Reproducibility Report of "Comparative Study of Regression Models and Deep Learning Models for Insurance Cost Prediction"

Melina Rogers

Introduction

The goal of the 2019 paper "Comparative Study of Regression Models and Deep Learning Models for Insurance Cost Prediction" published in Springer by Shinde et al. was to build various regression models using Medical Insurance data and compare their results. This would then determine the most accurate method to forecast future patient insurance charges.

Data Exploration and Pre-Processing

The paper writes that the dataset is "cleaned" first removing any unknown values, however, this dataset does not contain any unknown values and therefore was not part of the code when attempting to reproduce the results. It is unclear why they write that unknown numeric values were replaced by the mean of the corresponding features when this was not necessary. One reason this could have been stated/implemented was for generalizability purposes since the information in the dataset would not be hard to obtain (gender, region, age, etc.).

Further data processing included log transformation, label encoding, and one-hot encoding which was replicated in the Python code successfully (Fig. 1, Fig. 2). Further data exploration was done in the initial R code which found that without these processing steps, the results would be much less likely to predict the charges without overfitting.

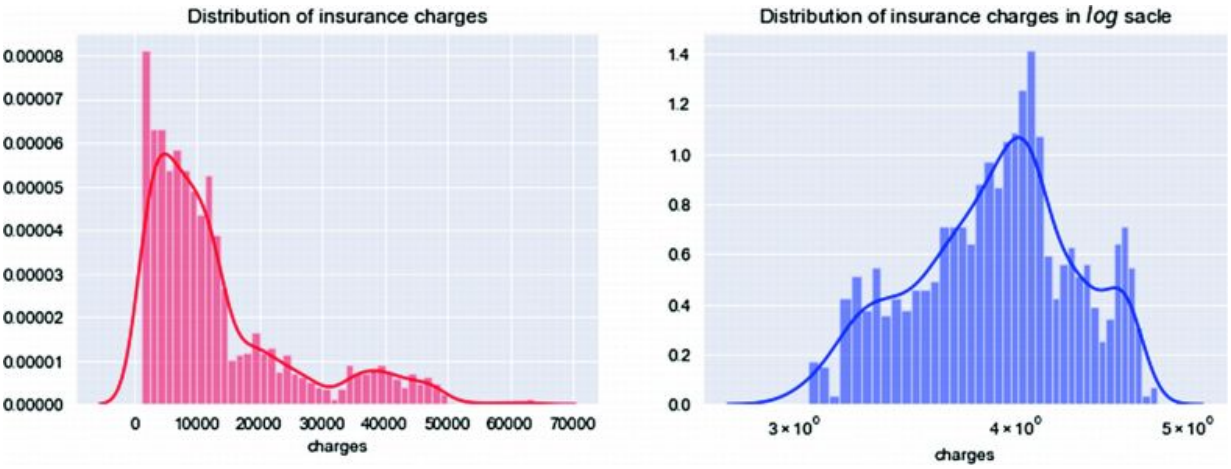

Fig. 1. Source: Shinde et al.

