National Institute of Technology Hamirpur

Computer Science and Engineering

Report of

Medical Cost Detection System Using Machine Learning Techniques

of

Machine Learning

CS-652

Submitted to:
Dr. Kamlesh Dutta

Submitted By:
Name: Kinshuk Sharely
Roll No : 20DCS024
Semester : 7th

Table of Contents

Medical Insurance Cost Detection System

Abstract

The purpose of this project is to figure out how much money a person should pay for medical insurance that will help them during difficult times. Sometimes, the cost changes a lot based on different things. So we're using a smart ML algorithms and Methodologies to guess how much someone should pay.

1. Introduction

Insurance companies want to be smart about who they talk to. They want to find people who might want to buy insurance. Our project uses special computer tools to look at lots of information and predict how much someone should pay for insurance. This helps insurance companies save time and money.

This project, titled "Medical Cost Detection System," harnesses the power of machine learning to predict insurance expenses borne by individuals.

2. Objective

- Address fairness issues to ensure that the calculated insurance costs are equitable and unbiased across different groups.
- Help insurance companies save time and money by using this system to figure out insurance costs instead of doing it all by hand.

3. Software/hardware

This project will be implemented using Python programming language in which we'll be making use of various libraries like Pandas, Numpy, Scikit-learn, Matplotlib and Seaborn.

The hardware that will be used for this project will be a machine having configuration tailored to enhance prediction accuracy, and support seamless model integration, ensuring the project's success in delivering accurate medical insurance cost estimates

Algorithms that will be used :

- **Multiple Regression:** A simple algorithm for predicting numerical values based on input features.
- **Decision Trees:** Useful for capturing complex relationships in the data and making decisions based on a tree-like structure.
- **Random Forest:** An ensemble technique that combines multiple decision trees for improved accuracy and generalization.

4. Methodology

Steps involved are as follows:

1. Data Collection
2. Data Preprocessing
3. Feature Engineering
4. Model Selection and Training
5. Model Evaluation using Metrics

4.1 Data Collection

An important step of this project is the selection of a suitable dataset for training and testing. Gather policyholder details, including demographics, contact information, and policy specifics. Ensure data privacy and legal compliance while gathering and handling sensitive information.

4.2 Data Analysis and Preprocessing

In order to ensure the quality of dataset Data preprocessing is done. Preprocessing includes various methods like checking for NULL value, finding outliers, finding imbalances, Scaling, etc.

4.2.1 Dataset – 01

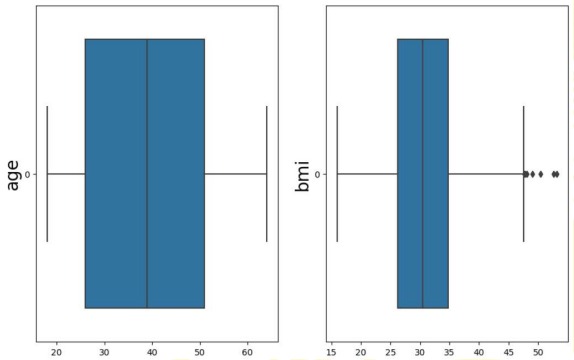
Descriptive Statistics

	age	bmi	children	charges
count	2772.000000	2772.000000	2772.000000	2772.000000
mean	39.109668	30.701349	1.101732	13261.369959
std	14.081459	6.129449	1.214806	12151.768945
min	18.000000	15.960000	0.000000	1121.873900
25%	26.000000	26.220000	0.000000	4687.797000
50%	39.000000	30.447500	1.000000	9333.014350
75%	51.000000	34.770000	2.000000	16577.779500
max	64.000000	53.130000	5.000000	63770.428010

