Predicting insurance charges with machine learning

Abelhard Jauwena*, Olusegun Odumosu*, Nishita Sharif*

*Department of Animal Biosciences

*University of Guelph, 50 Stone Rd E, Guelph, ON N1G 2W1

Abstract

The aim of this project is to predict health insurance premium charges based on several different factors using three regression-based (i.e., the Support Vector Regressor, Adaptive Boosting Regressor, and *k*-Nearest Neighbors Regressor) and three classification-based (i.e., the Support Vector Classifier, Adaptive Boosting Classifier, and *k*-Nearest Neighbors Classifier) machine learning models in Python. Additionally, this investigation sought to identify which factors contribute to an individual's insurance premium charges. After thorough processing, we used the data to train all six models and tested the models on unseen test data. The Support Vector Classifier performed the classification task best according to accuracy and F1 scores, while the AdaBoost Regressor performed the regression task best according to RMSE and R2 scores. Age and smoker status were consistently identified as the most important features for determining insurance premium data charges. The results of this investigation may be used to inform best practices when constructing machine learning models in the future and illuminate what lifestyle practices individuals can modify to reduce their insurance expenses.

Keywords:

Health insurance premiums, Machine learning models, *k*-Nearest Neighbors, AdaBoost Regressor, AdaBoost Classifier, Support Vector Machine.

1. Introduction

The costs of health insurance premiums have been reported to be on the rise, whereas insurance coverage has been declining in recent years (Chernew et al., 2005). In this paper, we aimed to predict the health insurance charges using data from the Insurance Premium Data on Kaggle (Jain, 2020). We created and compared different regressors and classifiers to predict the premium charges and the tier of the premium charges respectively. The dataset includes commonly used factors to determine premium charges, namely age, BMI, number of children, sex, smoking status, and region of residence.

Several studies have explored the use of machine learning techniques to predict outcomes that financially affect patients. For instance, a recent study by Rakshit et al. (2021) demonstrated that machine learning models could accurately predict healthcare costs of breast cancer patients. Similarly, a study by Mahmoudi et al. (2020) demonstrated the use of machine learning models to predict the likelihood of hospitalization and readmission rates in patients.

We compared the performance of six machine learning models, including the Support Vector Regressor (SVR), Adaptive Boosting (AdaBoost) Regressor, and *k*-Nearest Neighbors (KNN) Regressor for regression models, and the Support Vector Classifier (SVC), AdaBoost Classifier, and KNN Classifier for classification models. We also identified the most important factors for all the models. The study is significant because the results obtained from it can help healthcare providers and individuals to better understand the factors that influence the premium charges. Additionally, this study could be of use in identifying patients at risk of higher healthcare costs to appropriately allocate resources and develop strategies to pre-empt such financial burdens.

2. Methods

Obtaining and Visualizing the Data

We obtained the dataset from a Kaggle post by Jain (2020). The dataset included records of seven variables obtained from 1,338 policyholders to whom insurance premiums were charged. These variables include age (18-64), sex (female and male), body mass index (BMI; 16.0-53.1), the number of children (0-5), smoking status ("yes" and "no" for smoker and non-smoker, respectively), region of residence ("northeast," "northwest," "southeast," and "southwest"), and the insurance premium amount charged to the policyholder (hereinafter, "insurance premium charges").

We created various plots to visualize the distributions for each variable. Specifically, we created histograms to visualize the distributions for age, the number of children, BMI, and insurance premium charges, while we created pie charts to visualize the distributions for sex, smoking status, and region of residence (**Figure 1**).

Processing the Data

We first checked the dataset for any missing entries, which were not present. Afterward, we converted the entries for each variable into appropriate data types (e.g., we converted the entries for BMI into floats to accommodate for fractions in decimal form).

The entries for sex, region of residence, and insurance premium charges also underwent further processing. Firstly, we label encoded the entries for sex and smoking status. For the entries for sex, we represented "female" as "0" and "male" as "1." Similarly, for the entries for smoking status, we represented a non-smoker as “0” and a smoker as “1.” We also one-hot encoded the entries for region of residence. Lastly, we divided the entries for insurance premium charges into three classes, which include the classes “Low,” “Medium,” and “High.” We created these classes by dividing the data using tertiles such that each one of the classes “Low,” “Medium,” and “High” contains approximately 33.3% of the data. Ultimately, these classes serve as the target output variables for our classification models.

We detected outliers in the dataset with the help of a KNN clustering model, where we instructed the model to find the three nearest neighbors for each data point (i.e., we set the “*k*” value to three). Specifically, we fitted the KNN clustering model to the entries for insurance premium charges, then used it to identify the three nearest neighbours for each data point and calculate the mean distance between them. We visualized the mean distances of the three nearest neighbours for each data point using a bar plot. Subsequently, we set the cutoff value for the mean distance to 750, and as such we removed data points whose mean distance exceeds this value (**Figure 2**). We intentionally did not set the cutoff value too low to retain some entries corresponding to high insurance premium charges in case they correlate with any of the other variables.

Splitting the Data

We first shuffled the entries in the dataset before we divide them into training and testing sets using a 70:30 train-to-test split. We split the dataset using stratified sampling, where we used the entries for smoking status and insurance premium charge classes to form the strata. The reason for using these two variables to stratify the dataset is that they, unlike the remaining variables, are unevenly distributed. As a result, the resulting strata should represent the dataset more proportionally.

Chosen Evaluation Measures

We chose appropriate evaluation measures to test the performance of all six models. Specifically, we used R^2 and RMSE scores to evaluate the performance of the SVR, AdaBoost Regressor, and KNN Regressor, and we used accuracy and F1 scores to evaluate the performance of the SVC, AdaBoost Classifier, and KNN Classifier.

Obtaining Each Model’s Baseline Performance

Before optimizing our models, we obtained their baseline performance for benchmarking purposes. Except for the KNN Regressor and KNN Classifier, we fitted each model to the training set using their respective default hyperparameter settings and evaluated the accuracy of their predictions on the testing set. We did not evaluate the baseline performance of the two KNN models because these models require that the user specify the *k* value before running them. As such, we used the performances of the two KNN models after hyperparameter tuning as benchmarks instead.

Tuning Each Model's Hyperparameters

SVM Models

We used the function "RandomizedSearchCV" to tune the hyperparameters for both the SVR and SVC. Specifically, for the SVR, we tuned for the "C," "kernel," "degree," "coef0," "epsilon," "shrinking," and "tol" hyperparameters (*sklearn.svm.SVC.*, n.d.). C is a regularization parameter that maintains balance between training and testing errors – therefore, a larger C value will result in a narrower balance between the training and testing errors and will penalize misclassifications and/or high prediction errors (*sklearn.svm.SVC.*, n.d.). Kernel transforms data into higher dimensions such that the data can be separated according to its characteristics. Degree is used in polynomial kernel functions and represents the complexity of the model. Coef0 is used in polynomial and sigmoid kernel functions and controls the constant term in the model. Epsilon defines the margin of error for the model – larger epsilon values allow for more deviation. Shrinking determines the usage of the shrinking heuristic; when shrinking is set to "True," the model will remove samples that are a great distance away from the decision boundaries. Lastly, tol represents the model's tolerance for stopping.

For the SVC, we tuned for the same parameters as we did for the SVR using RandomizedSearchCV. Additionally, we also tuned for the "decision_shape_function" hyperparameter, which has two possible options. The first option – the "One-vs-Rest" ("OVR") approach – creates binary classifications where one class is compared to the rest of the dataset (*sklearn.svm.SVR.*, n.d.). Conversely, the second option – the "One-vs-One" ("OVO") approach – creates binary classifications where one class is compared to another class. In the end, RandomizedSearchCV chose the OVO approach for decision_shape_function.

AdaBoost Models

We used the function "BayesSearchCV" to tune the hyperparameters for the AdaBoost Regressor and AdaBoost Classifier. For the AdaBoost Regressor, we tuned for the number of estimators, the learning rate and the loss function. The number of estimators is the maximum number of estimators at which boosting is stopped (*sklearn.ensemble.AdaBoostRegressor*., n.d.). The learning rate is the weight applied to each regressor. The loss function is the magnitude at which the weights are updated, they include linear, square and exponential(*sklearn.ensemble.AdaBoostRegressor*., n.d.). Conversely, for the AdaBoost Classifier, we tuned for the number of estimators, the learning rate, and the algorithm (i.e., one of "SAMME" or "SAMME.R"). SAMME.R is a variant of SAMME that uses a different approach to calculate the weight of each weak classifier. Instead of minimizing the exponential loss, SAMME.R minimizes a regularized version of the exponential loss, which helps to prevent overfitting (*sklearn.ensemble.AdaBoostClassifier.,* n.d.).

KNN Models

To find the best *k* value to use for the KNN Regressor, we fitted 50 different KNN Regressors to the dataset, each with a *k* value ranging from one to 50. We then obtained the R^2 and RMSE scores for each KNN Regressor and plotted them on a line plot. The KNN Regressor that yielded the highest R^2 score and lowest RMSE score had a *k* value of six. As such, we built our final KNN Regressor using a *k* value of six. Similarly, we also fitted 50 different KNN Classifiers to the dataset, each with a set *k* value ranging from one to 50. Afterward, we obtained the mean prediction error, accuracy score, and F1 score for each KNN Classifier and plotted them on a line plot. The KNN Classifier that yielded the lowest mean prediction error and the highest accuracy and F1 scores had a *k* value of five. Therefore, we built our final KNN Classifier using a *k* value of five.

