**Student: Seif Kungulio**

**Date: 05/04/2025**

**Subject: Final Project**

**Class: DSCI 512**

**Section: 01W**

**Instructor: Dr. Nengbing Tao**

**File Name: FinalProject\_Kungulio\_Seif.docx**

1. **Data Preparation**
   1. Load the dataset insurance.csv into memory.

A screenshot of a computer

AI-generated content may be incorrect.

The code loads and explores the “insurance.csv” dataset, which contains 1,338 rows and 7 variables related to individuals’ demographics, health behaviors, and insurance charges. An initial preview shows a mix of numeric and categorical variables, with noticeable variation in medical costs. For example, smokers tend to have higher charges. Overall, the dataset is well-structured and ready for further analysis or predictive modeling.

* 1. In the data frame, transform the variable charges by setting insurance$charges log(insurance$charges). Do not transform it outside of the data frame.

A screenshot of a computer

AI-generated content may be incorrect.

The code applies a log transformation to the “charges” variable in the “insurance” dataset to reduce skewness, normalize the distribution, and stabilize variance—making the data more suitable for regression modeling. The transformed values are compressed and more statistically manageable, improving model performance and interpretability.

* 1. Using the data set from 1.b, use the model.matrix() function to create another data set that uses dummy variables in place of categorical variables. Verify that the first column only has ones (1) as values, and then discard the column only after verifying it has only ones as values.

A computer screen shot of a program

AI-generated content may be incorrect.

The code prepares the “insurance” dataset for regression modeling by converting categorical variables into dummy variables using “model.matrix()”. It confirms that the first column (the intercept) contains only ones, then removes it to avoid redundancy, ensuring the dataset is fully numeric and ready for further analysis or modeling.

* 1. Use the sample() function with set.seed equal to 1 to generate row indexes for your training and tests sets, with 2/3 of the row indexes for your training set and 1/3 for your test set. Do not use any method other than the sample() function for splitting your data.

A computer screen with text

AI-generated content may be incorrect.

The code performs a randomized split of the “insurance” dataset into training and testing sets, using two-thirds of the data for training. It sets a seed for reproducibility and uses the “sample()” function to generate row indices for the training set. This ensures consistent and unbiased model evaluation on unseen data.

* 1. Create a training and test data set from the data set created in 1.b using the training and test row indexes created in 1.d. Unless otherwise stated, only use the training and test data sets created in this step.

A computer screen with text

AI-generated content may be incorrect.

The code splits the “insurance” dataset into a training set with 892 rows and a test set with 446 rows using previously sampled indices. This step ensures a proper division of data for building and evaluating predictive models, supporting accurate and unbiased model performance assessment.

* 1. Create a training and test data set from data set created in 1.c using the training and test row indexes created in 1.d

A computer screen with white text

AI-generated content may be incorrect.

The code splits the dummy-variable version of the insurance dataset into a training set with 892 rows and a test set with 446 rows using previously generated indices. This prepares fully numeric data for machine learning models that require consistent input formats for training and evaluation.

1. **Build a multiple linear regression model.**
   1. Perform multiple linear regression with charges as the response and the predictors are age, sex, bmi, children, smoker, and region. Print out the results using the summary() function. Use the training data set created in step 1.e to train your model.

A screenshot of a computer program

AI-generated content may be incorrect.

The regression analysis shows that insurance charges are strongly influenced by several factors, with the model explaining over 78% of the variation in charges. Key predictors such as age, BMI, number of children, and especially smoking status have significant positive effects on charges. Smoking is the most impactful, leading to much higher costs. While gender is statistically significant, its effect is minor. Regional differences also exist, with the Southeast and Southwest associated with lower charges. Overall, the model effectively highlights the main factors in driving insurance costs.

* 1. Is there a relationship between the predictors and the response?

