0.0, 0.0, 0.0, 0.0, 1.0) | 0.9491 | 0.1599 | 0.6346 | 1.1442 |
| (0.0, 0.0, 1.0, 0.0, 0.0) | 0.9553 | 0.1727 | 0.7043 | 1.1449 |
| (0.0, 0.0, 1.0, 0.0, 1.0) | 0.9572 | 0.1373 | 0.6543 | 1.11 |
| (0.0, 0.0, 1.0, 1.0, 0.0) | 0.954 | 0.157 | 0.6757 | 1.1092 |
| (0.0, 0.0, 1.0, 1.0, 1.0) | 0.9532 | 0.1479 | 0.6062 | 1.2484 |
| (0.0, 1.0, 1.0, 0.0, 0.0) | 0.9499 | 0.1616 | 0.628 | 1.2097 |
| (0.0, 1.0, 1.0, 0.0, 1.0) | 0.9545 | 0.1378 | 0.6833 | 0.9501 |
| (0.0, 1.0, 1.0, 1.0, 0.0) | 0.9517 | 0.1261 | 0.6891 | 0.8106 |
| (0.0, 1.0, 1.0, 1.0, 1.0) | 0.948 | 0.131 | 0.6132 | 0.9791 |
| (1.0, 0.0, 0.0, 0.0, 0.0) | 0.9193 | 0.4309 | 0.4933 | 2.6868 |

Then we measure the feature importance to each target vaule in each group.The figure8, figure 9 and figure 10.

**3.2MLM**

Data Preprocessing: Includes cleaning, normalization, feature selection, and potentially feature engineering to prepare the input data for the model.

Model Selection: Based on the problem at hand, a model like the Random Forest Regressor is chosen for its ability to handle complex datasets with a mix of categorical and numerical features.

Hyperparameter Tuning: Using techniques such as grid search or random search to find the optimal model parameters.

Model Training: The model is trained on the preprocessed data using cross-validation to ensure generalization.

Model Evaluation: The model's performance is evaluated using suitable metrics, such as Mean Squared Error (MSE) and R-squared for regression tasks.

Model Morphism: Involves refining the model through processes such as feature reselection, hyperparameter retuning, or using different models entirely based on the performance outcomes.

**Part 4. Source Code**

[YuzhenZhou1327/ESE527\_Project\_HRS (github.com)](https://github.com/YuzhenZhou1327/ESE527_Project_HRS)

**Part 5. Next Steps**

**Part 6.Appedix**