Unfortunately, we did not evaluate the performances of both the KNN Regressor and KNN Classifier using different distance measures. Considering that the choice of distance measure can significantly impact the performance of KNN models, future studies should investigate the effects of different distance measures on the performance of KNN models (Alfeilat et al., 2019).

Selecting the Most Important Features for Each Model

In addition to hyperparameter tuning, we also obtained the most important features for each model (i.e., the features that the model relied most heavily on to make predictions) using the "permutation feature importance" technique. This technique works by iteratively shuffling the data for each input variable while keeping the data in the other input variables constant (*4.2. Permutation*, n.d.). For each iteration, it then calculates the drop in the model's performance in making accurate predictions for the output variable (*4.2. Permutation*, n.d.). As such, the most important feature is the one that causes the largest drop in the model's performance (*4.2. Permutation*, n.d.).

We included only the five most important features for each model as input variables in their final versions.

3. Results

Model Evaluations

After conducting hyperparameter tuning for KNN, we determined that a *k* value of six was optimal for the KNN Regressor, while a *k* value of five was optimal for the KNN Classifier (refer to **Figure 3** and **Figure 4**). The selected *k* values produced consistent results across all evaluation metrics. As shown in **Table 1** and **Table 2**, all models that underwent feature selection and hyperparameter tuning performed substantially better than the default model.

We used several evaluation metrics to assess the performance of the models. R^2 measures the proportion of variance explained by a regression model, while RMSE measures the average deviation of predicted values from observed values in a regression model (Chicco et al., 2021). Accuracy measures the proportion of correctly classified instances in a classification model, and F1 score measures the harmonic mean of precision and recall in a classification model (Hossin & Sulaiman, 2015). As suggested by Chicco et al. (2021), we used R^2 values for discussion as they are more informative than RMSE. Furthermore, the R^2 and RMSE scores were consistent for all models (refer to **Table 1** and **Table 2**).

The R^2 values for the default regression models were -0.1017, 0.8247, and 0.7565, while the R^2 values for the models that underwent feature selection and hyperparameter tuning were 0.7134, 0.8564, and 0.8099 for SVR, AdaBoost Regressor, and KNN Regressor respectively. For the best model, both accuracy and F1 scores agreed, and we opted to use the accuracy score for discussion. The accuracy scores for the default classification models were 0.8404, 0.7903, and 0.8443, while the accuracy scores for the final models were 0.8975, 0.8607, and 0.8475 for SVC, AdaBoost Classifier, and KNN Classifier respectively (refer to **Table 1** and **Table 2**).

Permutation feature importance analysis revealed that age and smoker (smoking status) were consistently considered among the top three most important features across all models (refer to **Figures 5-7** and **Table 3**). Age was ranked as the most important feature for four models, while smoker was considered the most important feature for two models.

Overfitting and underfitting considerations

Learning curves were plotted throughout the development of the models to evaluate overfitting and underfitting (**Figures 8-13**). The SVR model showed significant improvement in performance after feature selection and hyperparameter tuning, as seen in **Figure 8**, with the average R2 score increasing from approximately -0.11 to approximately +0.675. However, the final model appeared to be underfitting slightly, as both the training and cross-validation scores continued to trend upwards. The AdaBoost Regressor also showed improvement in performance from the default model, with the final model exhibiting slight overfitting as shown in **Figure 9**. The training and validation scores started to diverge at higher training examples. The KNN Regressor did not exhibit improvement in either overfitting or underfitting, with both the default and final models showing signs of overfitting and underfitting, as depicted in **Figure 10**. All SVC models exhibited both overfitting and underfitting, with the scores trending upwards and the gap between the training and validation scores remaining significant (**Figure 11**). The AdaBoost Classifier showed relatively good performance in terms of overfitting but was underfitting (**Figure 12**). Similarly, the KNN Classifier showed signs of overfitting and underfitting for both the default and final models (**Figure 13**). All models demonstrated some degree of underfitting or overfitting, and future studies could consider using more data or feature engineering to address these issues.

Predictions

To assess the performance of the regression models, Lin's Concordance Correlation Coefficient (CCC) and R2 were used. Lin's CCC measures both precision and accuracy of a set of data compared to a "gold standard" and ranges from -1 to +1, where a value of 1 indicates a perfect fit (Akoglu, 2018). For the SVR model, the default and final models had CCC values of 0.0032 and 0.8281, respectively, and R2 values of -0.1017 and 0.7134, respectively (**Figure 14**). Both CCC and R2 improved for the final model. The AdaBoost Regressor model had a higher R2 with selected features (0.8210 vs 0.8201), but a lower CCC (0.9029 vs 0.9030), indicating that the model with selected features explained more variance but was less precise and accurate (**Figure 15**). KNN Regressor showed improvement for the final model, with a higher CCC (0.8954 vs 0.8959) and a higher R2 (0.81 vs 0.76) (**Figure 16**).

For the classification models, confusion matrices were plotted to evaluate performance, as shown in Figures 17-19 . The figures indicated that the low and medium classes were predicted correctly more often than the high classes, which may be due to the way the charge classes were binned. The low, medium, and high charge classes were distributed roughly evenly into three, without taking into account the charges themselves. Thus, the "High" charges were merely the 33.3% of the records with the highest charges. This type of binning does not consider the distribution of the charges, but it helps prevent class imbalance.

Overall, the best model for predicting insurance premium charges was the AdaBoost Regressor, with an R2 of 0.8564 and an RMSE of 4249.3268 (**Table 1 and Table 2)**. The best model for predicting the tier of insurance premium was the SVC, with an accuracy of 0.8975 and an F1 score of 0.8963 (**Table 1 and Table 2**).

4. Discussion and Conclusion

Our models identified age and smoking status as the factors that have the most significant impact on the amount of insurance premium charged to an individual. This finding aligns with the fact that age and smoking status have been strongly associated with increased healthcare-related expenses (Alemayehu & Warner, 2004; Izumi et al., 2001). A cross-sectional study conducted in Michigan found that one-third of an individual's lifetime healthcare expenses are spent in middle age, while almost one-half is spent in an individual's senior years (Alemayehu & Warner, 2004). Additionally, for those who regularly smoke tobacco, healthcare expenses can increase by up to 4-11% (Izumi et al., 2001). As such, it is unsurprising that these factors were consistently identified as significant determinants of an individual's insurance premium charges.

Throughout the construction of the models in this investigation, it became apparent that there are discernible differences between the models, their functionalities, and the data for which they are most effective. The default SVR returned a negative R^2 score when predicting the output of the test data; it predicted worse than the mean value of the charges. However, after feature selection and hyperparameter tuning, it performed significantly better when predicting the output of the test data, with an R^2 of 0.7134. Comparing the default and final SVR illustrates the sensitivity of the SVR to its hyperparameters. If hyperparameters are not tuned properly and thoroughly, SVRs may suffer from either overfitting or underfitting, making them unsuitable for a given dataset (Jiang et al., 2013). Despite using a virtual machine (hosted by ComputeCanada) for this investigation, when plotting the learning curve for the SVR with all hyperparameters tuned, the code ran indefinitely. When plotting the learning curve for an SVR with the default gamma parameter, the learning curve plotting code ran successfully. As such, the final model, while not tuned with the optimal hyperparameters, was chosen to ensure that the learning curve could be visualized. Had this issue been rectified, and given the SVR's sensitivity to hyperparameters, the final SVR may have performed significantly better.

After exploring all six models, we found that the SVC performed best among the classifiers according to both accuracy and F1 scores, while the AdaBoost Regressor performed the best among the regressors according to both R^2 and RMSE scores. The KNN models did not perform particularly well when performing regression or classification.

The SVC used the OVO approach, which allowed the model to use multiple binary classifications to find the optimal hyperplane between each pair of classes. The hyperparameter tuning step of creating the SVC provides insight into the nature of the relationship between the predictor variables and the output (i.e., the "Low," "Medium," or "High" classes for insurance premium charges). That is, the SVC, with non-linear hyperplanes, performed best due to the data's complex and non-linear nature.

When comparing the regression models created in this investigation, the AdaBoost Regressor performed the best, both before and after modifying the models in accordance with the data. The AdaBoost Regressor is able to combine weak regressors to create a stronger ensemble; this approach lends itself to more effectively identifying the relationship between predictor variables and the output variable. In this investigation, the AdaBoost models used only decision trees. Had other algorithms been used when creating the final ensemble AdaBoost models, the AdaBoost models may have performed even better than was observed here.

