Predicting Healthcare Insurance Charges

by

Richard Alexander Gonzalez Saenz (ID: 2023207)

Higher Diploma in Science in Data Analytics for Business

Data Visualization Techniques, Machine Learning for Business
David McQuaid, Muhammad Iqbal

CCT College

Dublin, Ireland

Table of Contents

Introduction

The healthcare landscape in the United States has been significantly impacted by inflation, leading to increased medical insurance costs. Predicting these costs accurately is crucial for both insurers and beneficiaries. This report analyzes a dataset containing information about insurance beneficiaries to build predictive models for insurance charges. The data preparation, feature engineering, exploratory data analysis (EDA), and machine learning modeling processes are detailed to achieve the best possible accuracy.

Data Preparation

Characterisation of the Dataset

The dataset contains 1338 rows and 8 columns. The columns include age, gender, BMI, number of children, smoker status, residential region, No Claims Bonus, and charges.

ATTRIBUTES	
Age	Age of the primary beneficiary under the insurance policy.
Gender	Gender of the insurance contractor (0-female or 1-male).
BMI	Body Mass Index, a numerical indication of relative body weight based on height and weight.
Children	Number of children or dependents covered by the health insurance.
Smoker	Binary variable indicating whether the beneficiary is a smoker (1-yes/0-no).
Region	Residential area of the insurance beneficiary in the US (northeast, southeast, southwest, northwest).
NoClaimsBonus	Percentage of No Claims Bonus (5%, 10%, 15%, 20%).
Charges	Individual medical costs billed by health insurance.

The dataset contains missing values in several attributes, as shown below:

Missing values in each attribute:

MISSING VALUES	
Age	0
Gender	5
BMI	12
Children	10
Smoker	4
Region	2
NoClaimsBonus	3
Charges	7

Data Cleaning

To address the missing values, rows with any missing values were removed, resulting in a dataset with 1295 rows. This step ensures that the dataset is complete and ready for analysis.

Renaming Columns

Columns were renamed for better readability and consistency:

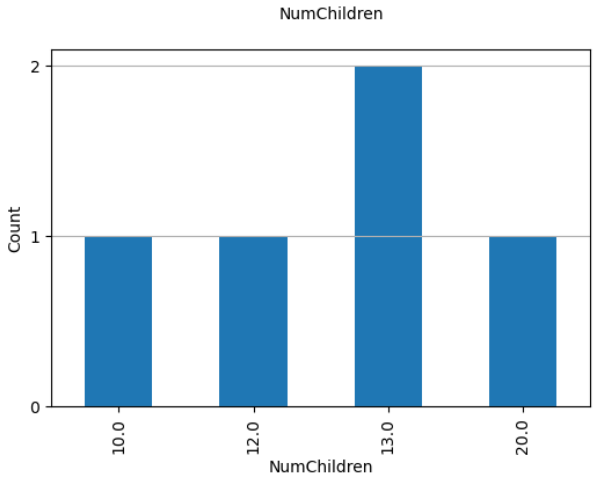
- **age** to **Age**
- **sex** to **Gender**
- **bmi** to **BMI**
- **children** to **NumChildren**
- **smoker** to **Smoker**
- **region** to **Region**
- **charges** to **Charges**

Normalizing and Correcting Values

The ***Region*** column values were normalized to lowercase, and misspellings were corrected to ensure consistency.

Removing Outliers

Values in the ***NumChildren*** column greater than 5 were removed due to their rarity, which could bias the model.



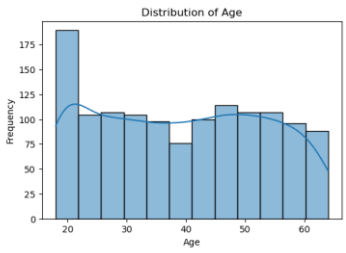
Exploratory Data Analysis (EDA)

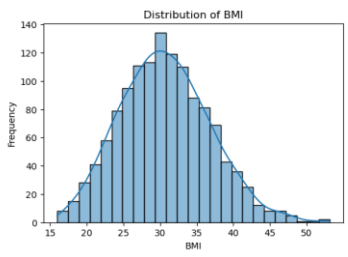
EDA was performed to understand the distribution and relationships within the data. Various visualizations, such as histograms and scatter plots, were used to explore the data.