The multiple linear regression model shows a strong relationship between the predictors and insurance charges, with an adjusted R-squared of 78.2%, indicating the model explains most of the variation in charges. Key predictors such as age, BMI, number of children, and especially smoking status significantly influence charges. Smokers are associated with notably higher charges, making it the strongest predictor. Geographic regions also impact charges, with the Southeast and Southwest linked to lower costs. Overall, the model confirms that several demographic and behavioral factors meaningfully affect insurance charges.

* 1. Does sex have a statistically significant relationship to the response?

The model shows that sex is a statistically significant predictor of insurance charges, with males having slightly higher charges than females. However, the effect size is small, indicating that while gender does influence charges, its practical impact is minimal compared to stronger predictors like smoking status.

* 1. Perform best subset selection using the stepAIC() function from the MASS library, choose best model based on AIC. For the “direction” parameter in the stepAIC() method, set direciton”backward”

A screenshot of a computer program

AI-generated content may be incorrect.

The backward stepwise regression analysis identified a model that includes age, sex, BMI, number of children, smoking status, and region as significant predictors of insurance charges. All variables remained in the model because removing any of them increased the AIC, indicating each contribute meaningfully. Smoking status had the strongest impact, significantly increasing charges, followed by age and regional differences. The model explains about 78% of the variance in charges (R² = 0.784), and all predictors are statistically significant. Overall, the model is robust, accurate, and highlights key factors influencing medical insurance costs.

* 1. Compute the test error of the best model in #3d based on AIC using LOOCV using trainControl() and train() from the caret library. Report the MSE by squaring the reported RMSE.

A computer screen with white text

AI-generated content may be incorrect.

The Leave-One-Out Cross-Validation (LOOCV) was used to evaluate the performance of the best regression model selected by AIC. Using the “caret” package in R, the model was trained and tested across all individual data points. The resulting Mean Squared Error (MSE) was approximately 0.184, indicating that the model has good predictive accuracy and generalizes well to unseen data.

* 1. Calculate the test error of the best model in #3d based on AIC using 10-fold Cross-Validation. Use train and trainControl from the caret library. Refer to model selected in #3d based on AIC. Report the MSE.

A computer screen with white text

AI-generated content may be incorrect.

The model was trained with selected predictors (age, BMI, children, smoker, and region) and achieved a Mean Squared Error (MSE) of approximately 0.181. This low MSE indicates strong predictive accuracy and suggests that the model generalizes well to unseen data, reaffirming its reliability for estimating insurance charges.

* 1. Calculate and report the test MSE using the best model from 2.d and test data set created in step 1.e.

A computer screen with text

AI-generated content may be incorrect.

The resulting MSE is approximately 0.231, indicating a low average prediction error on unseen data. While slightly higher than the cross-validation errors, this is expected and confirms that the model generalizes well and maintains strong predictive performance.

* 1. Compare the test MSE calculated in step 2.f using 10-fold cross-validation with the test MSE calculated in step 2.g. How similar are they?

A computer screen with white text

AI-generated content may be incorrect.

The code compares the 10-fold cross-validation MSE (0.1807) with the test set MSE (0.2313). While the test MSE is slightly higher—as expected due to its evaluation on unseen data—the values are relatively close. This indicates that the model generalizes well, is not overfitting, and performs reliably on new data.

1. **Build a regression tree model.**
   1. Build a regression tree model using function tree(), where charges is the response and the predictors are age, sex, bmi, children, smoker, and region.

A computer screen shot of a computer program

AI-generated content may be incorrect.

The regression tree model was built to predict insurance charges using several predictors, but only age and number of children were used in the final tree, indicating their stronger influence. The model has five terminal nodes and a residual mean deviance of 0.5649, reflecting moderate predictive accuracy. Residuals show a balanced distribution around zero, but the range indicates potential prediction errors. Overall, while the model highlights key predictors, data preprocessing needs improvement for better performance.

* 1. Find the optimal tree by using cross-validation and display the results in a graphic. Report on the best size.

A graph with a line

AI-generated content may be incorrect.

The plot shows that deviance decreases as tree size increases, with a sharp improvement up to size **3**. Beyond that, additional splits yield minimal performance gains. This suggests that a tree size of 3 is optimal, offering a good balance between model accuracy and complexity, and helps avoid overfitting based on the bias-variance trade-off.