The KNN Regressor and KNN Classifier performed suboptimally, which may be a result of the KNN models' rudimentary algorithm. Because the KNN algorithm predicts outcomes based on the proximity of a sample to its neighbours, it is unable to determine complex, non-linear relationships. The simplicity of KNN also exposes the overfitting that was present in both the classification and regression models. Although the KNN models seemed to parse out the relationships between the predictor variables and the output in the training data, it did not extrapolate those relationships to the test data as well as the SVC and AdaBoost models did. Furthermore, the learning curves for the KNN models appeared to reveal some underfitting of the models. Compared to SVC, KNN has been found to be more sensitive to training sizes (Noi & Kappas, 2018). Thus, the underfitting issue could be rectified by larger training dataset sizes.

The distribution of premium charges in this investigation was right-skewed, which likely explains why the models did not predict the test output of all datapoints well, especially at the higher end of the premium charges. Moreover, the data in the "High" class had premium charges that were widely distributed; this variance meant that the classifiers were less effective at identifying the correct class in which the data belonged. Previous studies have shown that, for imbalanced datasets, random oversampling methods can be used to mitigate overfitting in AdaBoost, SVM, and KNN (Rahman et al., 2015). As such, future investigations should explore whether different sampling methods could rectify or further mitigate the overfitting issues faced in the present investigation.

As machine learning and other technology continues to progress and become applicable to various problems in the financial, medical, and scientific fields, it is essential that data scientists take the necessary measures to ensure efficient and effective results. Specifically, machine learning models should be created with the characteristics of the dataset and problem at hand in mind, including the sources, limitations, and quality of the data.

Acknowledgments

We would like to thank Dr. Dan Tulpan for his guidance in this project and his willingness to help. This paper would not have been possible without their influence.

Authors' contributions

All authors improved and contributed to the editing of the manuscript. All authors read and approved the final manuscript.

Disclosures

The authors declare no real or perceived conflicts of interest.

References

Akoglu, H. (2018). User's guide to correlation coefficients. *Turkish Journal of Emergency Medicine*, *18*(3), 91–93. https://doi.org/10.1016/j.tjem.2018.08.001

Alemayehu, B., & Warner, K. E. (2004). The lifetime distribution of health care costs. Health services research, 39(3), 627–642. https://doi.org/10.1111/j.1475-6773.2004.00248.x

Chernew, M., Cutler, D. M., & Keenan, P. S. (2005). Increasing Health Insurance Costs and the Decline in Insurance Coverage. *Health Services Research*, *40*(4), 1021–1039. https://doi.org/10.1111/j.1475-6773.2005.00409.x

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, e623.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, *5*(2), 1.

Izumi, Y., Tsuji, I., Ohkubo, T., Kuwahara, A., Nishino, Y., & Hisamichi, S. (2001). Impact of smoking habit on medical care use and its costs: a prospective observation of National Health Insurance beneficiaries in Japan. International journal of epidemiology, 30(3), 616–623. https://doi.org/10.1093/ije/30.3.616

Jain, S. (2020). *Insurance Premium Data* [Data set]. Kaggle. https://www.kaggle.com/datasets/simranjain17/insurance

Jiang, M., Jiang, S., Zhu, L., Wang, Y., Huang, W., & Zhang, H. (2013). Study on parameter optimization for support vector regression in solving the inverse ECG problem. *Computational and mathematical methods in medicine*, *2013*, 158056. https://doi.org/10.1155/2013/158056

Laxy, M., Stark, R., Peters, A., Hauner, H., Holle, R., & Teuner, C. M. (2017). The Non-Linear Relationship between BMI and Health Care Costs and the Resulting Cost Fraction Attributable to Obesity. International journal of environmental research and public health, 14(9), 984. https://doi.org/10.3390/ijerph14090984

Mahmoudi, E., Kamdar, N., Kim, N., Gonzales, G., Singh, K., & Waljee, A. K. (2020). Use of electronic medical records in development and validation of risk prediction models of hospital readmission: systematic review. *BMJ (Online)*, *369*, m958–m958. https://doi.org/10.1136/bmj.m958

Permutation feature importance. (n.d.). scikit-learn. Retrieved April 20, 2023, from https://scikit-learn.org/stable/modules/permutation_importance.html

Rahman, H. A. A., Wahl, Y. B., He, H., & Bulgiba, A. (2015). Comparisons of ADABOOST, KNN, SVM and Logistic Regression in Classification of Imbalanced Dataset. Communications in Computer and Information Science, 545, 54–64. https://doi.org/10.1007/978-981-287-936-3_6

Rakshit, P., Zaballa, O., Perez, A., Gomez-Inhiesto, E., Acaiturri-Ayesta, M. T., & Lozano, J. A. (2021). A machine learning approach to predict healthcare cost of breast cancer patients. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-91580-x

sklearn.ensemble.AdaBoostRegressor. (n.d.). scikit-learn. Retrieved April 20, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html

sklearn.ensemble.AdaBoostClassifier. (n.d.). scikit-learn. Retrieved April 20, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

sklearn.svm.SVC. (n.d.). scikit-learn. Retrieved April 20, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html *sklearn.svm.SVR.* (n.d.). scikit-learn. Retrieved April 20, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

Thanh Noi, P., & Kappas, M. (2017). Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. Sensors (Basel, Switzerland), 18(1), 18. https://doi.org/10.3390/s18010018

Figures

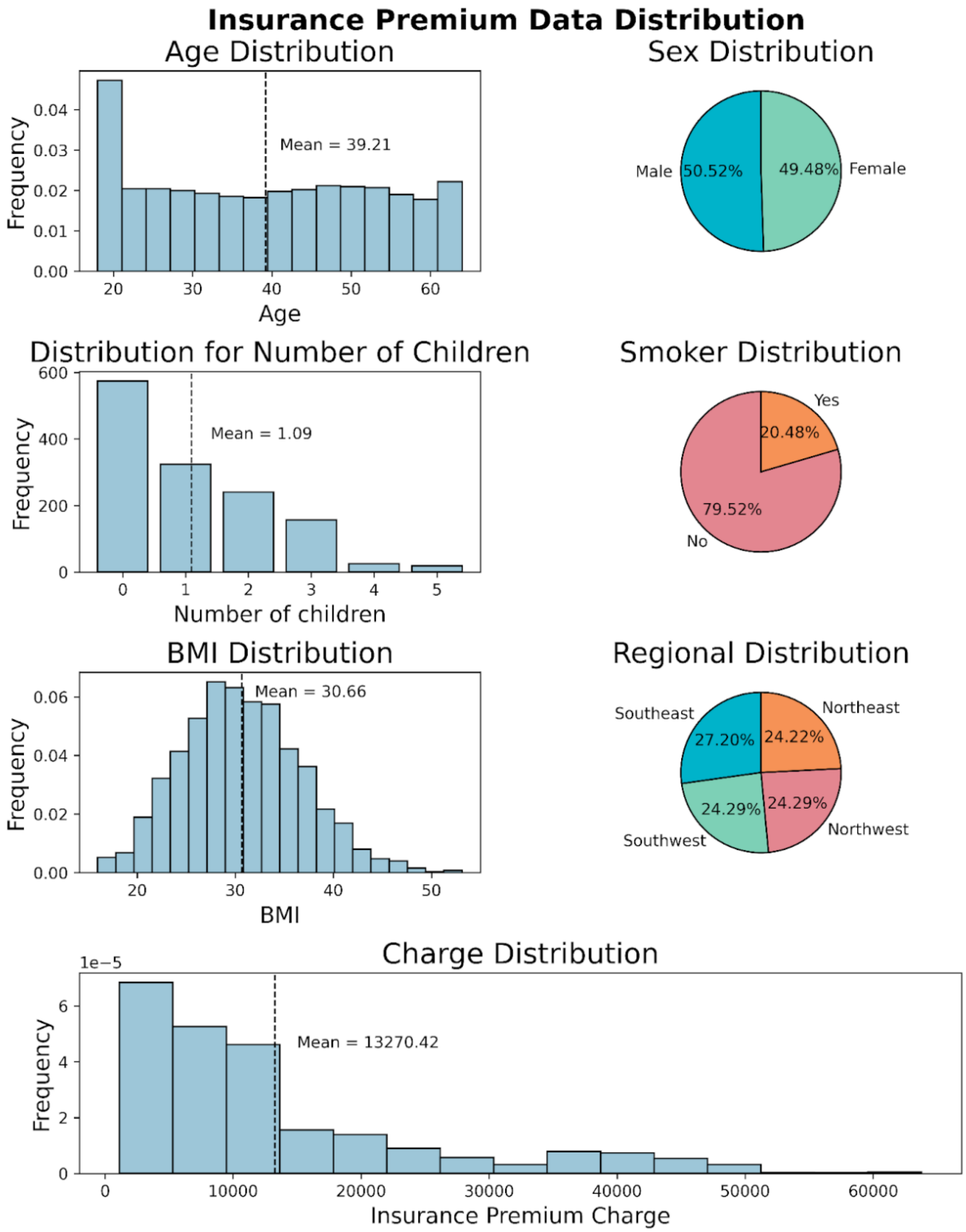

Figure 1. Distribution plots for age, sex, number of children, smoking status, BMI, region, and charges in the dataset.

