PREDICTING VIRGINIA BEACH RESIDENT SATISFACTION RATE WITH MACHINE LEARNING

A PREPRINT

Alan Zhai
Computer Science and Environmental Science
University of Virginia
asz9qm@virginia.edu

Siyeon (Shaun) Kim
Computer Science and Statistics
University of Virginia
sk5ps@virginia.edu

Jaspreet Ranjit
Department of Computer Science
University of Virginia
jr4fs@virginia.edu

July 30, 2020

1 Abstract

The goal of this project is to leverage machine learning techniques to produce a model that predicts overall life satisfaction for residents of Virginia Beach using various aspects of the city such as public transportation, commodities, and other properties relevant to city planning. The primary focus of this research is Virginia Beach, but the goal is to produce a framework for a model that can be generalized and used in other cities to predict life satisfaction rates for residents. The data from the survey was standardized and pre-processed to convert categorical variables to numerical values. The problem of predicting life satisfaction was modeled with regression and random forest models. Classes of life satisfaction were also generated and classification was also performed on life satisfaction. Demographic features of residents were also introduced to serve as another avenue of exploration in predicting life satisfaction rates.

2 Motivation

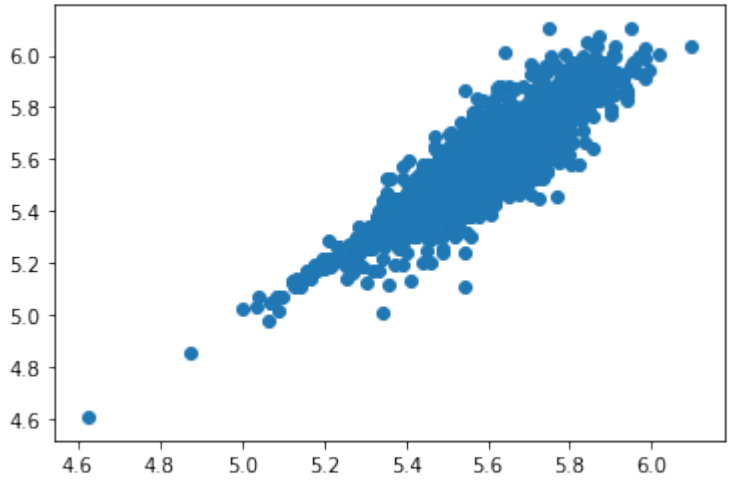
The primary focus of the project is to predict the overall satisfaction score of the residents of Virginia Beach on the basis of the characteristics of the city. As students who are interested in urban planning, it is important to know what aspects and design features of a city are working well and what areas could use improvement with the overall goal being to increase citizen satisfaction. Furthermore, current city planners use common statistical techniques to model the data and draw conclusions. Although this can be useful to visualize the data, it does not provide much insight or information as to what factors contribute the most to a resident's living experience. As a result, it would be beneficial to explore and develop a model that could give city planners more insight into which aspects of the city need to be improved, or which aspects should be more heavily considered in planning new cities. By developing a model, it would also give further insight into what specific areas need to be improved in order to promote overall life satisfaction in residents.
Another use case of the proposed model is for prospective home buyers who are looking to move to another location. The model can provide guidance as to which city will provide better quality of life and help individuals make more informed decisions. In macro-scale, although the scope of the data is limited to Virginia Beach, the proposed model can be extended to other cities who collect similar citizen satisfaction data. For this specific motivation, the demographic data will be used as a predictor from the survey result to tailor one's satisfaction score more personally.

3 Methods and Results for Regression

Prior to discussing what models were used for this experiment, it is crucial to go over how the label - the overall satisfaction score - is calculated in the study. The overall satisfaction score is a summation of encoded value of the survey result from Q2-Q36. These sets of survey questions ask a subject how satisfied he or she is with specific aspect of Virginia Beach. For "very dissatisfied," the score is 1; for "dissatisfied," the score is 2; for "satisfied," the score is 3; for "very satisfied," the score is 4. The higher overall satisfaction score indicates that the observation is more satisfied with the living conditions of the Virginia Beach. Later Demographic data was introduced to the model in order to allow for more accurate predictions. The demographic data included gender, income, education level, number of years that the residents lived in Virginia Beach, etc. All the features were scaled similarly to the overall satisfaction score. To predict the score, we used the following methods for our experiments:

1. ***Linear Regression with Demographic Data***
   Initially, we used linear model to predict the overall satisfaction score. This was the baseline model since linear regression is primarily used for regression models to start with. The root mean square error (rmse) for the experiment that included demographic data in the predictors was 0.13158. Given the context of this data set and the problem, this rmse is particularly high. The mean and the standard deviation are 5.595749 and 0.169593 respectively; the error spans almost one full standard deviation and thus does not serve as a good model for the satisfaction score. The linear regression model is not complex enough for the problem and the data does not fit well to the model: both the training and testing error are 0.125 and 0.1315, so additional models will need to be investigated.
2. ***SGD Regression with Demographic Data***
   The second method we used was the SGD regression model. The best rmse for this model was 0.129, which was ever so slightly better than our linear model. For hyper-parameter tuning, we used cross validation and grid-search technique to find the optimal tolerance and max iteration values. However, the difference made by the hyper-parameter tuning was very minimal.
3. ***Polynomial Regression with Demographic Data***
   For our next step, we built a polynomial model. We tried polynomial degree from 2 to 4. However, the model was massively over-fitting for any degree above 3. Thus, we used the polynomial model with degree 2. On the training set, the model's rmse was 0.07782, which is a great improvement from out initial linear regression. However, on our test data set, the rmse was approximately 0.16. Thus, even the polynomial regression with degree 2 was over-fitting the data set, yielding greater rmse than our linear model.

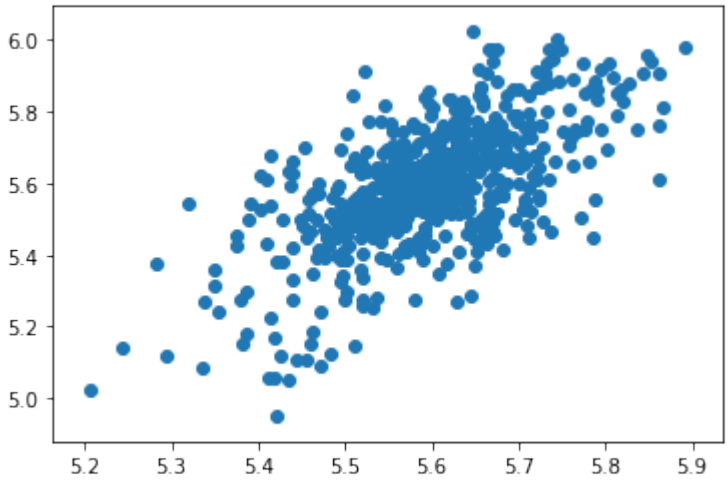
   We also attempted many regularization techniques on our polynomial model to solve the over-fitting issue. First, we applied Lasso regularization technique, which again yielded 0.16. For the Ridge regularization method, it yielded 0.07802. The elastic net method produced the rmse of 0.1588. Out of these regularization methods, we decided to use ridge methods. However, when we tested the ridge model with 10-fold cross validation to test the validity of the model, the model yielded rmse of 0.165 on average, showing that the models were also overfitting.



4. ***Random Forest with Demographic Data***
   We also used random forest model to predict the overall life satisfaction score. Random forest was chosen

since it is a powerful tree ensemble method that should improve upon traditional regression techniques. Initially, the random forest model was able to produce rmse of 0.04738 on our training data. However, when we ran 10-fold CV, the rmse was around 0.126 on average. Again, the model massively over-fit the data.

Attempting to minimize rmse as much as possible, we tried hyper-parameter tuning using grid-search. Our best parameters were bootstrap: false, max features: 7, n estimators: 150. Even, after the hyper-parameter tuning, the final rmse on our test data was 0.13061, which is marginally better than our linear model.



5. ***Gradient Boosting Regression***
   Gradient boosting regression builds an additive model which allows for the optimization of differentiable loss functions. It works by fitting a regression tree in each stage on the negative gradient of the loss function specified during initialization. Gradient boosting consists of three parts: the loss function to be optimized, a weak learner that makes predictions, and an additive model to add weak learners to minimize the loss function. The loss function for this regression problem was the mean squared error. Gradient boosting is general enough that any differentiable loss function can be used so a new boosting algorithm does not need to be calculated for each of the different loss functions. One of the reasons gradient boosting was tested next was because, decision trees are used as the weak learner and the additive model is developed by adding trees one at a time to existing trees that are already in the model where gradient descent minimizes the loss when adding trees [3]. Since gradient bootsting is similar to random forest in that both models are ensemble tree algorithms, we decided that it would be beneficial to try out another ensemble method for an improvement in accuracy. The RMSE for this model was: 0.133 which still did not beat random forest. One drawback to gradient boosting is that it can overfit the training dataset very quickly so it would be useful to try out regularization methods to avoid overfitting for the future.

6. ***Neural Network with Demographic Data***
   Our final model was the neural network method. We used the existing MLPRegressor Neural Net to predict the overall satisfaction score. We predicted that this model would perform most optimally since the MLP regressor neural net optimizes the squared-loss using stochastic gradient descent. Furthermore, the regression models that have already been tried did not perform well. Although neural networks are more often used for classification, they can also be used for regression when traditional regression models do not fit the data perfectly and a more complex network is required [2]. In this case, a neural network can provide a more powerful prediction. Regression models only work well if they fit the data well. Since this was not entirely the case, we decided to use a neural network that would dynamically pick the best type of regression. However, our rmse on the test data set was identical to our random forest rmse, which is 0.130626 thus showing that the data was not a good representation of the label.