* 1. Justify the number you picked for the optimal tree with regard to the principle of variance-bias trade-off.

A tree size of **3** is optimal because it strikes a balance between bias and variance. It significantly reduces bias compared to smaller trees while avoiding the overfitting and high variance that come with larger trees, aligning well with the bias-variance trade-off principle.

* 1. Prune the tree using the optimal size found in 3.b.

A computer screen with white text

AI-generated content may be incorrect.

The pruned regression tree, reduced to three terminal nodes, uses only the variable **age** for prediction. Despite being simpler, it maintains similar predictive performance to the unpruned tree, with only a slight increase in deviance. The residuals remain centered around zero, indicating unbiased predictions. Overall, the pruned model achieves a strong balance between simplicity and accuracy, aligning well with the bias-variance trade-off.

* 1. Plot the best tree model and give labels.

A white background with black lines

AI-generated content may be incorrect.

The pruned regression tree uses **age** as the sole predictor to group individuals into three categories based on insurance **charges**. It splits at ages **22.5** and **36.5**, showing that predicted charges increase with age. Despite its simplicity, the model effectively captures the relationship between age and healthcare costs, offering clear and interpretable predictions.

* 1. Calculate the test MSE for the best model.

A computer screen with white text

AI-generated content may be incorrect.

The pruned regression tree achieved a test MSE of 0.7035, indicating reasonable predictive performance on unseen data. A warning about NAs introduced by coercion suggests that some variables in the test set may be improperly formatted, likely due to unconverted categorical data. Despite this, the model—based solely on age—performed adequately, though data formatting should be reviewed for cleaner execution.

1. **Build a random forest model.**
   1. Build a random forest model using function randomForest(), where charges is the response and the predictors are age, sex, bmi, children, smoker, and region.

A screenshot of a computer program

AI-generated content may be incorrect.

The Random Forest regression model predict healthcare charges using predictors such as age, sex, BMI, children, smoker status, and region. The model was built with 500 trees and set to measure variable importance. The summary includes predicted values for 892 observations, and tracks model performance via Mean Squared Error (MSE) and R-squared across all trees. It also supports out-of-bag (OOB) error estimation for internal validation. The setup enables later analysis of which variables most influence the predicted charges.

* 1. Compute the test error using the test data set.

A computer screen with white text

AI-generated content may be incorrect.

The Random Forest regression model was evaluated using test data by generating predictions and calculating the Mean Squared Error (MSE) to assess its performance. The resulting MSE of **0.1779905** suggests that the model provides accurate predictions and generalizes well to new data, particularly if the response variable was log-transformed.

* 1. Extract variable importance measures using the importance() function.

A screen shot of a computer

AI-generated content may be incorrect.

The variable importance output from the Random Forest model shows that **smoker** status and **age** are the most influential predictors of healthcare charges, with the highest values in both %IncMSE and IncNodePurity. **BMI** and **children** have moderate impact, while **region** and **sex** contribute the least. This indicates that lifestyle and demographic factors, especially **smoking**, play a key role in predicting healthcare costs.

* 1. Plot the variable importance using the function, varImpPlot(). Which are the top 3 important predictors in this model?

A comparison of a model

AI-generated content may be incorrect.

The variable importance plot from the Random Forest model shows that **smoker**, **age**, and **BMI** are the top three predictors influencing healthcare charges. Smoker is the most significant by far, followed by age, while BMI ranks slightly above children in importance. These variables contribute most to the model’s accuracy and decision-making process.

1. **Build a support vector machine model.**
   1. The response is charges and the predictors are age, sex, bmi, children, smoker, and region. Please use the svm() function with radial kernel and gamma 5 and cost 50.

A screenshot of a computer

AI-generated content may be incorrect.