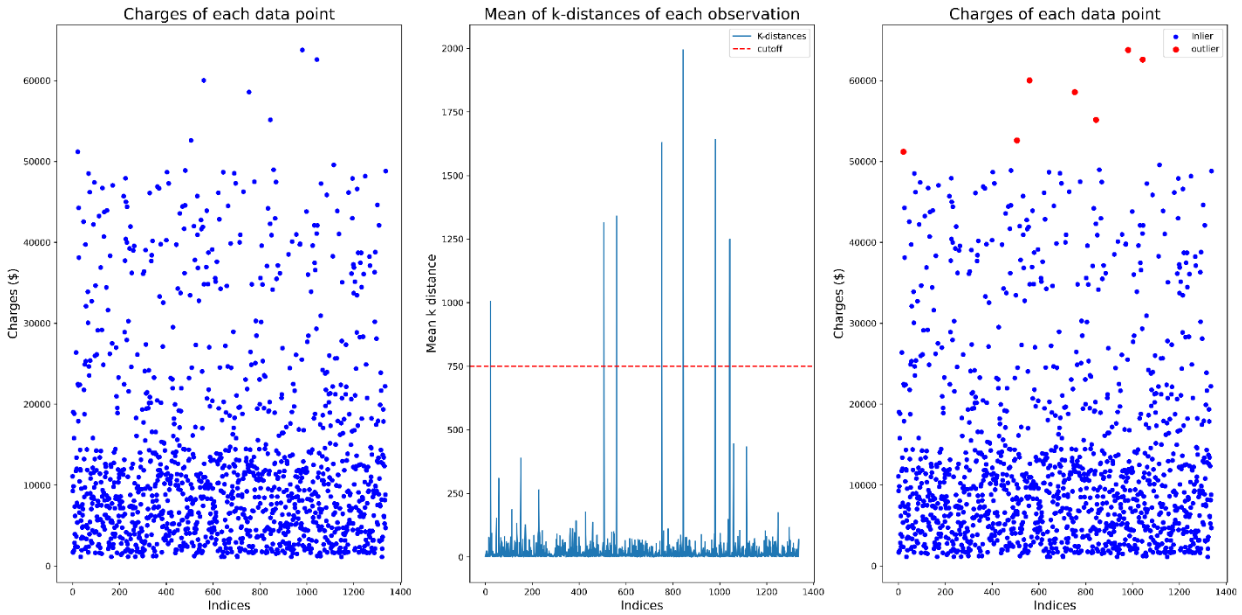

Figure 2. (From left to right) Plots showing the distribution of charges in the dataset, the mean of *k*-distances for each data point obtained using KNN clustering, and the distribution of charges in the dataset with outliers and inliers distinguished.

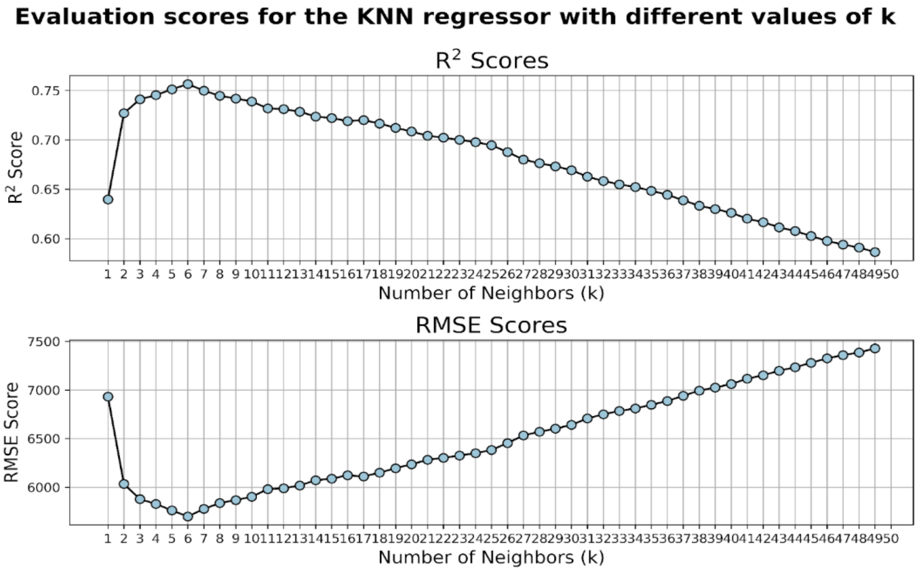

Figure 3. The R^2 and RMSE scores of different KNN Regressors with *k* values ranging from one to 50.

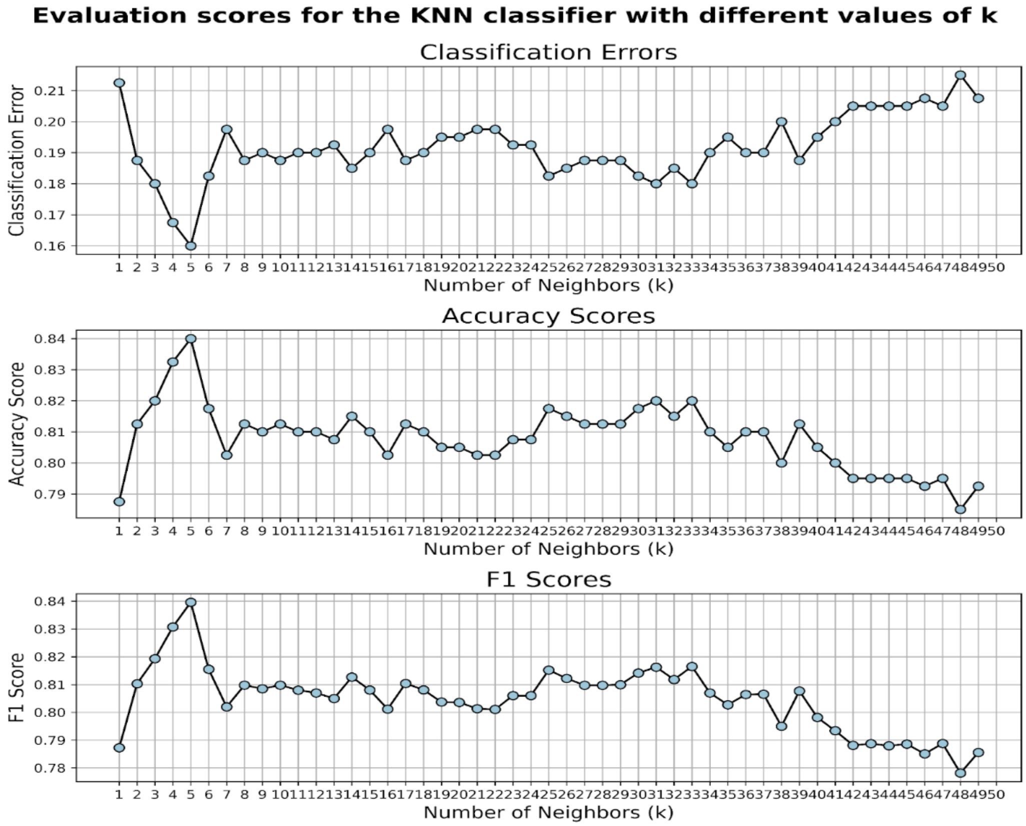

Figure 4. The mean prediction errors as well as the accuracy and F1 scores of different KNN Classifiers with *k* values ranging from one to 50.

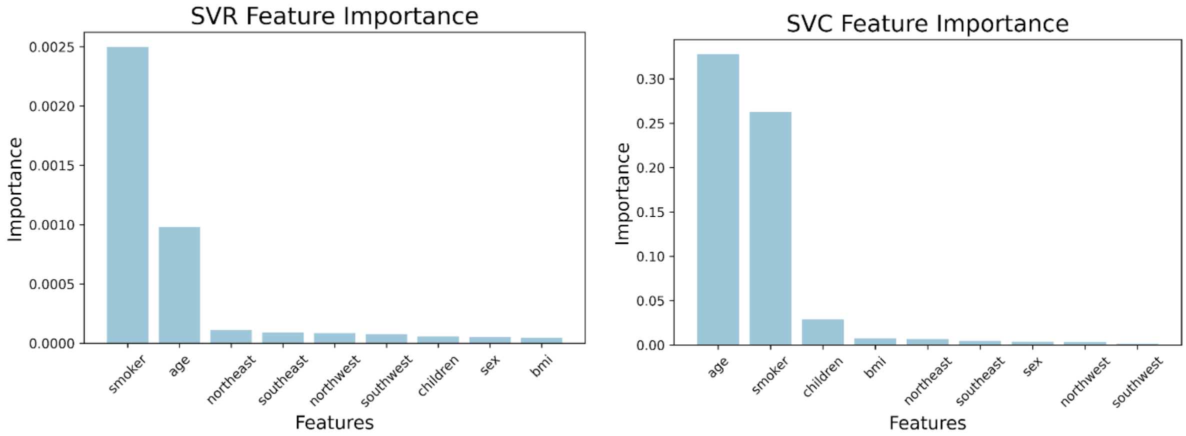

Figure 5. Barplots of the importances of each feature in the dataset for the SVC and SVR. Feature importance scores obtained from permutation feature importance.