From the 5 different models we tested, random forest and neural network performed the best with the rmse of 0.1306. Again, this rmse is not the best give the standard deviation of the data set.

The link to the Google colab for regression can be found here: `https://colab.research.google.com/drive/1L4G7MNO02mWcffCC-MchIl0Uy7b3AADJ`

4 Methods and Results for Classification

1. ***Classification with Demographic Data***
   Since our regression models were not accurate in predicting life satisfaction, we decided to perform classification on the data set. We divided the data into four classes by splitting the overall satisfaction score into quartiles, which resulted in each class containing around a quarter of the subjects. Classification was performed using Kernelized SVM with kernels being linear, rbf, and polynomial. The models yielded an accuracy of 41.9, 42.4, and 31.2 respectively.
2. ***Optimizing RBF Hyperparameters***
   We attempted to optimize parameters for the rbf model in hopes to increase the accuracy as the model had the highest base accuracy. This was performed using randomized search to yield a better range of parameters and then grid search was performed to maximize the accuracy. However, the best accuracy achieved was 52.9 which was far from adequate. The final hyper parameters configuration for this was 'C': 2.5, 'coef0': 2.8, 'degree': 2, 'gamma': 0.26, 'kernel': 'rbf'.
3. ***Optimizing Linear Hyperparameters***
   Since the base accuracy for linear was slightly lower than the accuracy for rbf, we decided to perform randomized search and then grid search to further optimze the parameters. The best accuracy was 44.1 and the hyper parameter configuration was 'C': 0.8, 'coef0': 2.8, 'gamma': 0.2, 'kernel': 'linear'.
4. ***Optimizing Polynomial Hyperparameters***
   We also decided to also try to optimize the parameters for poly since the previous two models were not satisfactory. This was done through randomized search to yield the best range of parameters and then grid search was performed to maximize the accuracy of the model. The final accuracy was 49.0 with parameter configuration of 'C': 3, 'coef0': 2.8, 'degree': 5, 'gamma': 2.7, 'kernel': 'poly'.

Out of the three models tested, rbf had the highest accuracy after hyper parameter tuning, the accuracy was 52.9 percent. However, given that we divided the score into 4 quartiles, the hypothetical random selection will yield 25 percent accuracy. Thus, our model definitely works better than random selection. However, it is not satisfactory.

The link to the Google colab for the classifcation can be found here: `https://colab.research.google.com/drive/1_PADHzkwDqf6tpWbI5PaJ6boxrQDO_Fz`

5 Conclusion and Next Steps

The primary reason the model did not perform well was due to the biased data set. There was no collection of features that could accurately and precisely predict a complex metric such as life satisfaction. Furthermore, the data did not show any strong correlations between the features and the label. This made it difficult to perform develop regression models that could predict the life satisfaction. In the future, it would be beneficial to perform feature engineering. Feature engineering transforms the raw data into features that are a better representation of the underlying problem. This could potentially increase the performance by selecting combinations of features that are better predictors of the labels. Since this dataset contained a large number of features, this approach was not heavily investigated but could further improve the model by creating new input features from existing ones. This would require more domain knowledge on the underlying problem to determine which features would most likely contribute to life satisfaction in Virginia Beach. Feature engineering would also have been difficult on the dataset as most of the features were categorical and would be hard to interpolate features from. In addition, stacking models could be useful for this type of dataset since a single model did not provide a useful prediction. However, in the end, the model is as good as the data. Although there was a lot of data to analyze, if the dataset itself did not present any strong correlations to begin with, the resulting models cannot make useful predictions. So it would be useful to find more data that could be analyzed to develop a stronger model.

More fundamentally, the dataset itself may fundamentally have high level of irreducible error beyond mathematical correlation. When people are conducting surveys, even a person is satisfied, he or she can mark dissatisfied towards some aspects of Virginia Beach. In order for this dataset and the model to work properly, we should find another dataset that is related to the subject of our interest to combine to strengthen the dataset.

6 Contributions

Every member of the team contributed to finding the appropriate data set and developing the primary interest for the project. Alan pre-processed and cleaned the data and initiated building models for the check point. Shaun implemented

more complex models like polynomial regression, SGD regression, different regularization, and random forest hyper parameter tuning. Jaspreet implemented the neural network to predict the satisfaction score. Alan tested a classification model to see if the model performed better on classification rather than regression. Jaspreet and Shaun wrote the script and developed the video. The final report was worked on as a group.

7 Acknowledgment

This study was sponsored by the CS 4774 class at the University of Virginia as a part of the Machine Learning for Virginia project. As such, we would like to acknowledge Professor Richard Nguyen.

References

[1] https://missinglink.ai/guides/neural-network-concepts/neural-networks-regression-part-1-overkill-opportunity/

[2] https://towardsdatascience.com/deep-neural-networks-for-regression-problems-81321897ca33

[3] https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

[4] https://setscholars.net/2019/02/09/how-to-use-mlp-classifier-and-regressor-in-python/