This SVM regression model predicts insurance charges using predictors like age, sex, BMI, children, smoking status, and region. It employs a radial kernel with a high gamma (5) and cost (50), indicating a complex, flexible model tailored to the training data. The model uses epsilon-regression with a tolerance of 0.1, allowing small prediction errors. A total of 715 support vectors were used, suggesting many data points influence the model, which may lead to overfitting. While the model is likely accurate on training data, its generalization should be tested on unseen data to confirm performance.

* 1. Perform a grid search to find the best model with potential cost: 1, 10, 50, 100 and potential gamma: 1,3 and 5 and potential kernel: “linear”, “radial” and “sigmoid”. And use the training set created in step 1.e.

A screenshot of a computer

AI-generated content may be incorrect.

A grid search was conducted to find the best SVM regression model using different combinations of kernel types, cost values, and gamma values. The best-performing model used a **radial** kernel with **cost = 1** and **gamma = 1**, achieving the lowest cross-validation error of **0.1902**. This configuration outperformed others, including models with linear and sigmoid kernels. The results indicate that a radial SVM with low cost and gamma values provides the best balance of flexibility and generalization for predicting insurance charges.

* 1. Print out the model results. What are the best model parameters?

A screenshot of a computer

AI-generated content may be incorrect.

The best SVM regression model uses a **radial kernel** with **cost = 1**, **gamma = 1**, and **epsilon = 0.1**, selected through cross-validation. It relies on **493 support vectors**, indicating a balanced model that captures non-linear patterns while maintaining generalizability. This configuration provides an effective trade-off between model accuracy and complexity for predicting insurance charges.

* 1. Forecast charges using the test dataset and the best model found in c).

A screenshot of a computer

AI-generated content may be incorrect.

The best SVM model was used to predict insurance charges on the test dataset, producing results on a log scale. The predicted values ranged from 7.123 to 10.786, with a mean of 9.039 and a median of 9.142, indicating a symmetrical distribution. The model appears to generalize well, showing consistent predictions without extreme outliers. To better interpret the results, back-transformation to the original dollar scale and error evaluation would be useful for the next steps.

* 1. Compute the MSE (Mean Squared Error) on the test data.

A computer screen with white text

AI-generated content may be incorrect.

The SVM model achieved a test Mean Squared Error (MSE) of **0.226** on the log-transformed charges, indicating accurate predictions with low average squared error. While this reflects good model performance on the transformed scale, back-transforming to the original scale would provide clearer real-world interpretation.

1. **Perform the k-means cluster analysis.**
   1. Remove the sex, smoker, and region, since they are not numerical values.

A computer screen with white text

AI-generated content may be incorrect.

The code shows a data preprocessing step, where non-numeric variables (“sex”, “smoker”, and “region”) are removed to create a new dataset, “insurance\_numeric”, containing only numerical features: “age”, “bmi”, “children”, and “charges”. This ensures compatibility with modeling techniques that require numerical input. The preview of the data confirms the dataset is clean and ready for further analysis, with “charges” appearing to be log-transformed to improve model performance.

* 1. Determine the optimal number of clusters. Justify your answer. It may take longer running time since it uses a large dataset.

A graph showing the number of points

AI-generated content may be incorrect.

The Gap Statistic plot indicates that the optimal number of clusters is **2**, where the gap value peaks. This suggests the data is best divided into two well-separated groups, as adding more clusters beyond this point does not significantly improve clustering quality and may lead to unnecessary complexity.

* 1. Perform k-means clustering using the 3 clusters.

A screenshot of a computer

AI-generated content may be incorrect.

K-means clustering was performed on the “insurance\_numeric” dataset using three clusters. The model successfully assigned all 1,338 observations into groups, with metrics such as the total sum of squares and within-cluster sum of squares offering insights into the compactness and separation of the clusters. Although the Gap Statistic had previously indicated that two clusters might be optimal, this step intentionally explores three clusters for comparison and deeper analysis. The summary output confirms that the algorithm executed correctly, laying the groundwork for further investigation into cluster characteristics, such as their sizes and centroid values.

* 1. Visualize the clusters in different colors.

A diagram of multiple colored circles

AI-generated content may be incorrect.