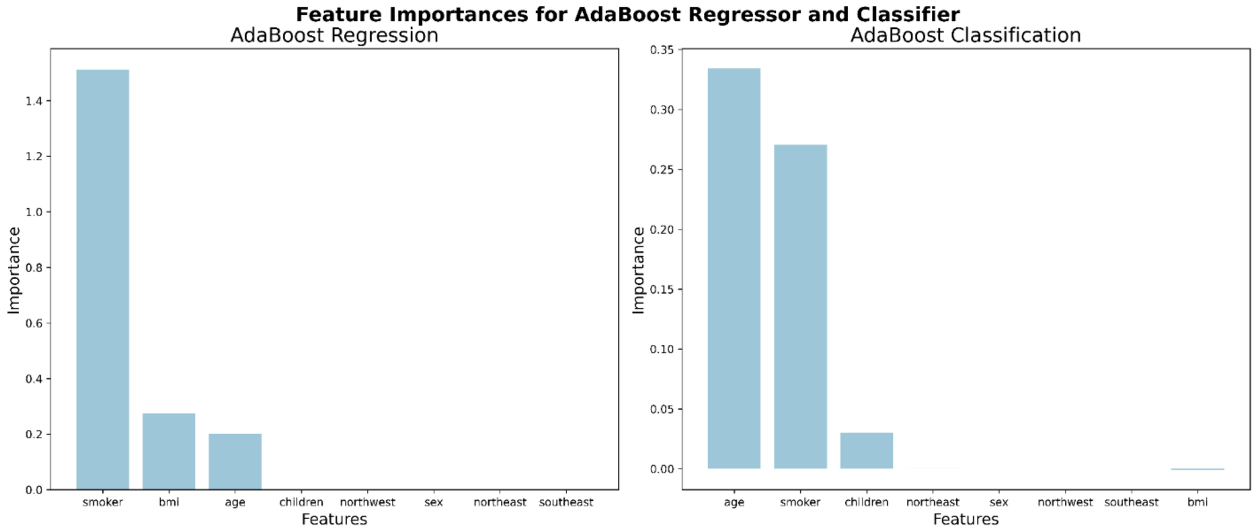

Figure 6. Barplot of the importances of each feature in the dataset for the AdaBoost Regressor and AdaBoost Classifier. Feature importance scores obtained from permutation feature importance.

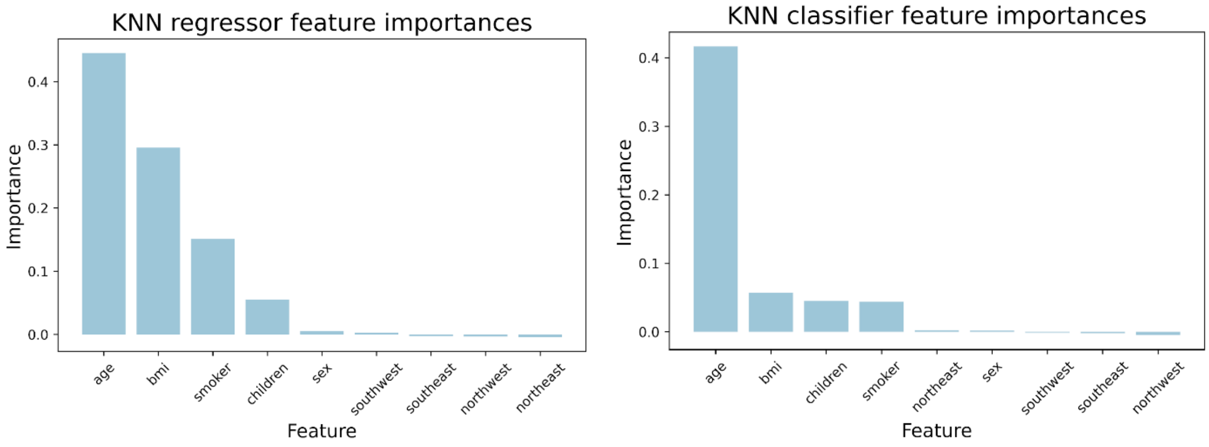

Figure 7. A barplot of the importances of each feature in the dataset for the KNN Regressor and Classifier. Importance scores were obtained using permutation feature importance

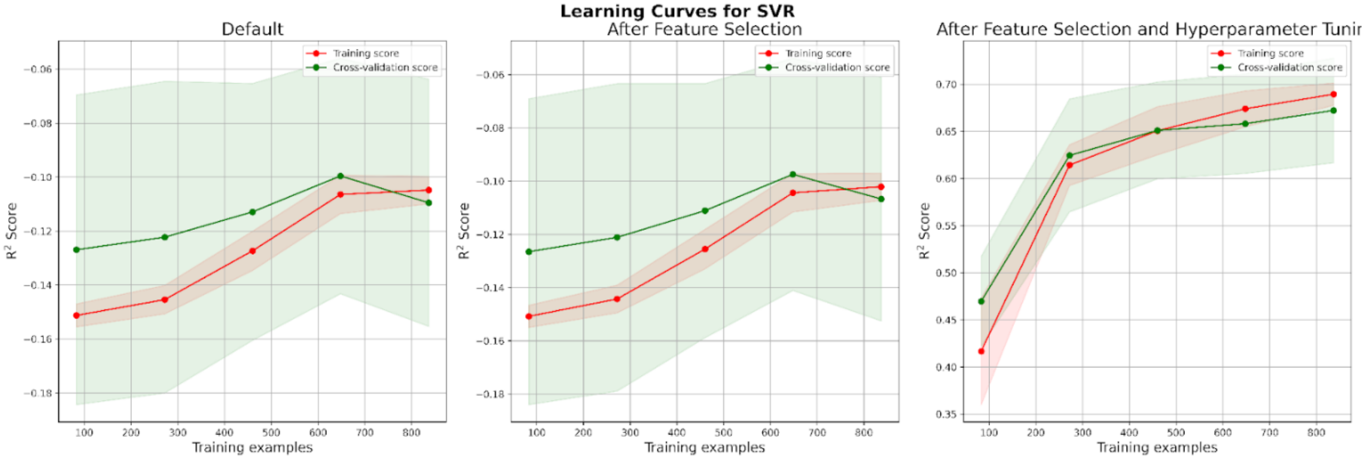

Figure 8. Learning curves evaluated using R^2 for the default SVR, SVR after feature selection, and SVR after both feature selection and hyperparameter tuning.

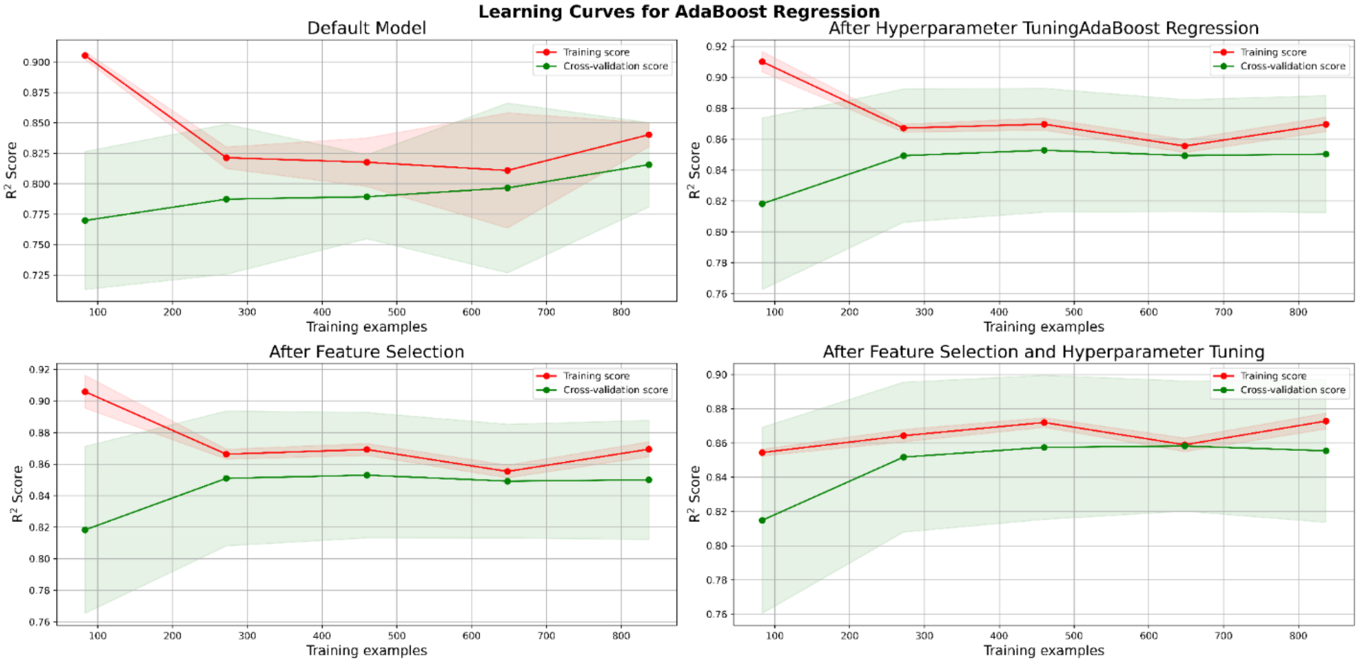

Figure 9. Learning curves evaluated using R^2 for the default AdaBoost Regressor, after hyperparameter tuning with all features, after feature selection, and after both feature selection and hyperparameter tuning.

