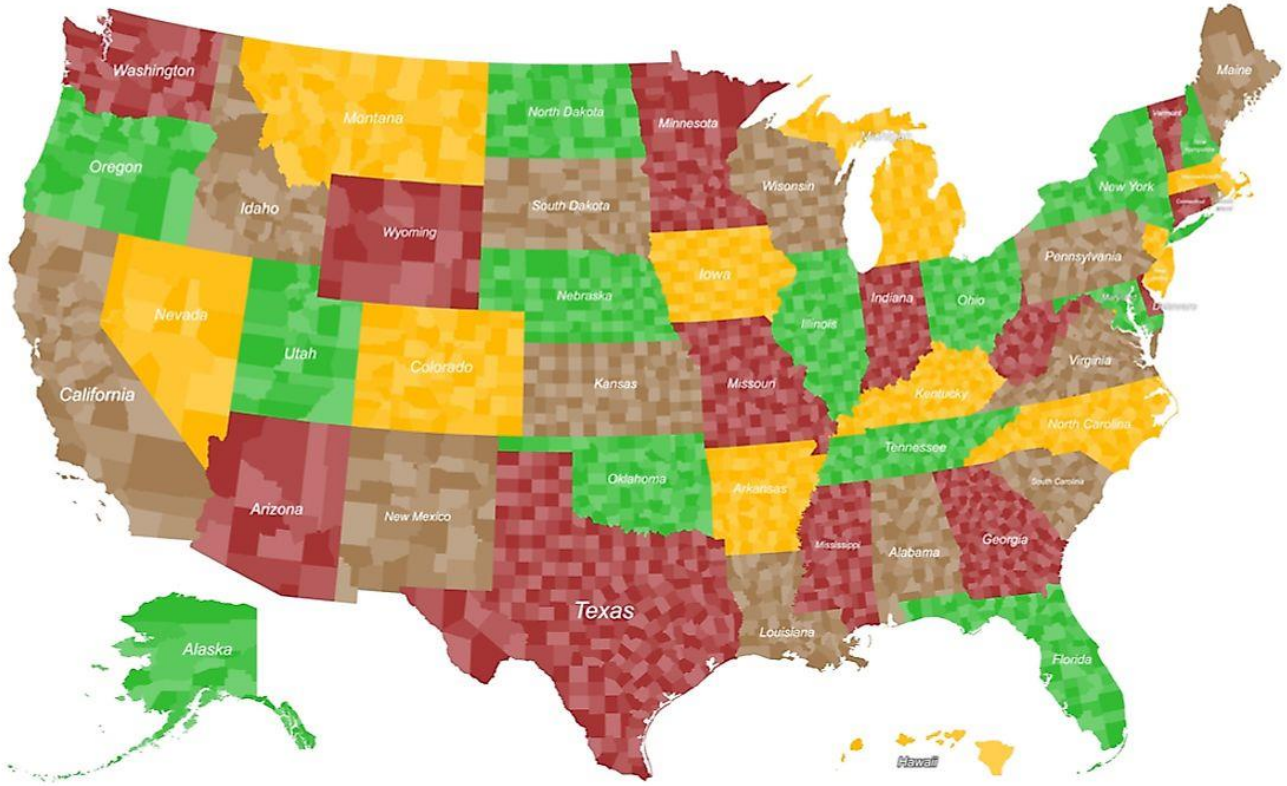
Python B – Foundations of Computer Science

PYTHON – Regression models

Predicting the death by cancer rate for US counties



Report

Simon Kennedy

19 June 2022

Predicting the death by cancer rate for US counties – Simon Kennedy

Abstract

Four regression based models were built to predict the death rate from cancer for US counties. The original data was cleaned and preprocessed to prepare and improve the data for the modelling process. The model candidates are 1. Linear model with SGD Regressor, 2. Support Vector Machine for regression(SVR), 3. Random Forest Regressor, 4. Neuronal Network model. The data was split into training and testing sets. Following initial fitting, hyperparameter tuning was performed using K fold cross validation. The models were evaluated with all models outperforming the baseline. Hyperparameter tuning improved model accuracy in all but the Random Forest Regressor. Principal Component Analysis(PCA) improved the SVR model. The Neuronal Network model was the best performing model on this data and was saved for future use.

Introduction

Four regression based predictive models will be built and evaluated for the prediction of mean deaths by cancer for US counties based on demographic and cancer mortality rates from 2010 to 2016 in the cancer_reg.csv file from the OLS Regression Challenge (Rippner, N 2016), using the Python language.

The dataset has 3047 rows and 34 columns. The target variable is '*TARGET_deathRate'*, which represents the mean deaths per 100,000 population from 2010 to 2016 for each US county. Each row contains the observations of a singular US county. The variables of the dataset pertain to mean cancer mortality rates and case numbers from 2010 to 2016 and demographic information from the 2013 US Census estimates. See Appendix A for the full data dictionary.