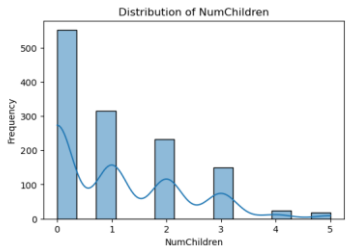
Distribution Analysis

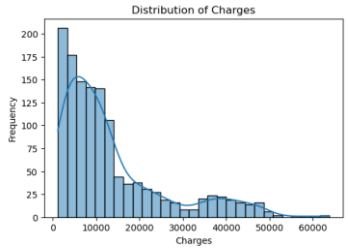
The distribution of numerical columns such as Age, BMI, NumChildren, and Charges was visualized using histograms. This helps understand the spread and skewness of these variables.

DISTRIBUTION OF DEPENDENT VARIABLE









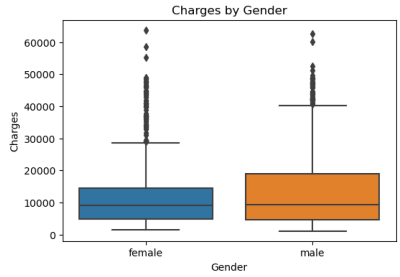
Pairplot

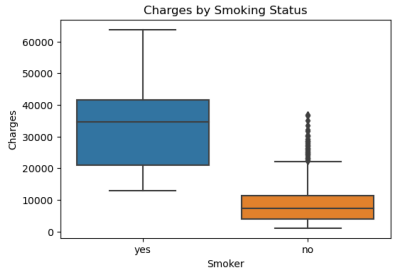
A pairplot was created to explore relationships between variables, particularly focusing on the impact of the **Smoker** variable.

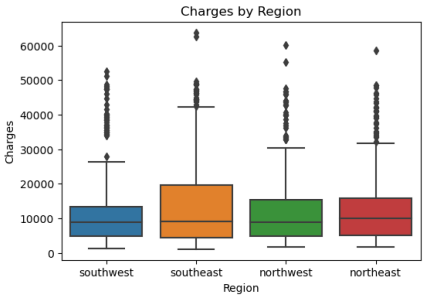
Boxplots

Boxplots were used to detect outliers and understand the spread of Charges across different categories such as Gender, Smoker status, and Region.

CHARGES VS GENDER, SMOKER AND REGION

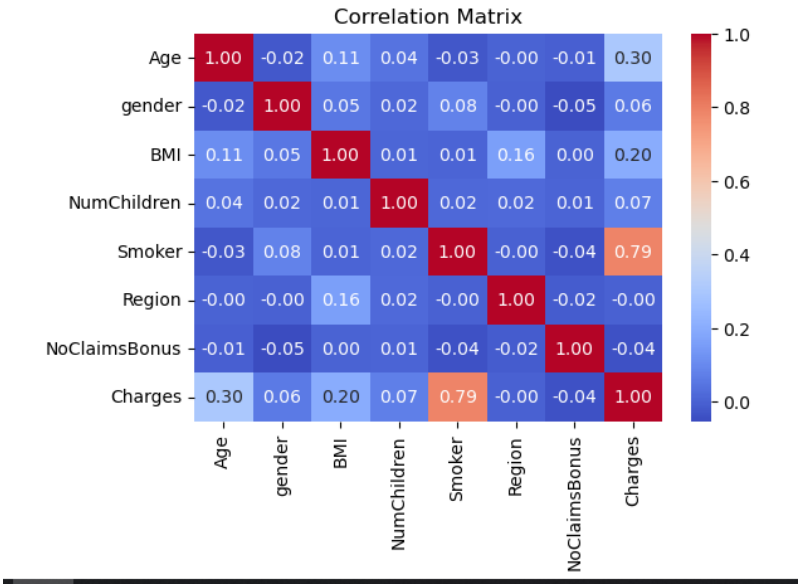






Correlation Heatmap

A correlation heatmap was generated to understand the relationships between numerical variables, aiding in feature selection and identifying multicollinearity.



