Connor Hanan

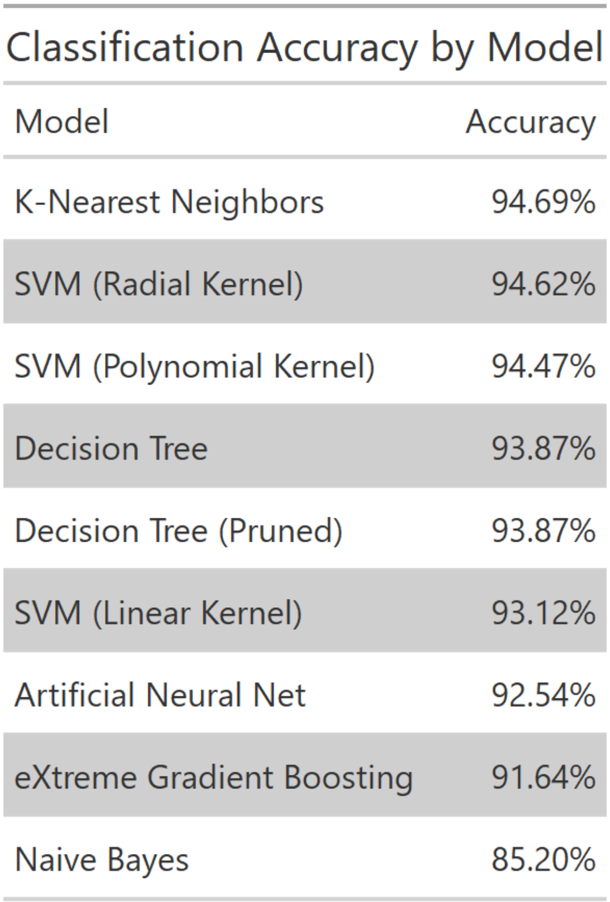
IST 421

November 30 2021

Final Project Report

For our final project, we chose an insurance data set. Our goal was to see if we could accurately predict insurance costs based on an individual's health characteristics. The plan was to see which models scored the highest classification accuracy. Additionally, we added another dataset that gave health attributes about specific states. We linked this to our dataset by assigning the states to the already existing regions. We made the addition because we wanted to see if there was more evidence for confirming a relationship between health insurance costs and region beyond the given attributes in our insurance dataset.

Our dataset contains seven attributes include Age, Sex, BMI, Number of Children, Smoker (yes/no), Region, and Charges. Our target attribute is Charges, which is how much a patient was charged for insurance. We set this as our target as we want to see the relationship between the attributes and insurance costs. There are 1,138 patients in this dataset, which include ages ranging from 18 to 64 year old, BMI ranging from 15 to 53, and number of children from 0 to 30. We will discretize the following attributes accordingly. Ages will be broken up into teens (ages 18 to 19), twenties (ages 20 to 29), thirties (ages 30 to 39), forties (ages 40 to 49), sixties (ages 60 to 64). BMI will be descritized based on the pre-existing BMI weight categories where a BMI of 18.5 is underweight, a bmi between 18.5 to 24.9 is normal, a bmi of 25 to 29.9 is overweight, and a bmi greater than 30 is obese. The number of children will be discretized as patients with zero children, patients with one or two children, and patients with three or more children. Finally, the insurance costs will be discretized in three bins. Low charges will range from $1000 to $2200, medium charges will range from $2200 to $4300, and high charge is $4300 and above. We chose these bin values to break up the costs evenly, not based on how many people's charges are in each bin. The majority of charges did fall in the low range.