The machine learning pipeline was following for this project. The dataset was firstly cleaned to remove outliers and null values and feature engineering to reduce the categories in a predictor. Correlation analysis revealed several variables with collinearity then feature selection was performed along with a Principal Component Analysis(PCA) to reduce the dimensionality of the data.

Normalisation and Principal Component Analysis were performed within the sklearn pipeline function to prepare the data for modelling.

The four models are 1. Linear model with SGD Regressor, 2. Support Vector Machine for regression(SVR), 3. Random Forest Regressor, 4. Neuronal Network model. They were training and tested using a 'hold out' set and hyperparameters were tuned with K fold cross validation in an attempt to optimise performance.

Each model's performance was evaluated by mean square error(MSE), root mean square error(RMSE) as well as cross validated results to check for over fitting. These metrics were compared firstly with the Baseline and then all other models.

We are looking for the best performing model, which will be saved for future use.

Materials and methods

1. Load the dataset

The data for the project was downloaded as a .csv file from data.world.com.

https://data.world/nrippner/ols-regression-challenge/file/cancer_reg.csv

The data was upload and processed using Python v 3.8.8 in 5 separate Jupyter notebooks. Notebook 1 contains all the cleaning and pre-processing steps and notebooks 2 to 5 contains the models and their evaluations.

2. Preprocessing

The original .csv file was read into Jupyter Notebook 1 as a pandas dataframe. All types were correctly read in, with all variables being numeric (integer or float) except for 'Geography' and 'binnedInc'. 'Geography' is a nominal

categorical variable representing the US county and State, and 'binnedInc' is an ordinal categorical variable representing median income per capita, binned by decile.

Missing values and Outliers

From inspection of the dataset several variables were shown to have less than the full count as well as some variables with max values significantly outside the normal range. 'PctSomeCol18_24' had 75% of its data missing, so was removed. 'PctEmployed16_Over' had only 5% missing and 'PctPrivateCoverageAlone' had 20% of its data missing. The missing or null data of these 2 variables was replaced with the column mean. Other more complex imputations methods may achieve a better result when modelling, however this was beyond the scope of this project.

Histogram plots were used to identify outliers. Plotting all 34 variables was too large visually to be useful. The variables identified as having max or min values outside of the normal range were plotted individually along with a 'zoomed' plot with low limits on the y axis to identify which values were outside the normal range and could thus be removed by filtering for values less than an appropriate value.

Feature Engineering - Categorical variables

The 'Geography' variable was split into two columns, 1. 'County' and 2. 'State'. This would produce a new predictor being 'State'. 'State' has only 51 different levels compared to the 3047 of 'Geography'. The 'State' string values were then transformed to numeric labels in preparation for the data modelling. 'County' was removed.

The 'binnedInc' variable also required conversion from string to numeric. As these were ordered categories the names were replaced manually to ensure they were kept in the correct salary order.

Feature Selection

The large number of features was best suited to correlation analysis rather than individual plotting against the target variable. A pairplot and correlation heatmap was produced. Features with high collinearity (pearsons correlation > 0.8) could be identified from the heatmap. Those with the lowest correlation to the target variable were dropped. Six features were dropped in total due to collinearity. No feature demonstrated a correlation greater than 0.5 with the target variable. All features without high pairwise correlation were therefore kept as part of the dataset.

The dataset was renamed and saved as **cancer_clean.csv** ready for use in building the predictive models.

Normalisation

Normalisation was performed using MinMaxScaler() within a the Pipeline function from sklearn.pipeline. MinMaxScaler() which transforms the variables to a range between 0 and 1. Normalisation was required for Principal Component Analysis(PCA) and for all of the predictive models used in this project.

Feature reduction using Principal Component Analysis(PCA)

PCA is an important strategy to investigate whether feature reduction can help improve the modelling process in terms of accuracy and computational expense, particularly with a dataset such as this with now 26 variables. The cumulative variance was plotted to give a visual representation of the how many components contributed to the variance. PCA will be applied to the Support Vector Machine for regression model (Notebook 3, Model 2.)

3. Split the dataset into Training and Testing Data

The preprocessed data was split using the 'hold out' strategy into 75% training and 25% testing using train_test_split from sklearn, with a seed set to ensure the same split occurred for each different model. K fold cross validation was used with 3 splits of the data during the hyperparameter tuning. This is to reduce the probability of over or underfitting which can occur with the 'hold out' strategy. K fold was chosen over RepeatedKFold as it was less computationally expensive which was important for the more complex models.