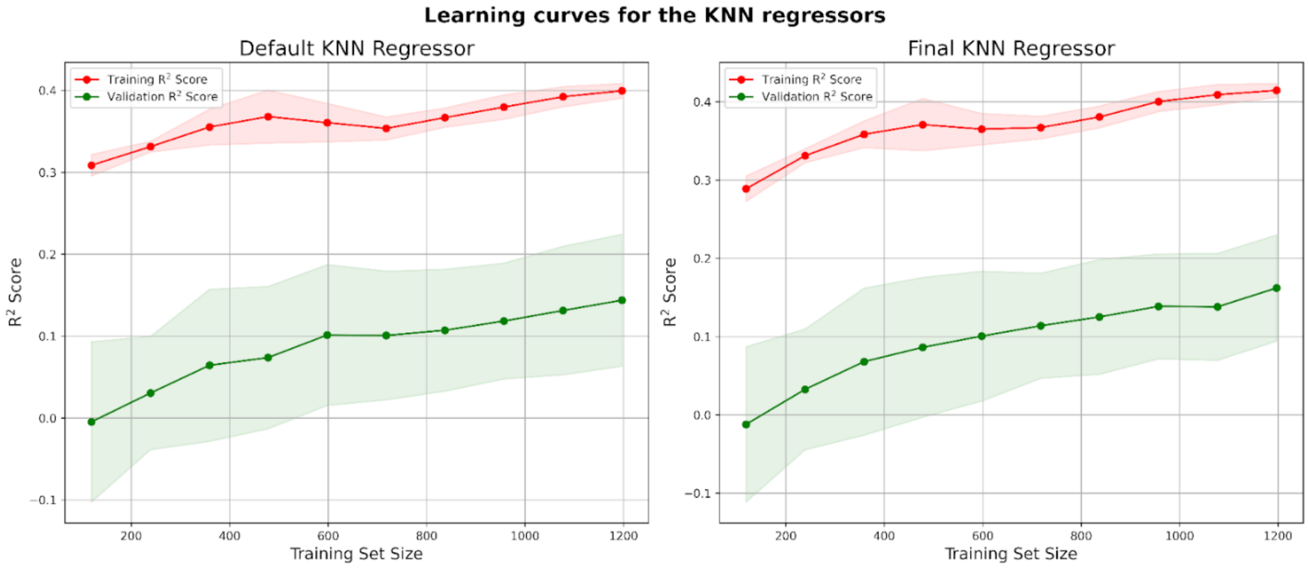

Figure 10. Learning curves evaluated using R^2 for the default and final KNN Regressors. The default KNN Regressor is with the optimal *k* and the final KNN Regressor is with the optimal *k* after feature selection.

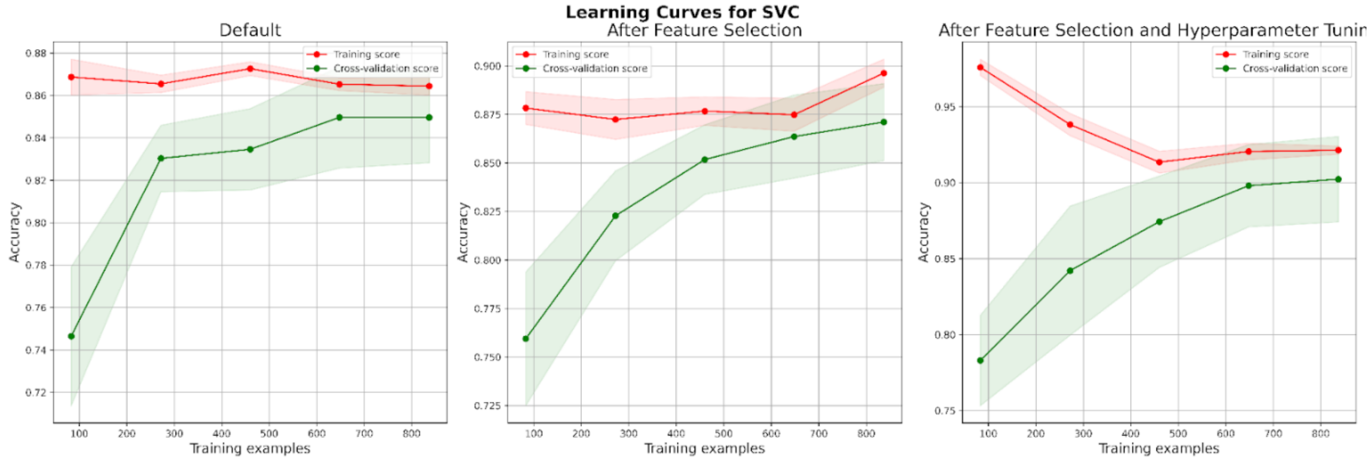

Figure 11. Learning curves evaluated using accuracy for the default SVC, SVC after feature selection, and SVC after both feature selection and hyperparameter tuning.

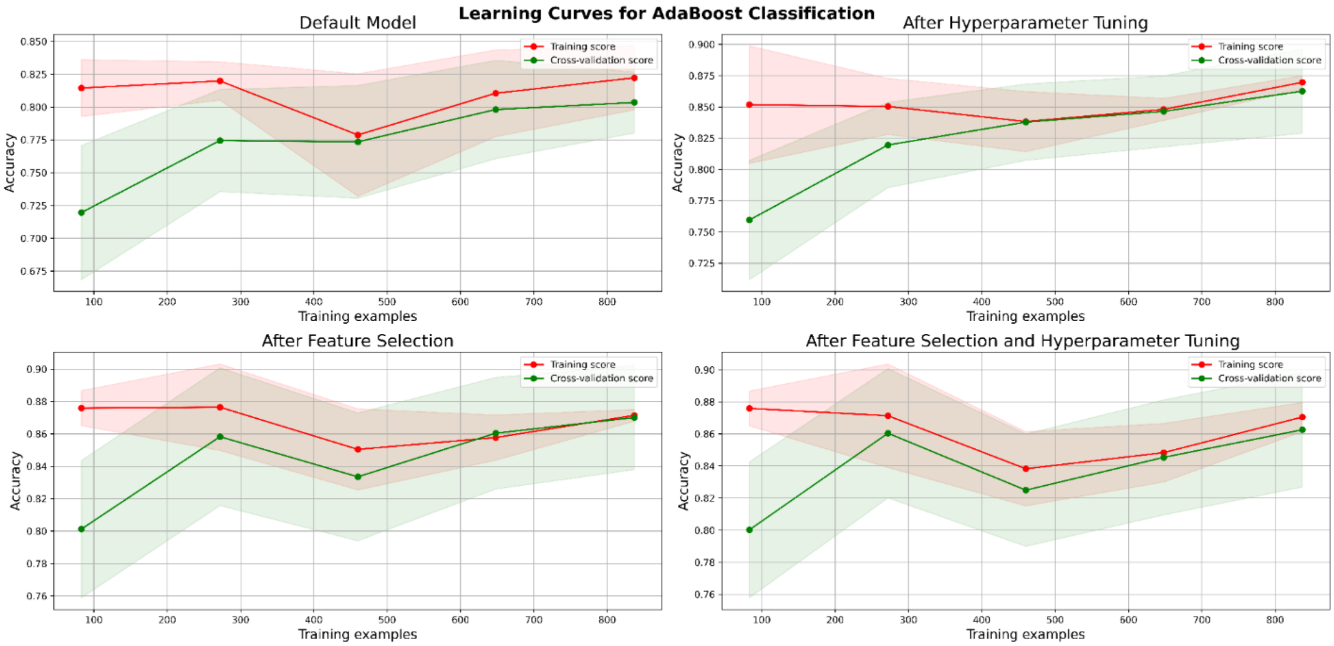

Figure 12. Learning curves evaluated using R^2 for the default AdaBoost Classifier, after hyperparameter tuning with all features, after feature selection, and after both feature selection and hyperparameter tuning.