Feature Engineering, Encoding, and Scaling

Feature Engineering

The BMI column was categorized into different weight categories based on CDC guidelines: Underweight, Normal weight, Overweight, Mild Obesity, Moderate Obesity, and Severe Obesity.

BMI_Category
Overweight
Mild Obesity
Mild Obesity
Normal weight
Overweight

Encoding

- Binary Categorical Variables: Gender and Smoker were encoded using Label Encoding.
- Nominal Categorical Variables: Region and BMI Category were encoded using One-Hot Encoding.
- Ordinal Categorical Variables: NoClaimsBonus and NumChildren were also encoded using One-Hot Encoding.

Scaling

Numerical features were scaled using StandardScaler to ensure they are on the same scale, which is essential for many machine learning algorithms.

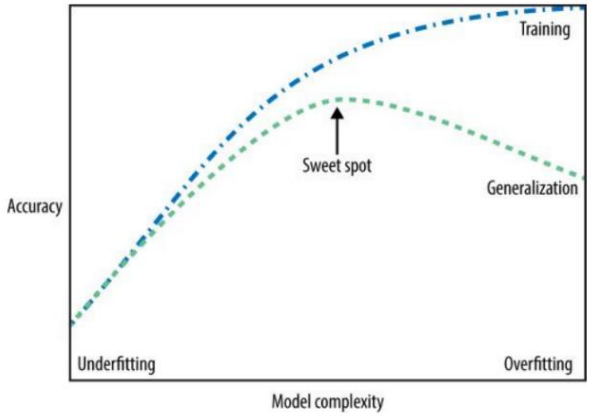
Machine Learning Models

Supervised Learning: Regression Models

Regression models were chosen to predict the continuous target variable, **Charges**.

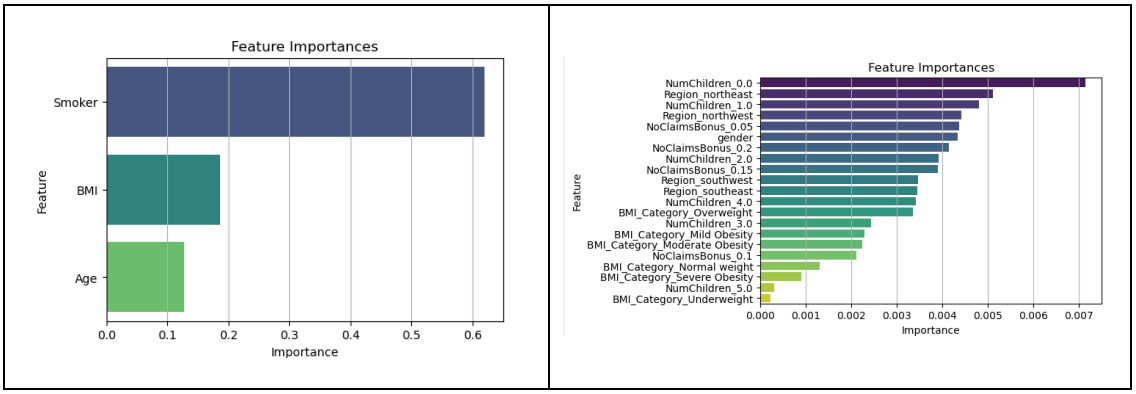
Sweet Spot

In the image, the area of the square is dedicated to searching for the right amount of model complexity that can distinguish the original data from noise. The sweet point means that the model is complex enough that it can catch the patterns of the data but not that complex that it memorizes the training data. This balance is critical in modeling to obtain high predictive performance on new, unseen data.