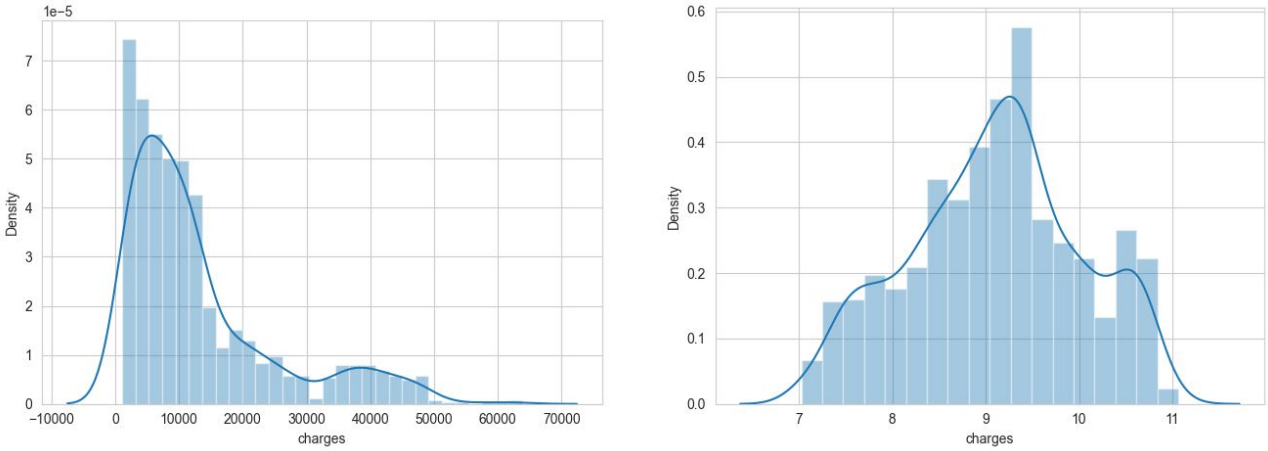

Fig. 2. Reproduced Charts.

Methods

The methods implemented in the study were: multiple linear regression, random forest regression, XGBoost (Improved gradient boosting), support vector regression, and a deep learning neural network. Of these methods, the only one not reproduced in this project was the deep learning model as it was difficult to implement/figure out in the given time period along with no information being provided in the paper.

The major issue with this study that made the results unreproducible was the lack of explanation in the methods section. When implementing the models, a random state value is needed which acts as a permutation seed that Scikit-learn uses to generate the splits. Because this was not provided, it is not possible to match the exact results of the initial model. In addition, the methods section only makes reference to the train/test split values used, and the methods of comparison (RMSE, R^2) which does not give me any further information as to how they fine tuned parameters, what type of ANN was used, etc. Attempts to adjust parameters, use common random state values (0, 1, 42), and additional reading into different approaches for implementation were explored but none of these were successful at reproducing the exact results.

Evaluation and Interpretation of Findings

The models were compared using RMSE and R-squared values which can be seen in Fig. 3. The R-squared results from this project's models more or less aligned with the study's findings however there was a major discrepancy between the support vector regression values.

Because there was no mention of cross validation in the paper, it did not initially seem necessary to implement it. Once finding the SVR result to be much higher than the original, re-running the code with the addition of cross validation was done which reduced the R-squared result to 83.29. After this finding, cross validation was applied to the remaining methods but did not produce results any closer to the initial values. For example, when cross validation was applied to the random forest regressor, the R-squared value was 83.50 which is less accurate than what was originally achieved. Further tuning of the SVR method might yield better results but given the time constraints and lack of information from the study, this project's implementation seemed to be the best that could be done.

The RMSE values in the initial study were much lower than what was found in this project's results. Since the range of charges vary by thousands, a RMSE value of .3377 (vs. .0925) would not ultimately affect the predicted charges enough to matter, giving us a good line of best fit.

Overall, the random forest regressor performed the best which agrees with the theory since it is able to learn and achieve high performance on complex datasets with non-linear relationships. This method is the most on par performance-wise with a neural network and is easy to implement/train though it should be noted that training decision trees can lead to overfitting. To further improve on this work, implementing a deep neural network, attempting to fine tune the parameters, or exploring larger datasets would lead to more insight into which methods would be most applicable.

	RMSE		R-Squared	
Algorithm Used	Paper	Project	Paper	Project
Multiple Linear Regression	0.0925	0.3377	78.35	79.03
Support Vector Regression	0.0965	0.3222	76.44	87.64
Random Forest Regressor	0.0737	0.2896	86.26	86.41
XGBoost	0.0694	0.3448	87.83	85.84

Fig. 3. Comparison of Results (Paper source: Shinde et al.)

References

Kaggle Medical Cost Personal Datasets. Kaggle Inc. https://www.kaggle.com/mirichoi0218/insurance

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

Shinde A., Raut P. (2020) Comparative Study of Regression Models and Deep Learning Models for Insurance Cost Prediction. In: Abraham A., Cherukuri A., Melin P., Gandhi N. (eds) Intelligent Systems Design and Applications. ISDA 2018 2018. Advances in Intelligent Systems and Computing, vol 940. Springer, Cham. https://doi.org/10.1007/978-3-030-16657-1_103