The cluster visualization shows the results of k-means clustering with three clusters applied to the “insurance\_numeric” dataset, using PCA for dimensionality reduction. Cluster 3 appears well-separated on the left, while Clusters 1 and 2 overlap more in the center and right, indicating similarities between them. The plot suggests that while the data can be grouped into three clusters, further analysis may be needed to confirm whether this separation is meaningful or if fewer clusters might be more appropriate.

1. **Build a neural networks model.**
   1. Remove the sex, smoker, and region, since they are not numerical values.

A computer screen with text

AI-generated content may be incorrect.

The code filters the “insurance” dataset to retain only numerical variables—”age”, “bmi”, “children”, and “charges”—by removing categorical features like “sex”, “smoker”, and “region”. The resulting dataset, “insurance\_nn”, contains clean numeric data suitable for statistical modeling. The first six rows confirm successful extraction, showing varied values across all selected columns. This step ensures the data is ready for machine learning or regression analysis.

* 1. Standardize the inputs using the scale() function.

A screenshot of a computer

AI-generated content may be incorrect.

The code standardizes the numerical features (“age”, “bmi”, “children”, and “charges”) in the dataset using the “scale()” function, converting each value into a z-score with a mean of 0 and standard deviation of 1. This ensures all variables are on the same scale, preventing any single feature from disproportionately influencing the model. The output confirms successful standardization, making the data suitable for scale-sensitive algorithms like clustering or neural networks.

* 1. Convert the standardized inputs to a data frame using the as.data.frame() function.

A computer screen with white text

AI-generated content may be incorrect.

The code converts the standardized data from a matrix to a data frame using “as.data.frame()”, ensuring compatibility with R functions that require data in “data.frame” format. The “class()” check confirms the successful conversion, making the dataset ready for further analysis or modeling.

* 1. Split the dataset into a training set containing 80% of the original data and the test set containing the remaining 20%.

A computer screen with white text

AI-generated content may be incorrect.

The code splits the standardized dataset into training and test sets, using 80% of the data (1,070 records) for training and 20% (268 records) for testing. A random seed ensures reproducibility, and the “sample()” function is used to randomly select training indices. This setup allows the model to be trained in one subset and evaluated on another, supporting accurate performance assessment and preventing overfitting.

* 1. The response is charges and the predictors are age, bmi, and children. Please use 1 hidden layer with 1 neuron.

A computer screen shot of a program

AI-generated content may be incorrect.

A simple neural network model was built using the “neuralnet” package to predict insurance charges based on age, BMI, and number of children. The model uses one hidden layer with a single neuron and was trained on standardized data. The summary confirms successful training on 1,070 observations, with a structure appropriate for regression tasks. This model is now ready for evaluation or further refinement.

* 1. Plot the neural networks.

A diagram of a network

AI-generated content may be incorrect.

The neural network, consisting of a single hidden neuron, was trained to predict insurance charges using age, BMI, and number of children as input features. The connection weights reveal that age has a moderately strong negative influence on the prediction, while BMI and children have relatively minor effects. The model achieved a training error of 352.77 and required 22,494 iterations to converge. This visualization confirms that the model was successfully trained and provides clear insight into how each variable contributes to the output.

* 1. Forecast the charges in the test dataset.

A computer screen with white text

AI-generated content may be incorrect.

The trained neural network model was used to predict insurance charges on the test dataset, producing outputs in standardized form. The predicted values range from -1.15 to 0.90, with a mean near zero, reflecting the scaled nature of the data. These results confirm successful forecasting, but the predictions need to be converted back to the original scale for real-world interpretation and evaluation.

* 1. Get the observed charges of the test dataset.

A screenshot of a computer

AI-generated content may be incorrect.

This step extracts and summarizes the actual (observed) insurance charges from the standardized test dataset. The observed values range from -2.26 to 2.14 and are centered around zero, reflecting the standardized scale. These values are essential for evaluating the model’s performance by comparing them to the predicted charges.

* 1. Compute test error (MSE).

A computer screen with text

AI-generated content may be incorrect.

