IFT6390 Project1

Marthy Garcia, Rafael Hernandez

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 dataset 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.

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 with 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 depicts a linear relationship. The same behaviour is observed with multiple features. Therefore a linear regression seems promising.

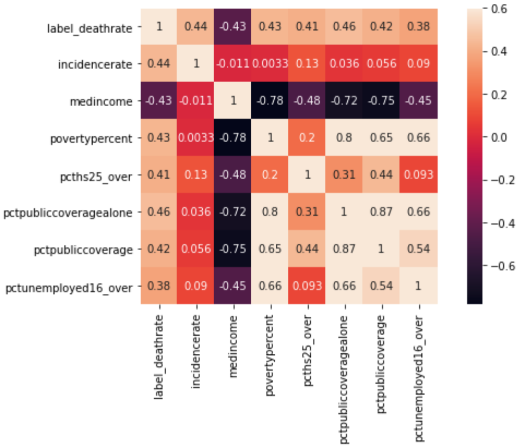

Figure 1: Correlation matrix

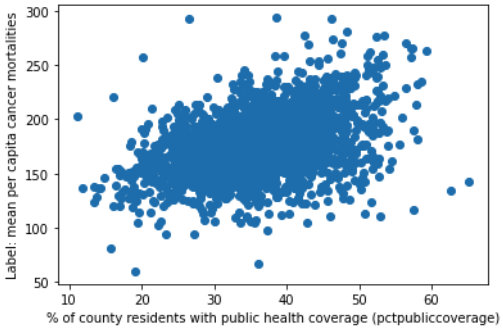

Figure 2: Target variable vs % of country residents with public coverage

Most of the numerical variables follow skewed distributions and have different scales. Therefore, a logarithmic transformation was applied, ultimately improving our model fit and score. As a result, a logarithmic transformation increased the accuracy of the model by almost 60%. As for categorical variables, they were transformed using one-hot encoding.

4 Model Selection

4.1 Hyper-parameter tuning

As per Figure 1, some features exhibit a high correlation. Feature selection was done by applying a grid search Recursive Feature Elimination (RFE) cross-validation. To verify the validity of the model, a 10- fold cross-validation was selected and later compared based on their respective R^2 performance. By looking at the figure 3, once again, using the R^2 measure, we can compare the performance obtained by both the training set and the validation set during the feature selection method. Visually we can notice the slope becoming horizontal around 15 features, making it a sound number for choosing our features.

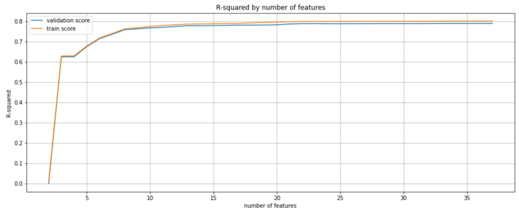

Figure 3: R^2 by number of features

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:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

Where

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & ... & x_{1,15} \\ 1 & x_{2,1} & ... & x_{3,15} \\ \vdots & \vdots & ... & \vdots \\ 1 & x_{n,1} & ... & x_{n,15} \end{bmatrix} \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \epsilon = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

The vector β consists of the weight of each feature. We solve for β as follows:

$$\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

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 $O(k)$.

5 Model score

To ensure reproductivity of the results, a seed was set in all steps. The model was run in Colab free version. Due to the size of the dataset being small, the average fit time was 0.106s with a standard deviation of 0.019.

Performance was validated by applying a 10- fold cross-validation and measured by the R^2. Figure 4 shows the R^2 value for the training and validation 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 summarized the results for the training, validation and test data sets.

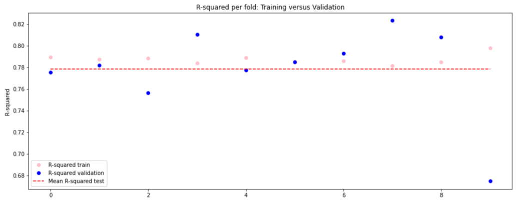

Figure 4: Train and validation R^2 per fold

Set	mean R^2	std.dev R^2	MAE	MSE
Train	0.787	0.004	0.053	0.005
Valid	0.778	0.041	0.054	0.006
Test	0.775	-	0.054	0.005

6 Conclusion and further work

The objective of our model was to predict the cancer mortality mean per capita (100,000). We performed the following steps as an iterative process: feature engineering (preprocessing), 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 verifying the distribution of our performance. For instance, we can see on 4 that our model did not perform well on the last fold.

This project had limitation in the gathering of additional information. We recommend that future versions of the model are focused on collecting additional observations.

References

[1] Linear Regression Exercise 1: Data word https://data.world/exercises/linear-regression-exercise-1/workspace/project-summary?agentid=exercisesdatasetid=linear-regression-exercise-1

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

[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/