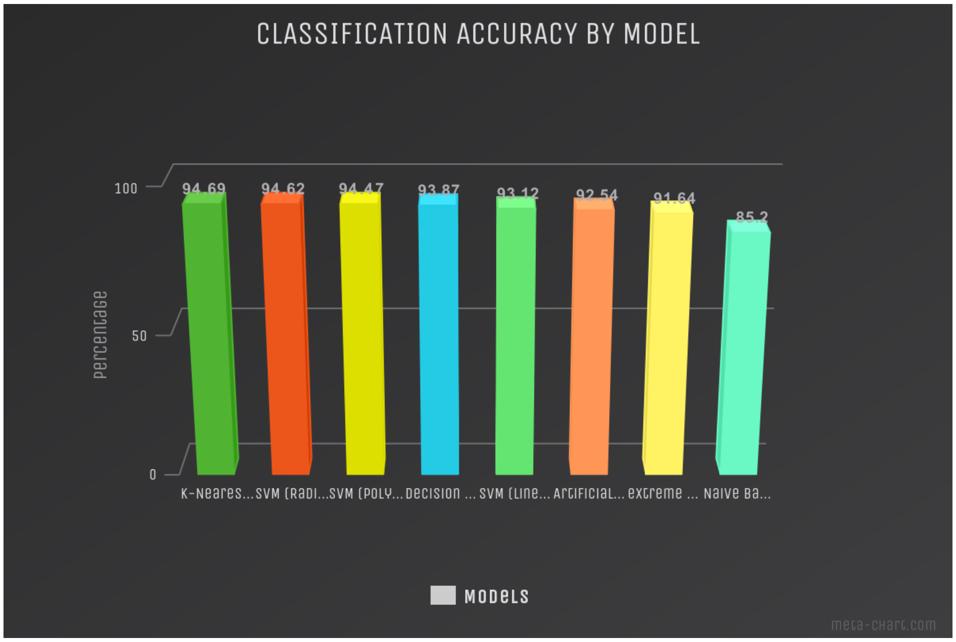


Classification Accuracy by Model

Model	Accuracy
K-Nearest Neighbors	94.69%
SVM (Radial Kernel)	94.62%
SVM (Polynomial Kernel)	94.47%
Decision Tree	93.87%
Decision Tree (Pruned)	93.87%
SVM (Linear Kernel)	93.12%
Artificial Neural Net	92.54%
eXtreme Gradient Boosting	91.64%
Naive Bayes	85.20%

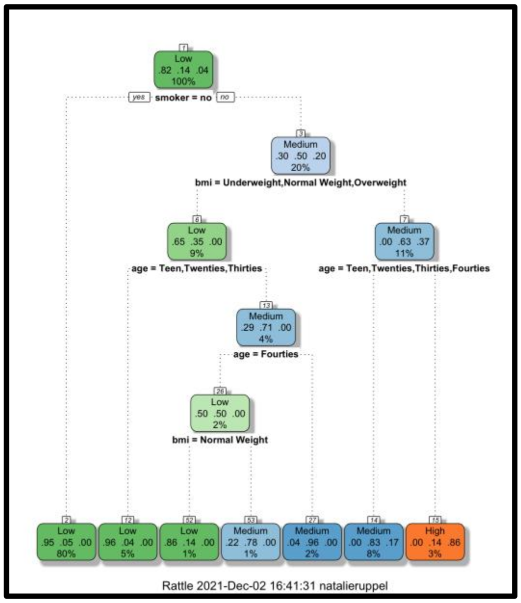
Result:

Our results show that the model with the highest classification accuracy and therefore the best model is the K-Nearest Neighbor algorithm. The K-Nearest Neighbor algorithm works by finding the distances between a centroid and all the examples in the data,



classifying the points which are closest to the centroid as that class. It then recomputes where the mathematical centroid of each group is, and regroups the points to each's closest centroid. This process repeats until there are no more changes, at which time each point is then assigned the label of the centroid to which it is grouped. We, of course, chose three centroids, since we have three classes for our charges classification. The rest of the algorithm set up is actually fairly simple, and it runs fairly quickly (which is a refreshing change of pace from some of our later algorithms). At the end of it all the K-Nearest Neighbor algorithm ended up being our best performing algorithm, with an accuracy of ~94%.

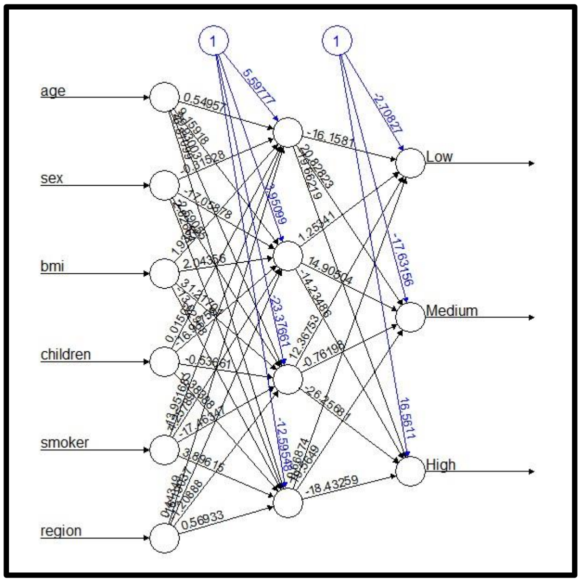
In machine learning, the polynomial kernel is a kernel function commonly used with support vector machines and other kernelized models, that represents the similarity of vectors or training samples in a feature space over polynomials of the original variables, allowing learning of non-linear models. The radial kernel support vector machine is a good approach when the data is not linearly separable and the linear kernel is used when the data is linearly separable, that is, it can be separated using a single line. It is one of the most common kernels to be used. The purpose of the kernel is to draw a line(s) called the hyperplane to maximize the distance between the different classes of points. The radial kernel was the most accurate of the three with an accuracy of 94.62%, followed by the polynomial kernel with an accuracy of 94.47%, and then the linear kernel with an accuracy of 93.12%



Rattle 2021-Dec-02 16:41:31 natalieruppel

Next, we used the decision tree model which we have discussed often in class. A decision tree is a specific type of flow chart used to visualize the decision-making process by mapping out different courses of action, as well as their potential outcomes. Whether the person was a smoker or not was the highest indicator of a potential charge, followed by the BMI, then age. There was no difference between the pruned and unpruned decision tree, both had an accuracy of 93.87%. Potentially if the data was not as clean, there might have been a difference between the two, but the data was already nicely prepared for us.