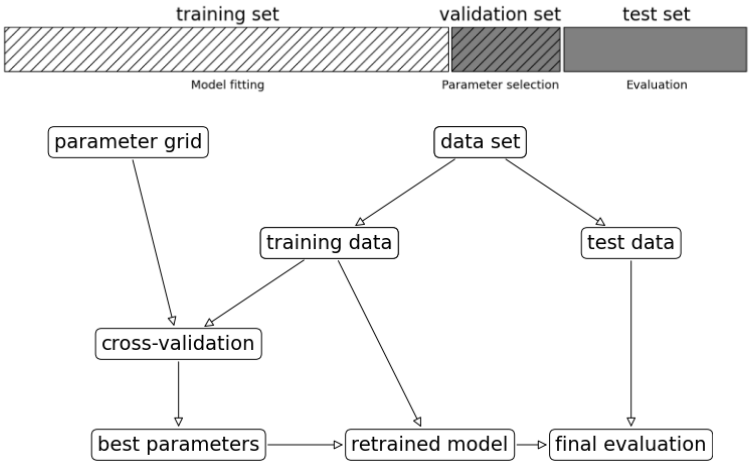
Feature Selection: Random Forest Regressor

A Random Forest Regressor was used to identify the most important features. The features **Smoker**, **BMI**, and **Age** were found to be the most important.



Hyperparameter Tuning and Model Evaluation

Hyperparameter tuning was performed using GridSearchCV to find the optimal parameters for each model. The models were evaluated using cross-validation to ensure robust performance metrics.



Train Test Split

The dataset was split into training, validation set and testing set to evaluate model performance.

Choice of Models

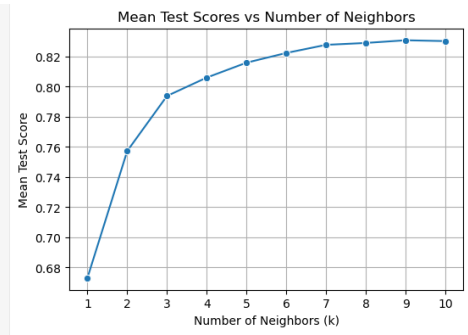
Supervised learning models were chosen for this analysis due to the labeled nature of the dataset. The models considered include:

- K-Neighbors Regressor
- Linear Regression
- Random Forest Regressor
- Support Vector Regressor

K-Neighbors Regressor, Hyperparameter Tuning and Accuracy

GridSearchCV was used to tune the hyperparameters of the K-Neighbors Regressor. The best parameters were found to be **n_neighbors=9**, achieving a test set score of 0.85.

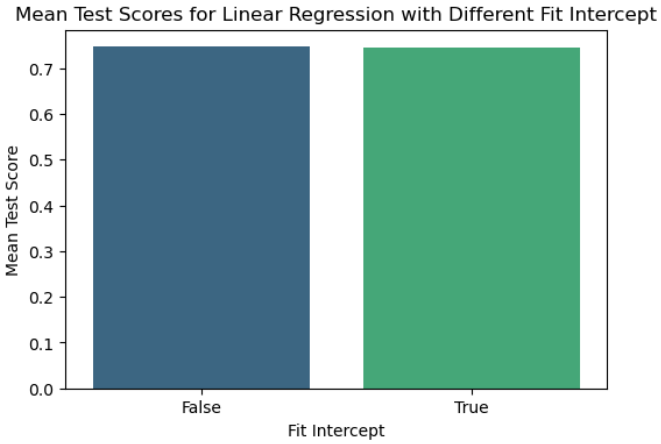
TEST SET SCORE	0.85
BEST PARAMETERS	{'n_neighbors': 9}
BEST CROSS-VALIDATION SCORE	0.83



Linear Regression, Hyperparameter Tuning and Accuracy

GridSearchCV was used to tune the hyperparameters of Linear Regression. The best parameters were **fit_intercept=False**, achieving a test set score of 0.74.