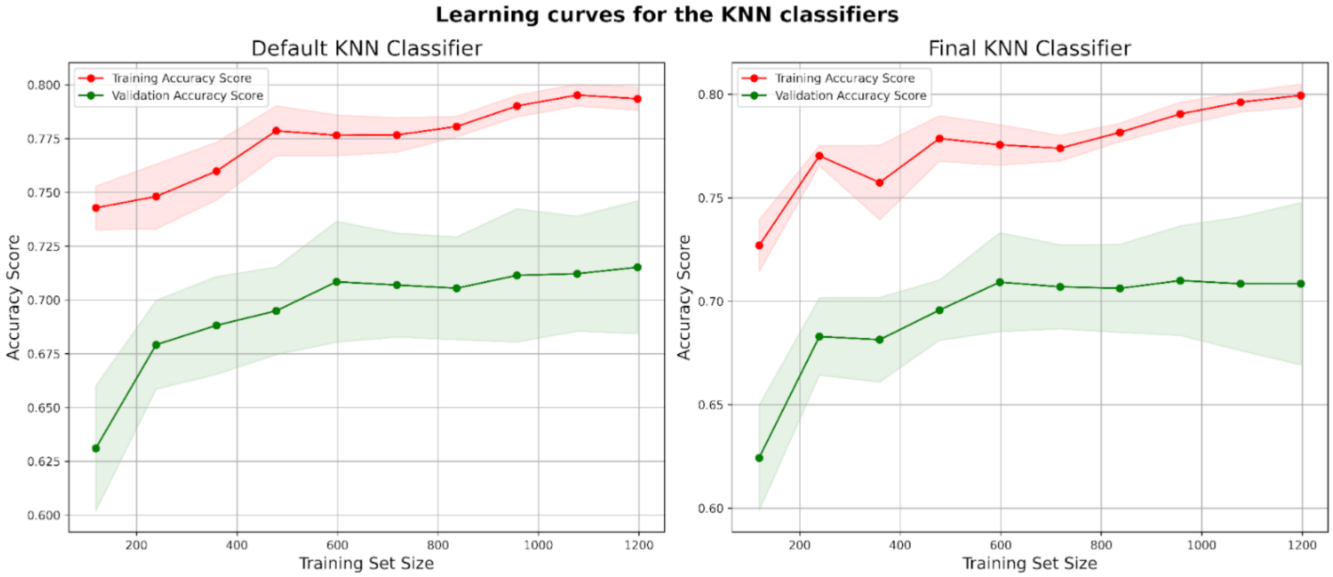

Figure 13. Learning curves evaluated using accuracy for the default and final KNN Classifiers. The default KNN Classifier is with the optimal *k* and the final KNN Classifier is with the optimal *k* after feature selection.

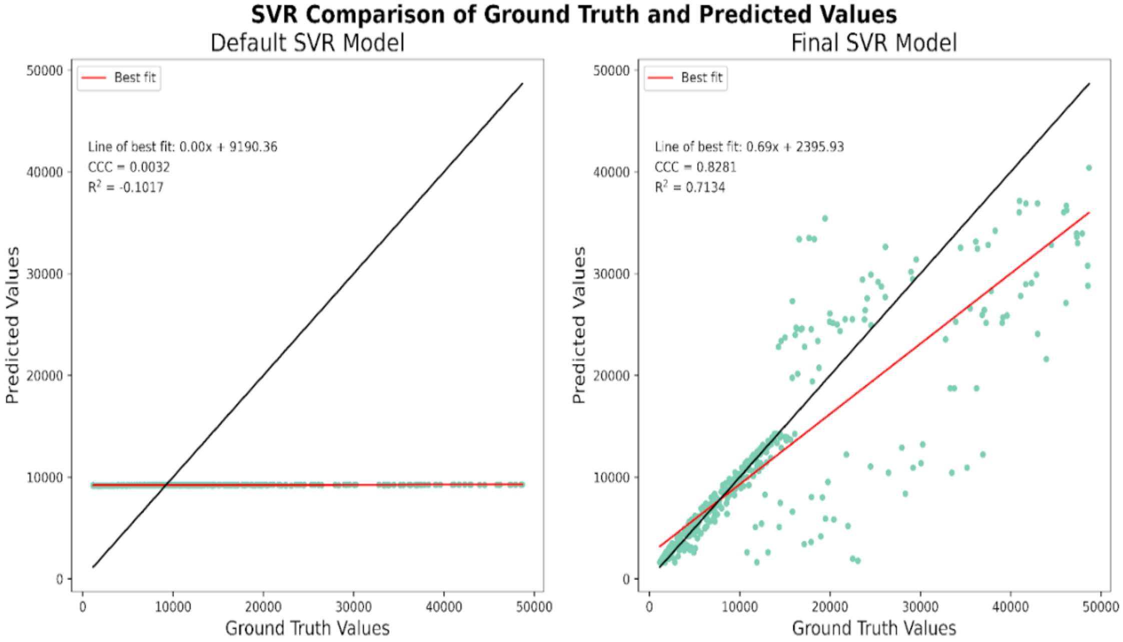

Figure 14. Scatterplots comparing the true values with the predictions outputted by the default and final SVRs.

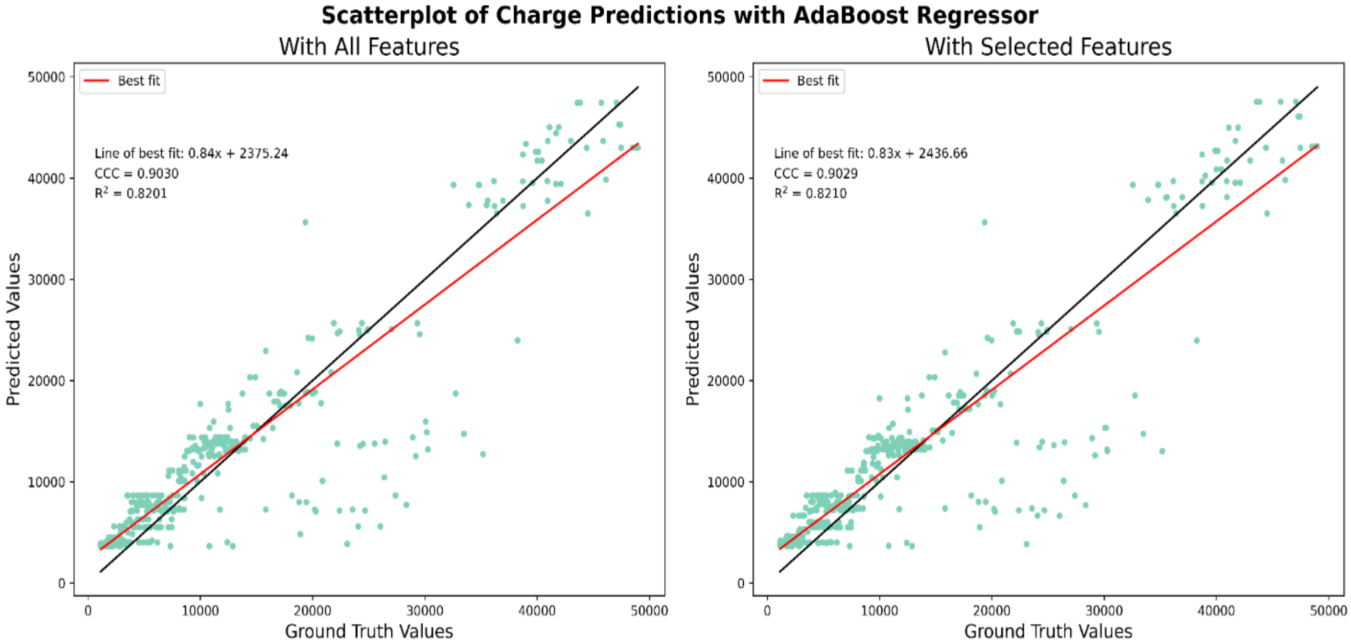

Figure 15. Scatterplots comparing the true values with the predictions outputted by the default and final AdaBoost Regressors.

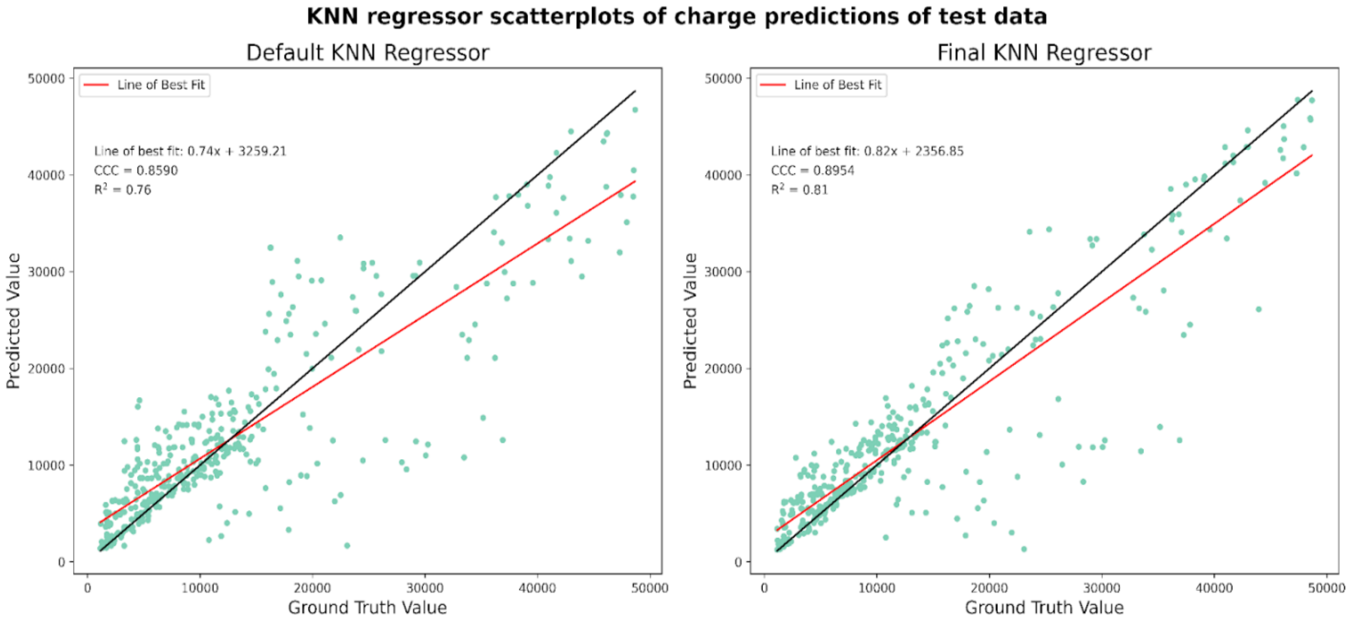

Figure 16. Scatterplots comparing the true values with the predictions outputted by the default and final KNN Regressors.