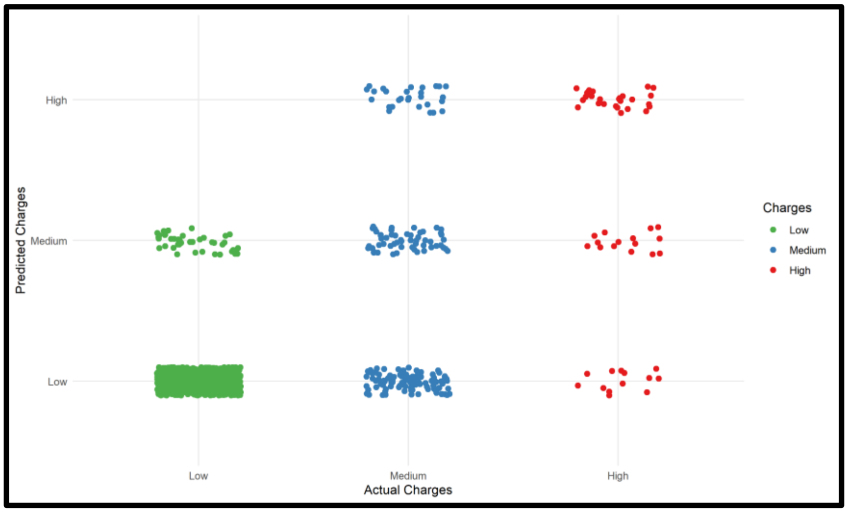
Additionally, we tried out the artificial neural net model. In multi-class classification, the neural network has the same number of output nodes as the number of classes. Each output node belongs to some class and outputs a score for that class. The first step we had to take was to convert all values in our dataset to numerical ones, be it either through binning, factoring or assigning values. Once prepped, the data was ready to be sent to the nnet function from the nnet package - we had issues with the neuralnet package (extremely low accuracy or never converging). We chose to use four nodes in the hidden layer, since it is recommended by the package documentation that the starting number be somewhere between the number of inputs and outputs. The algorithm then runs through the data, guessing and adjusting weights and biases using backpropagation until it has optimized the error as much as possible. The model ended up with about 92% accuracy, which is very good, though not our best accuracy score.



Next, we tried the very complex XGBoost model. XGBoost is a decision-tree-based ensemble machine learning algorithm that uses an extreme gradient boosting framework. The extreme version is the exact same as the original, with the extreme one being focused on speed and performance. Compared to the neural network, there were many more inputs we had to adjust, even though the data had to be cleaned the same way (all numeric values). For instance, we had to select the booster (“gbtree”), set the eta (0.001, also known as the learning rate), max depth (5, number of layers of tree) – and those are just some of the general parameters. Later in the algorithm, we had to choose an objective, which we employed the multi:softprob for multiclass classification, as well as the evaluation metric, where we chose mlogloss (multiclass negative log-likelihood). The XGBoost algorithm surprisingly performed slightly worse than the decision tree (~91% vs. ~94%), even though it is essentially a type of random forest model.

Last, we tried the Naive Bayes model, which has often been discussed in class as well. It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a a particular feature in a class is unrelated to the presence of any other feature. After running naiveBayes() in R on the data, we plotted the corresponding confusion matrix: the points across the (positive) diagonal represent the charges that were accurately predicted. The majority of the predicted charges were in the “low” category, but that is most likely due to most of the charges being placed in the low category when we broke down the charges. Overall the accuracy of this model was 85.20%, making it our lowest performing model.

In conclusion, both of our research questions were answered in running these models. The model that performed best was the K Nearest Neighbor model with a classification accuracy of 94.69%. The worst performing model was the Naive Bayes, which had a classification accuracy of only 85.20%. Additionally, the attribute that was most influential was Smoker(yes/no). Overall, most of our models worked effectively in determining a high classification accuracy and the smoking attribute revealed a clear and strong influence over health insurance costs.


High
Medium
Low
Predicted Charges
Low
Medium
High
Actual Charges
Charges
Low
Medium
High