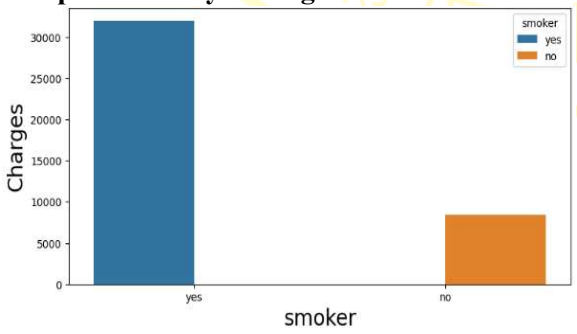
Missing Values

age	0
sex	0
bmi	0
children	0
smoker	0
region	0
charges	0

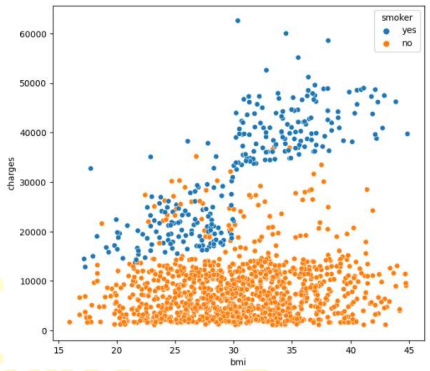
Box Plot to identify Outliers in Numeric Columns

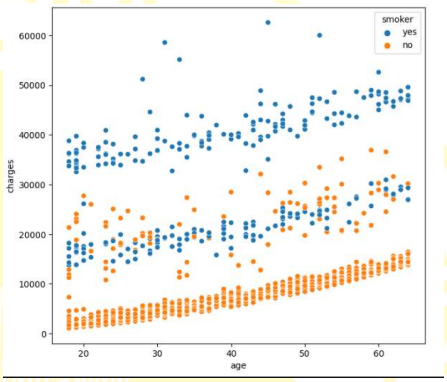


Bar plot to Analyze the given data

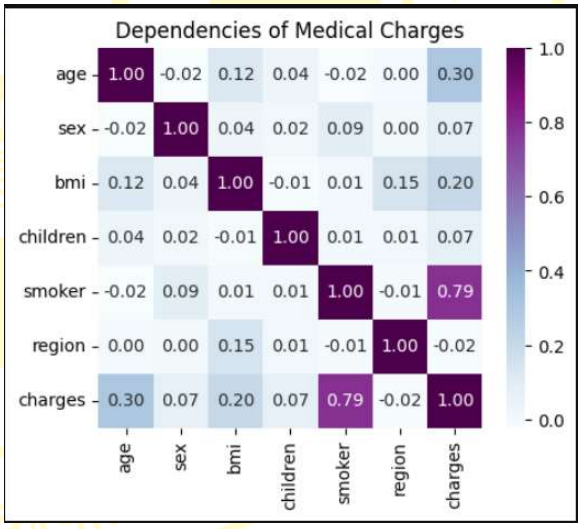


Scatter Plot (Age, BMI vs Charges)

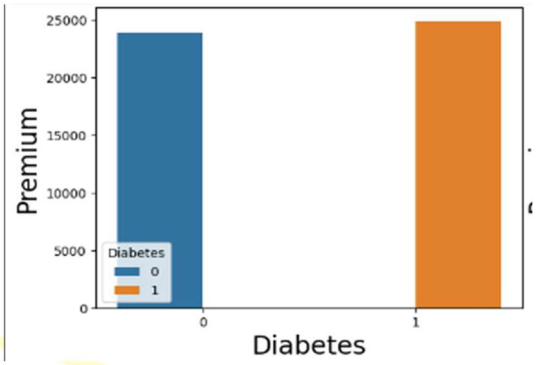




Correlation



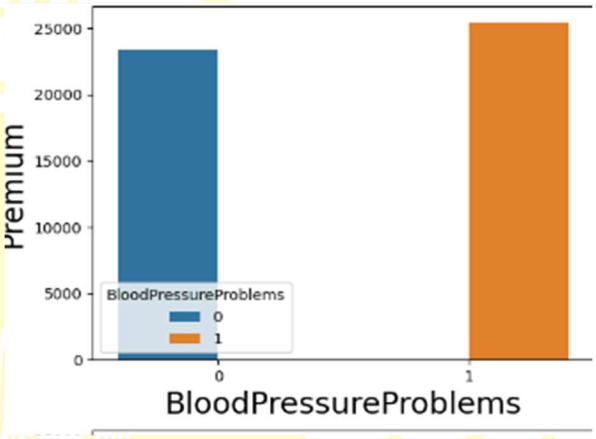
4.2.2 Dataset – 02



Descriptive Statistics

Age	Diabetes	BloodPressureProblems	AnyTransplants	AnyChronicDiseases	Height
45	0	0	0	0	155
60	1	0	0	0	180
36	1	1	0	0	158
52	1	1	0	1	183
38	0	0	0	1	166