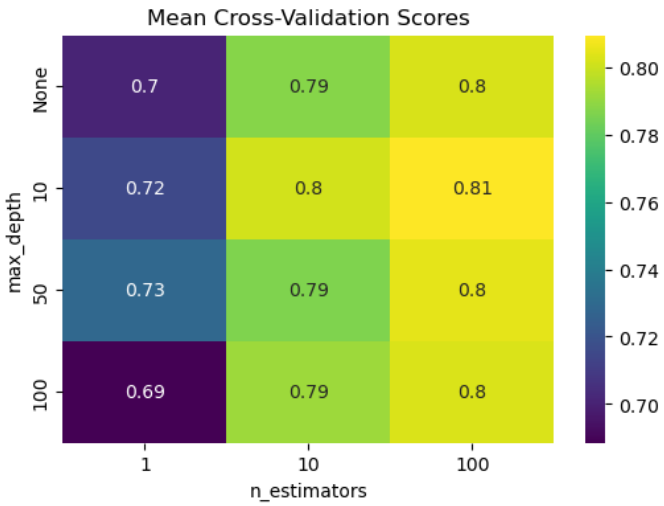
TEST SET SCORE	0. 74
BEST PARAMETERS	{'fit_intercept': False}
BEST CROSS-VALIDATION SCORE	0.75



Random Forest Regressor, Hyperparameter Tuning and Accuracy

GridSearchCV was used to tune the hyperparameters of the Random Forest Regressor. The best parameters were **max_depth=10** and **n_estimators=100**, achieving a test set score of 0.83.

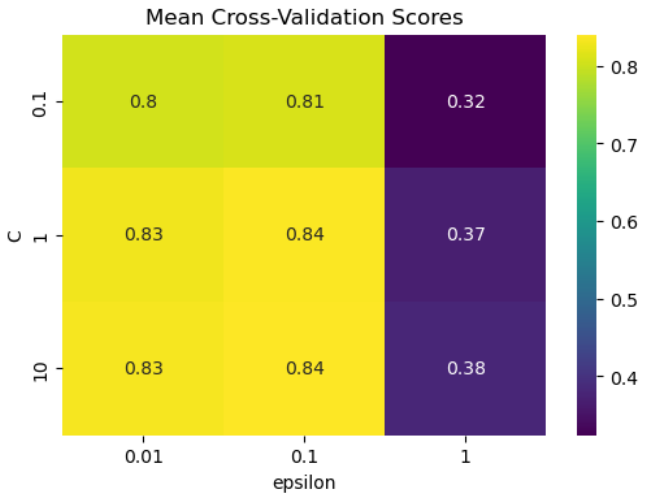
TEST SET SCORE	0.83
BEST PARAMETERS	{'max_depth': 10, 'n_estimators': 100}
BEST CROSS-VALIDATION SCORE	0.81



Support Vector Regressor, Hyperparameter Tuning and Accuracy

GridSearchCV was used to tune the hyperparameters of the Support Vector Regressor. The best parameters were **C=10** and **epsilon=0.1**, achieving a test set score of 0.87.

TEST SET SCORE	0.87
BEST PARAMETERS	{'C': 10, 'epsilon': 0.1}
BEST CROSS-VALIDATION SCORE	0.84



K-Fold cross-validation

K-Fold Cross-Validation was used to evaluate the models' performance. The Support Vector Regressor achieved the highest cross-validation score, followed by the K-Neighbors Regressor, Random Forest Regressor, and Linear Regression.