The Mean Squared Error (MSE) was computed to assess the prediction accuracy of the neural network on the test dataset. The resulting MSE of 0.8143 reflects the average squared difference between the predicted and actual values, based on standardized data. This metric serves as a critical indicator of the model's performance and provides a baseline for future tuning or comparison with other models.

1. **Putting it all together.**
   1. For predicting insurance charges, your supervisor asks you to choose the best model among the multiple regression, regression tree, random forest, support vector machine, and neural network models. Compare the test MSEs of the models generated in steps 2.g, 3.f, 4.b, 5.e, and 7.d. Display the names for these types of these models, using these labels: Multiple Linear Regression, Regression Tree, Random Forest, Support Vector Machine, and Neural Network and their corresponding test MSEs in a data.frame. Label the column in your data frame with the labels as Model.Type and label the column with the test MSEs as Test.MSE and round the data in this column to 4 decimal places. Present the formatted data to your supervisor and recommend which model is best and why.

A screenshot of a computer

AI-generated content may be incorrect.

The code compares five predictive models for estimating insurance charges using Test Mean Squared Error (MSE). Among Multiple Linear Regression, Regression Tree, Random Forest, Support Vector Machine, and Neural Network, the Random Forest model achieves the lowest MSE (0.1780), indicating the best performance. This suggests Random Forest is the most accurate and reliable model for predicting insurance charges in this analysis.

* 1. Another supervisor from the sales department has requested your help to create a predictive model that his sales representatives can use to explain to clients what the potential costs could be for different kinds of customers, and they need an easy and visual way of explaining it. What model would you recommend, and what are the benefits and disadvantages of your recommended model compared to other models?

A **Regression Tree** model is recommended for the sales department because it provides a simple, visual, and easy-to-explain way to predict insurance costs. It allows sales reps to clearly show clients how different factors affect pricing through intuitive “if-then” decision paths. While it may not be as accurate as models like Random Forests, its interpretability makes it ideal for client communication. For best results, it can be used alongside more accurate models for internal predictions.

* 1. The supervisor from the sales department likes your regression tree model. But she says that the salespeople say the numbers in it are way too low and suggests that maybe the numbers on the leaf nodes predicting charges are log transformations of the actual charges. You realize that in step 1.b of this project that you had indeed transformed charges using the log function. And now you realize that you need to reverse the transformation in your final output. The solution you have is to reverse the log transformation of the variables in the regression tree model you created and redisplay the result.

**Follow these steps:**

**--------------------------**

Copy your pruned tree model to a new variable.

A computer screen shot of a computer program

AI-generated content may be incorrect.

The pruned regression tree model uses only **age** as the predictor and has **3** terminal nodes, making it simple and easy to interpret. With a residual mean deviance of **0.5959** and balanced residuals, the model provides reasonably accurate predictions while maintaining high transparency. This makes it well-suited for situations where clarity and explainability are more important than maximum predictive accuracy.

In your new variable, find the data.frame named “frame” and reverse the log transformation on the data.frame column yval using the exp() function. (If the copy of your pruned tree model is named copy\_of\_my\_pruned\_tree, then the data frame is accessed as copy\_of\_my\_pruned\_tree$frame, and it works just like a normal data frame.).

A computer screen with white text

AI-generated content may be incorrect.

The code reverses the earlier log transformation on predicted insurance charges from a pruned regression tree, converting them back to dollar amounts using the “exp()” function. The summary shows that predicted charges range from $3,723 to $13,300, with a mean of $7,561. This step makes the model’s output interpretable and usable for real-world applications, such as explaining costs to clients.

After you reverse the log transform on the yval column, then replot the tree with labels.

A graph of a person

AI-generated content may be incorrect.

The pruned regression tree uses **age** as the sole predictor to estimate insurance charges, dividing clients into three age groups with distinct cost predictions: $3,723 for those under 22.5, $6,576 for ages 22.5–36.5, and $13,300 for those 36.5 and older. This simple, interpretable model effectively highlights how age influences insurance costs, making it ideal for client-facing explanations.