4. Build Data Models

Four prediction models were built from the sklearn library using the **cancer_clean.csv** dataset:

1. Linear model with SGDRegressor
2. Support Vector Machine for regression
3. Random Forest Regression
4. Neuronal Network model

A pipeline was used to scale the data and define the model prior to fitting on the training data using only the models default parameters.

Model 1. Linear model with SGDRegressor is presented in Notebook 2. A linear model was used as the first model to begin the project as it is a simple type of the predictive modelling. The SGDRegressor applies a stochastic gradient descent loss estimation to each sample (Scikit-learn 2022) and has multiple hyperparameters including penalty (the regularisation terms of l1, l2 and elasticnet). This gives the model more flexibility to fit this high dimensional data compared to pure linear regression, which has no hyperparameters for tuning.

Model 2. Support Vector Machine for regression (Notebook 3) is better suited to this dataset being a multiplanar model. It was tuned particularly for all its different kernel types (linear, poly, rbf and sigmoid) giving it excellent flexibility to fit with this data. Several other hyperparameters were tuned to enable further improvement in model performance. PCA was also performed on the tuned SVR model.

Model 3. Random Forest Regressor (Notebook 4) is chosen as a candidate as it is different and complex style and able to decorrelate trees, reduce error and generally achieve less overfitting (Towards Data Science 2022), which could be an advantage for this data that has some collinearity and high dimensionality.

Model 4. Neuronal Network model (Notebook 5) is able to model non-linear relationships and potential relationships that are not seen through standardised exploratory data analysis. They are known to perform well with high dimensional data such as this dataset making this an important model to trial (Elite Data Science 2022).

5. Hyperparameter tuning

A grid of different hyperparameters with a range of values was used to tune each model using GridSearchCV() on the K fold cross validated data. This was then fitted on the training data and the process was timed. The best parameters were then selected and used to create the optimal tuned model.

6. Model Evaluation

A Dummy Regressor Baseline was established using the 'mean' strategy to predict the baseline values. Model predictions were then gained from the training and testing data.

The metrics used to assess performance of all models including the baseline, untuned and tuned models were 1. mean square error (MSE) and 2. root mean square error (RMSE). These were obtained on both the training and testing data and compared for an initial gauge on over or underfitting. The tuned model's accuracy was evaluated for the cross validated negative root mean square error.

The predictions from each tuned model were plotted against the actual values and the best model was saved for future use.

Results and Discussion

Feature Selection significantly reduced computational processing times during hyperparameter tuning in the more complex models (from 40 minutes to 22 minutes in the Neuronal Network model).

The Baseline model has an RMSE of 28.69 deaths per 100,000 population. Each model aims to achieve a lower RMSE than this which represents a better performance.

Model 1. Linear model with SGD Regressor improved only 0.7 percent with hyperparameter tuning. The linear nature of the model restricts the extent of improvement available from tuning its parameters as compared to multiplanar(SVR), Random Forest and neural network models. It did however tune in only 5.8 seconds meaning

Notebook	Model	MSE	RMSE	TUNED MSE	TUNED RMSE	% imp with TUNING	With PCA MSE	With PCA RMSE
1.	**Baseline**	823.34	28.69	-	-	-	-	-
2.	**1. Linear SGDRegressor**	422.21	20.55	416.00	20.40	+0.7	-	-
3	**2. SVR**	528.98	23.00	348.64	18.67	+23.2	345.64	18.58
4.	**3. Random Forest**	362.61	19.04	363.87	19.08	-0.2	-	-
5.	**4. ANN mlp**	372.74	19.31	337.46	**18.37**	+5.1	-	-

Table 1.1 Performance comparisons for all the models.

no real computational delays. It had a cross validated (cv) accuracy of -20.47. This model had the lowest performance as measured by RMSE, which is expected due to its inability to account for non-linear relationships likely present in this high dimensional data.

Model 2. Support Vector Machine for regression (SVR) responded very well to hyperparameter tuning with an improvement of 23.2%. This model with its different SVR kernels of 'linear', 'poly', 'rbf' and 'sigmoid' combined with other parameter value ranges, is very flexible in finding the optimum combination. It has an RMSE of 18.67 and a cv accuracy of -19.14. The run time for the tuning was 12 minutes due its higher complexity.

PCA was applied to the tuned SVR model. All 26 components achieving the lowest RMSE. This did not reduce dimensionality but may have achieved slightly improved results by reducing collinearity.

Model 3. Random Forest demonstrated significantly higher RMSE on testing than training suggesting over or underfitting. This could be due to the size of the data, especially the high number of features compared to rows. The model did not improve with tuning, likely due to the tuning being on KFold cv data which is more robust and resistant to overfitting. The RMSE on testing data of 19.08 was still very comparable to the other models as was the cv accuracy of -19.14. The run time was 7 mins, being a slight advantage over the SVR.

