**Income Prediction**

Ryan Volpi

Data Set: Marketing/Income Data in the Elements of Statistical Learning Textbook

**Data Description**

This project is based on the Income Dataset from the book Elements of Statistical Learning. It consists of 8993 selected responses from a questionnaire filled out by shopping mall customers in the San Francisco Bay area. It includes 14 demographic attributes with a mix of continuous and categorical data types. One important characteristic of this dataset is a prevalence of missing data values for many observations.

The features included in the dataset are the following:

* Annual Income of Household (Numeric Tiered Categorical)
* Sex (Binary)
* Marital Status (Categorical)
* Age (Numeric Tiered Categorial)
* Education (Tiered Categorical)
* Occupation (Categorical)
* How long lived in area (Numeric Tiered Categorical)
* Dual Incomes (Categorical)
* Persons in Household (Numeric Tiered Categorical)
* Persons in Household under 18 (Numeric Tiered Categorical)
* Householder Status (Categorical)
* Type of Home (Categorical)
* Ethnic Classification (Categorical)
* Language Spoken in Home (Categorical)

The goal of our project is to predict values for the Income feature. This feature describes the annual household income of the responder. Though the response describes a continuous variable, it is encoded as an ordinal categorical variable with values corresponding to ranges of income. For example, ‘1’ refers to incomes less than $10,000, ‘2’ refers to incomes between $10,000 and $15,000, and so on. Because of the encoding of this response variable, we treated the problem as a supervised classification problem. The most common value in our response was ‘1’ which made up 0.192 of our observations. Therefore, we use 0.192 as our baseline accuracy, as a model which predicted ‘1’ for every observation would have that accuracy on average.

The provided sample size (n = 8993) is sufficient for our goal and the proportion of observations for each class of the target variable were represented in relatively even proportions. We therefore did not need to perform and under/over-sampling or make any other adjustments to account for sample size.

**Data Preprocessing**

All data preprocessing steps were implemented in pipelinesdesigned to prevent any chance of data leakage. Every transformation was fit on training data and merely applied to the test data. Filling missing values was a major consideration since a large number of observations were missing one or more values. To fill missing values, we created a class which implements K Nearest Neighbors imputation on every feature. The class trains a KNN algorithm to predict the non-missing values of each feature using the values of the other features. This trained model is then used to fill in the values of the missing observations. This imputer is integrated into our pipeline so that it is never trained on any validation or testing data. It serves as the first step in our preprocessing pipeline and is applied to every feature so that our dataset contains no missing values when passed to our classifiers.

Our dataset contained three primary types of features. These are binary, categorical, and numeric. For numeric variables, our preprocessing steps after missing value imputation included scaling the data to have min 0 and max 1, and then fitting and applying a yeo-johnson power transformation to remove skew. This guarantees that each feature has a comparable scale and is approximately normally distributed.

For categorical and binary features, after conducting missing value imputation, we used one-hot-encoding to transform each categorical feature into multiple dummy variables representing each possible response. This allows us to model the effects of individual responses such as english or spanish for language, If we were to keep the categorical features as is, our models would interpret them as numeric, making faulty assumptions such as that a value of 3 in language, corresponding to Spanish, is greater than a value of 1, representing English.

Because all of our features were either categorical, or numeric features collected in terms of discrete intervals (eg. age<18, 18<age<24, etc.), we did not have any outliers in any of our features. We experimented with utilizing PCA to reduce dimensionality and remove correlation but values between n = 5 and 48 resulted in no increase in model performance, so we removed PCA from the pipeline.

**Method**

We chose to implement models that we expected to succeed at multilabel classification of non-linearly separable data. These are Logistic Regression (L1, L2, and no regularization), K Nearest Neighbors, Random Forest Classification, and a Gradient Boosted Tree Classifier. All of these performed well for the given task and were able to identify meaningful patterns in the dataset.

Before performing any operations we separated the data into 80% training data and 20% testing data using stratified random sampling. The testing data was never used for training or hyperparameter optimization. It was only used to evaluate and verify the efficacy of the final models. We did this to ensure that we could create unbiased estimates of generalization performance.

In order to optimize the hyperparameters of each model, we performed a 5-fold cross-validated grid search. In this method, multiple parameter values were used to train and test each model using 5-Fold Cross-Validation. Preprocessing steps were recomputed and performed every time a model was fit to prevent data leakage from skewing our accuracy estimates. The parameters that resulted in the highest average accuracy on the validation folds were chosen. For every parameter, we ensured that we tested multiple values on either side of the optimal value to ensure we had found a minimum.

**Results**

Because our goal was classification, our primary metric for model evaluation was accuracy. Model parameters were selected based on cross-validation accuracy and the models with the best parameters were then re-fit on all of the training data and evaluated on the test set. The results are summarized in the table below.

Figure 1: Comparison of Methods Used

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Train Score | Validation Score | Test Score | **Test Score (Refit)** |
| K-Nearest-Neighbors Classifier (N = 23) | 0.394 | 0.324 | 0.326 | **0.326** |
| Logistic Regression  No Regularization | 0.362 | 0.334 | 0.342 | **0.342** |
| Logistic Regression  L1 Regularized (λ = 0.11) | 0.363 | 0.334 | 0.340 | **0.339** |
| Logistic Regression L2 Regularized (λ = 0.09) | 0.363 | 0.335 | 0.337 | **0.337** |
| Random Forest (N = 200) | 0.870 | 0.317 | 0.310 | **0.310** |
| Boosted Tree Classifier (mx\_dep = 3, min\_wt = 3) | 0.557 | 0.329 | 0.338 | **0.338** |

We achieved the highest test accuracy using logistic regression without regularization. This gave us an accuracy of 0.342 on the test data. KNN, boosted tree classification, and regularized logistic regression gave similar accuracies but our random forest model appeared to suffer from overfitting resulting in low test accuracy. This accuracy represents a significant improvement over our baseline accuracy of 0.194.

**Conclusion**

We successfully created a model capable of estimating an individual’s income category using 13 other demographic features with an estimated generalization accuracy of 0.342. This is especially impressive given that the data was not easily separable, contained many missing values, and truncated all numeric features, including the response, into discrete bins.