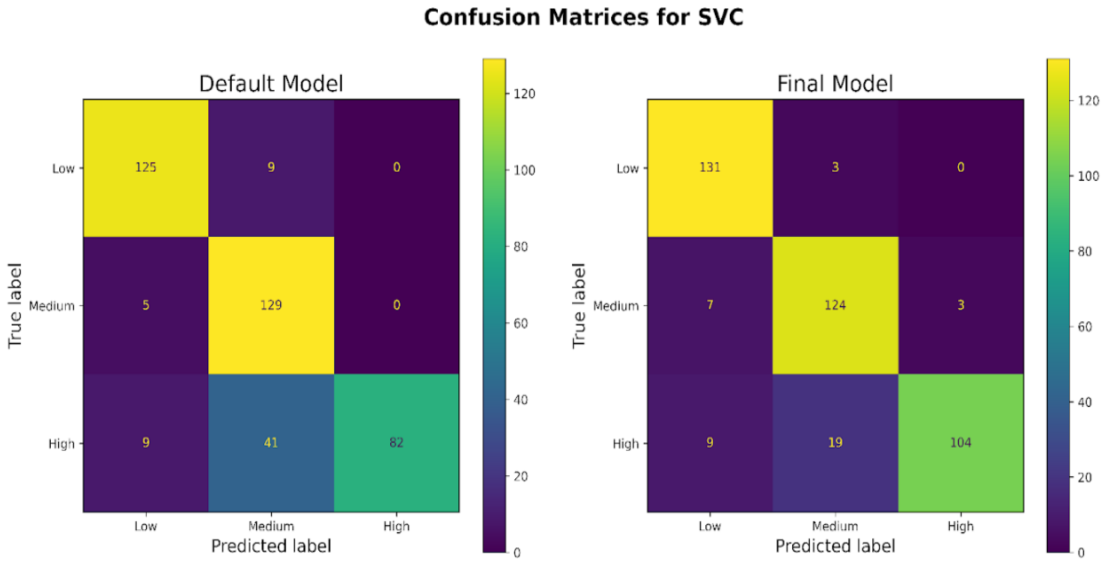

Figure 17. Confusion matrices comparing the true labels with the predictions outputted by the default and final SVCs.

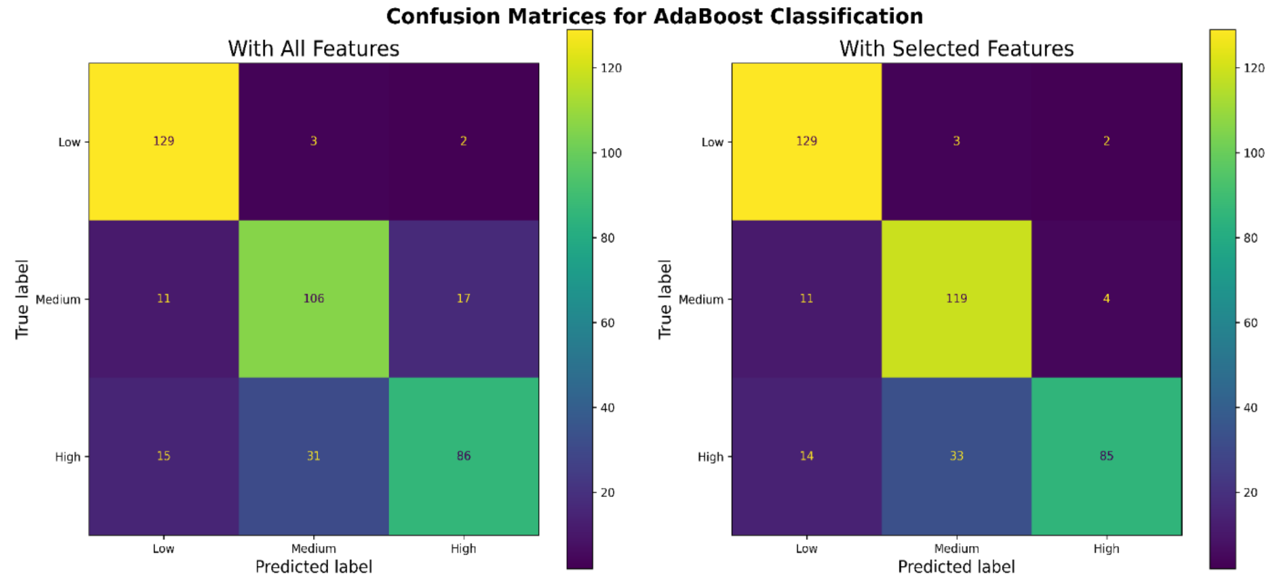

Figure 18. Confusion matrices comparing the true labels with the predictions outputted by the default and final AdaBoost Classifiers.

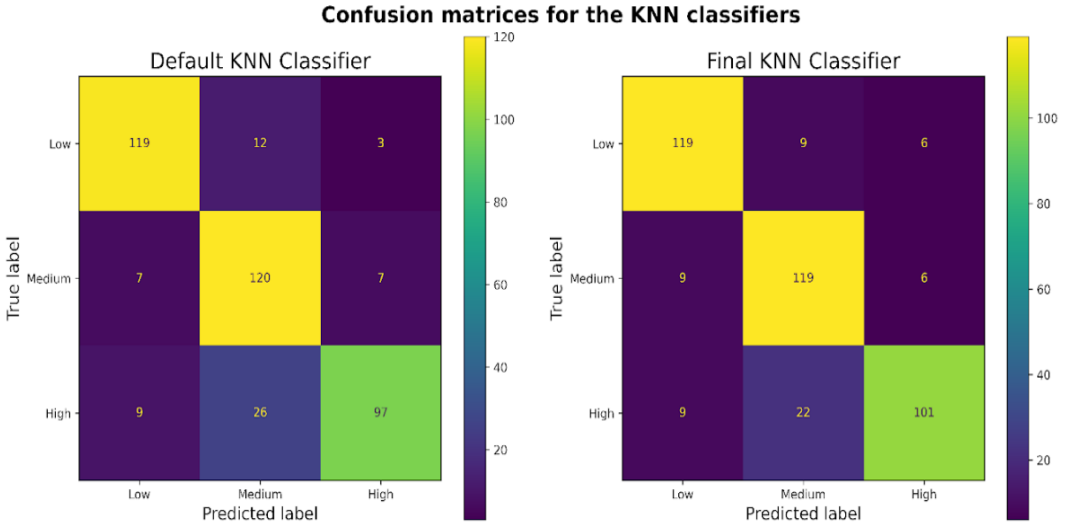

Figure 19. Confusion matrices comparing the true labels with the predictions outputted by the default and final KNN Classifiers.

Tables

Table 1.

	R^2		RMSE	
	Original	Final	Original	Final
Support Vector Regressor	-0.1017	0.7134	12122.0663	6182.7467
AdaBoost Regressor	**0.8247**	**0.8564**	**4747.6980**	**4249.3268**
KNN Regressor	0.7565	0.8099	5698.8685	5035.8842

The R^2 and RMSE scores for both the default and final versions of the SVR, AdaBoost Regressor, and KNN Regressor. Bolded scores indicate the best score for each evaluation measure at each step.

Table 2.

	Accuracy		F1	
	Original	Final	Original	Final
Support Vector Classifier	0.8404	**0.8975**	0.8358	**0.8963**
AdaBoost Classifier	0.7903	0.8607	0.7837	0.8555
KNN Classifier	**0.8443**	0.8475	**0.8396**	0.8470

The accuracy and F1 scores for the default and final modules versions of the SVC, AdaBoost Classifier, and KNN Classifier. Bolded scores indicate the best score for each evaluation measure at each step.

Table 3.

	SVR	SVC	AdaBoost Regressor	AdaBoost Classifier	KNN Regressor	KNN Classifier
Feature 1	Smoker	Age	Smoker	Age	Age	Age
Feature 2	Age	Smoker	BMI	Smoker	BMI	BMI
Feature 3	Northeast	Children	Age	Children	Smoker	Children
Feature 4	Southeast	BMI	Children	Northeast	Children	Smoker
Feature 5	Northwest	Northeast	Sex	Sex	Sex	Northeast

The five most important input features obtained from permutation feature importance. After data preparation, there were nine input features, namely age, sex, BMI, children, smoker, Northeast, Northwest, Southeast, and Southwest.