IFT6390 Project1

Marthy Garcia, Rafael Garcia

March 2021

1 Introduction

This paper report on the design and documentation of a machine learning model, as a part of a Introduction to Machine learning course at the University of Montreal.

Our main task is to predict the mean per capita (100,000) cancer mortalities for a given set of details about US counties. In this document, the authors describe the pre-processing of data, model selection, testing, hyper-parameter tuning, and validation.

2 Data

The data used in this project can be found in [1]. The data has various attributes about the US counties, cancer rates, resident characteristics distribution.The train data-set comprises 32 features and 3047 observations containing numerical and categorical variables. The full data dictionary can be found in [2].

Features with more than 50% missing values were dropped (e.g."pctsomecol18_24"). On numerical variables missing values were replaced by the mean of the fields. Categorical variables did not have missing values.

2.1 Training and test data sets

The original data was split in training and testing data set. As advised in [3], 20% of the data was randomly selected for testing, the remaining data was assigned to training.

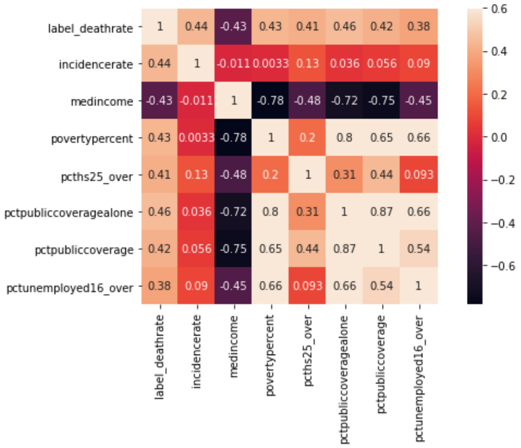

Figure 1: Correlation matrix

3 Pre-processing steps

An exploratory data analysis (EDA) that comprised correlation and distribution analysis was performed on the training data set to inform model selection. Figure 1 depicts the correlation matrix. Multiple variables show a strong correlation within them. As an example, the mean per capita cancer mortalities has a positive relationship with the median income per county. As per Figure 1, the median income per county is highly correlated multiple variables as well. This indicates that we should perform feature selection before implementing a model.

Figure 2 shows the target variable against the feature: % of country residents with public coverage. The latter depict a linear relationship. The same behaviour is observed with multiple features. A linear regression seems promising.

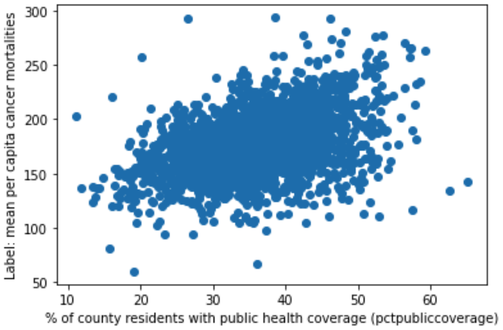

Figure 2: Target variable vs % of country residents with public coverage

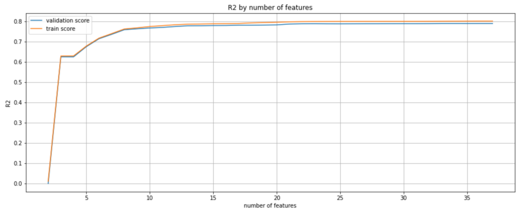

Figure 3: R^2 by number of features

Must of the numerical variables follow skewed distributions and have different scales. Therefore, logarithmic transformation was applied to improve our model fit and score. Logarithmic transformation increased the accuracy of the model by almost 60%. Categorical variables were transformed using one-hot encoding.

4 Model Selection

4.1 Hyper-parameter tuning

As per Figure 3, some features exhibit a high correlation. Feature selection was done by applying a grid search Recursive Feature Elimination (RFE) cross-validation. A 10- fold cross-validation was selected. Models were compared based on the R^2. Figure 3 shows the R^2 value as a function of the number of features. A total of 15 feature were selected.