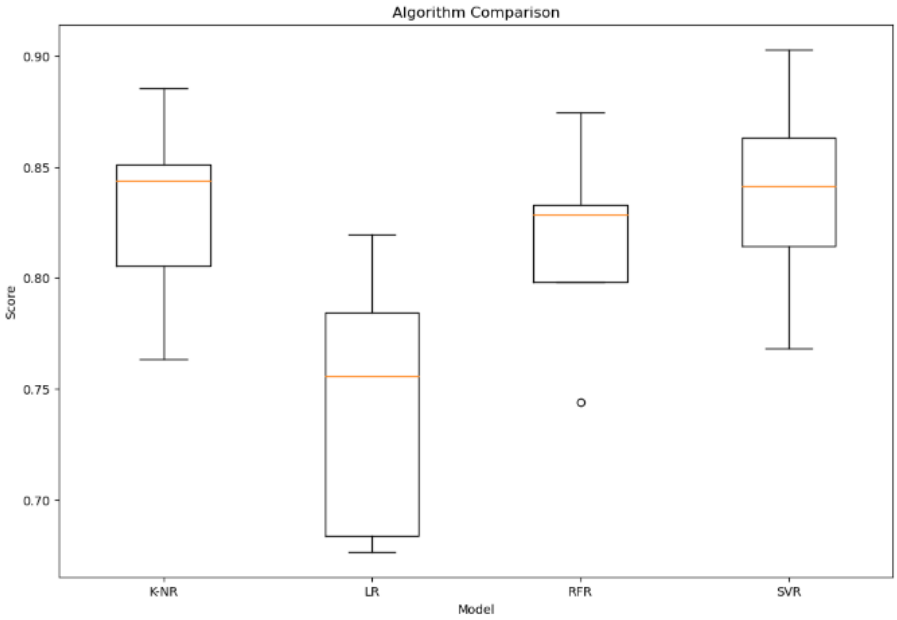
MODEL	MEAN	STANDARD DEVIATION
K-Neighboard Regressor	0.829749	0.041888
Linear Regression	0.744046	0.055810
Random Forest Regressor	0.815593	0.043252
Support Vector Regressor	0.837977	0.045294

Machine Learning Modelling Outcomes

A comparison of the models' performance was visualized using a boxplot. The Support Vector Regressor consistently outperformed the other models, indicating it is the most suitable model for this dataset.

The Support Vector Regressor (SVR) with specific hyperparameters (C=10, epsilon=0.1) provided the best performance on the test set with a score of 0.87.

ALGORITHM COMPARISON



Conclusion

In this report, we have effectively used this healthcare insurance dataset to identify an individual's medical expenditures with the help of various machine learning algorithms. In data preparation there was cleaning of missing values, renaming of variables and categorical variables encoding. Normalization of the data set and feature scaling were done in order to make the data appropriate for the classifiers.

Exploratory data analysis enable us have an insight of the distribution and relationships in the data set. To reduce the dimensionality of the data pre-processing of dataset PCA and LDA were employed. Both supervised and unsupervised ML algorithms machine learning models were trained and tested to tune hyparpameter vectors using GridSearchCV and the performance was further validate using K-Fold cross validation.

In all the models compared, it was realized that SVR had the highest accuracy and therefore was best suited for the task of healthcare insurance charge prediction in this set. Possible future work could be focused on the inclusion of other features, investigating more complex algorithms, and integrating more up-to-date information in order to improve the models' forecasting capabilities.

The analysis demonstrated that the Support Vector Regressor is the best-performing model for predicting individual medical costs billed by health insurance. Key findings from the data include the significant impact of **Smoker** status, **BMI**, and **Age** on medical costs. Proper handling of missing values and careful feature engineering contributed to the models' performance.

References

Müller, A.C. and Guido, S., 2016. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Inc.

Centers for Disease Control and Prevention (CDC), n.d. *Adult BMI Calculator*. Available at: https://www.cdc.gov/bmi/adult-calculator/bmi-categories.html [Accessed 28 July 2024].

Scikit-learn documentation, n.d. *scikit-learn: Machine Learning in Python*. Available at: https://scikit-learn.org/stable/ [Accessed 28 July 2024].

Github Repository

https://github.com/richardags-cct-college-dublin/RichardGonzalez_DP_ML_HDip_CA/