Model 4. Neural network mlp improved only slightly with tuning, however had a good performance with the best RMSE of 18.37 and cv accuracy of -19.61 again very similar to the other models. It had the highest run time of 22 mins on tuning. This is tolerable given the size of the dataset is not likely to change in the future given each observation is that of aggregated data for each US county (unless more variables are added).

The Neuronal network model is the preferred model with less signs of over or under fitting than the Random Forest and a better RMSE than both the Random Forest and SVR models. It was saved for future use.

Conclusion

All of the four models evaluated were more accurate at predicting the mean death rate by cancer in US counties than the mean baseline. The Support Vector Machine Regressor responded very well to hyperparameter tuning and also mildly to Principal Component Analysis. The Random Forest model has some issues with under or over fitting and may be better trained with a cross validated dataset. The Multi Layer Perceptron Neuronal Network model (model 4) was the best suited model to the high dimensional data of approx. 3000 instances demonstrating an RMSE of 18.37 mean deaths per 100, 000 population for a US county. This model has been saved for future use.

Some limitations of the project were the data having high dimensionality, collinearity and low target correlation. Approximately 3000 rows may have impacted the Random Forest variance in results for train compared to test. Further improvement in performance could be achieved by tuning or trialling different normalisation methods.

References

Rippner, N 2016, *OLS Regression Challenge*, data.world, viewed 19 May 2022, <https://data.world/nrippner/ols-regression-challenge/file/cancer_reg.csv>.

Scikit-learn documentation 2022,*'sklearn.linear_model.SGDRegressor'*, viewed 26 May 2022, < https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html>.

Towards Data Science 2022, *'Pros and cons of various machine learning algorithms'*, viewed 18 June 2022, <https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6>.

Elite Data Science 2022, 'Machine learning algorithms: strengths and weaknesses', viewed on 18 June 2022, <https://elitedatascience.com/machine-learning-algorithms>.

Appendix A

Data Dictionary	https://data.world/nrippner/ols-regression-challenge
TARGET_deathRate:	Dependent variable. Mean *per capita* (100,000) cancer mortalities(*a*)
avgAnnCount:	Mean number of reported cases of cancer diagnosed annually(*a*)
avgDeathsPerYear:	Mean number of reported mortalities due to cancer(*a*)
incidenceRate:	Mean *per capita* (100,000) cancer diagnoses(*a*)
medianIncome:	Median income per county (*b*)
popEst2015:	Population of county (*b*)
povertyPercent:	Percent of populace in poverty (*b*)
studyPerCap:	*Per capita* number of cancer-related clinical trials per county (*a*)
binnedInc:	Median income per capita binned by decile (*b*)
MedianAge:	Median age of county residents (*b*)
MedianAgeMale:	Median age of male county residents (*b*)
MedianAgeFemale:	Median age of female county residents (*b*)
Geography:	County name (*b*)
AvgHouseholdSize:	Mean household size of county (*b*)
PercentMarried:	Percent of county residents who are married (*b*)
PctNoHS18_24:	Percent of county residents ages 18-24 highest education attained: less than high school (*b*)
PctHS18_24:	Percent of county residents ages 18-24 highest education attained: high school diploma (*b*)
PctSomeCol18_24:	Percent of county residents ages 18-24 highest education attained: some college (*b*)
PctBachDeg18_24:	Percent of county residents ages 18-24 highest education attained: bachelor's degree (*b*)
PctHS25_Over:	Percent of county residents ages 25 and over highest education attained: high school diploma (*b*)
PctBachDeg25_Over:	Percent of county residents ages 25 and over highest education attained: bachelor's degree (*b*)
PctEmployed16_Over:	Percent of county residents ages 16 and over employed (*b*)
PctUnemployed16_Over:	Percent of county residents ages 16 and over unemployed (*b*)
PctPrivateCoverage:	Percent of county residents with private health coverage (*b*)
PctPrivateCoverageAlone:	Percent of county residents with private health coverage alone (no public assistance) (*b*)
PctEmpPrivCoverage:	Percent of county residents with employee-provided private health coverage (*b*)
PctPublicCoverage:	Percent of county residents with government-provided health coverage (*b*)

PctPubliceCoverageAlone:	Percent of county residents with government-provided health coverage alone (*b*)
PctWhite:	Percent of county residents who identify as White (*b*)
PctBlack:	Percent of county residents who identify as Black (*b*)
PctAsian:	Percent of county residents who identify as Asian (*b*)
PctOtherRace:	Percent of county residents who identify in a category which is not White, Black, or Asian (*b*)
PctMarriedHouseholds:	Percent of married households (*b*)
BirthRate:	Number of live births relative to number of women in county (*b*)

(*a*): years 2010-2016

(*b*): 2013 Census Estimates