4.2 Linear Regression

Linear regression is a method that finds the best linear fit to a given independent and dependent variables. The best fit is found by minimizing the sum of squared residuals.

As per the findings from Figure 2, a linear regression method was chosen as per its simplicity, interpretability, and simple implementation. The model is given as follow:

$$Y = XA + E$$

Where

$$X = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,15} \\ 1 & x_{2,1} & \dots & x_{3,15} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,15} \end{bmatrix} Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} E = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

The matrix A consist of the weight of each feature. We solve for A as follows:

$$A = (X^T X)^{-1} X^T Y$$

4.3 Time and space complexity

As per [4] the training complexity of the model is $O(k^2(n + k))$. Where k corresponds to the number of features. The test runtime complexity of linear regression is $O(k)$

The training space complexity is $O(nk + n)$. The test space complexity at run time is $O(k)$.

The model was run in Colab free version. Figure 4 shows the run time per fold. The average fit time is 0.106s with a standard deviation of 0.019.

5 Model score

To ensure reproductivity of the results, a seed was set in all steps.

Performance was validated by applying a 10- fold cross-validation and measured by the R^2. Figure 5 shows the R^2 value for the training and validation

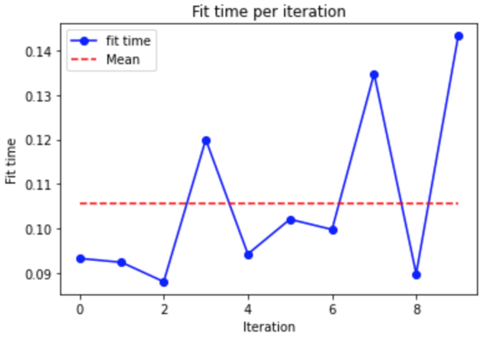

Figure 4: Fit time per fold

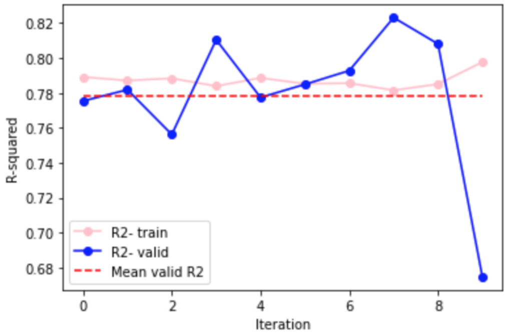

Figure 5: Train and validation R^2 per fold

sets. The test R^2 obtained is 0.775. The Mean Absolute Error (MAE) and the Mean Squared Error (MSE) were also collected. Table 5 summarize the results for the training, validation and test data sets.

Set	mean R^2	std.dev R^2	MAE	MSE
Train	0.787	0.041	0.053	0.005
Valid	0.778	0.041	0.054	0.054
Test	0.775	-	0.054	0.072

6 Conclusion and further work

The objective of this model is the prediction of the mean per capita (100,000) cancer mortalities. We performed the following steps as an iterative process: feature engineering (pre-processing), feature selection and model selection.

Logarithmic transformation of numerical variables significantly increase the performance of our model. An initial EDA, showed a high correlation between the variables. Hence, RFE was applied to select the most important features. A linear regression was built. Cross-validation helped to create a more robust model by no creating a validation set.

This project had limitation in the gathering of additional information. We recommend further work focused on collecting additional features.

References

[1] Linear Regression Exercise 1: Data word

[2] Linear Regression Exercise 1- metadata: Data word,`https://data.world/exercises/linear`

[3] Dangeti, Pratap, "Statistics for Machine Learning: Techniques for Exploring Supervised, Unsupervised, and Reinforcement Learning Models with Python and R," 2017, *Packt Publishing*

[4] Banerjee, Writuparna,"Train/Test Complexity and Space Complexity of Linear Regression," 2020,`https://levelup.gitconnected.com/`