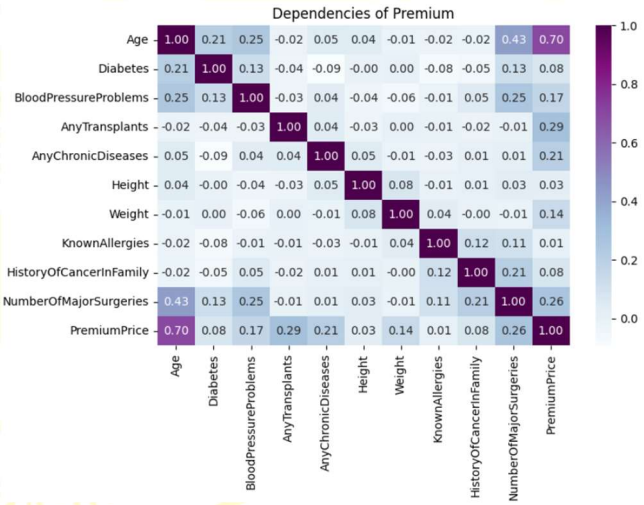
Weight	KnownAllergies	HistoryOfCancerInFamily	NumberOfMajorSurgeries	PremiumPrice
57	0	0	0	25000
73	0	0	0	29000
59	0	0	1	23000
93	0	0	2	28000
88	0	0	1	23000



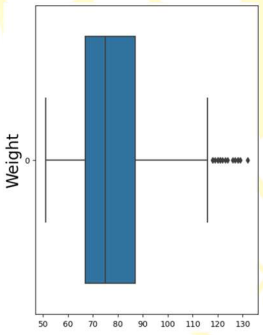
Missing Values

```
Age                        0
Diabetes                   0
BloodPressureProblems      0
AnyTransplants             0
AnyChronicDiseases         0
Height                     0
Weight                     0
KnownAllergies             0
HistoryOfCancerInFamily    0
NumberOfMajorSurgeries     0
PremiumPrice               0
dtype: int64
```

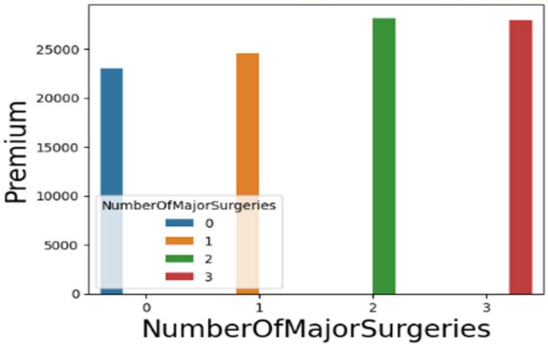
Correlation



Box Plot to identify Outliers in Numeric Columns



Bar plot to Analyze the given data



Premium is highly dependent on :

- **Age**
- **Blood Pressure Problems**
- **Any Transplants**
- **Any Chronic Disease**
- **Weight**
- **No. Of major Surgeries**

4.2.3 Dataset – 03

Descriptive Statistics

	Age	Health_Status	Smoker	Lifestyle	BMI	Chronic_Conditions
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	43.703000	2.516000	0.493000	1.956000	26.690437	0.524000
std	15.419876	1.120267	0.500201	0.820205	4.943427	0.499674
min	18.000000	1.000000	0.000000	1.000000	18.008184	0.000000
25%	30.000000	1.750000	0.000000	1.000000	22.458285	0.000000
50%	44.000000	3.000000	0.000000	2.000000	26.858508	1.000000
75%	58.000000	4.000000	1.000000	3.000000	31.028636	1.000000
max	70.000000	4.000000	1.000000	3.000000	34.999354	1.000000

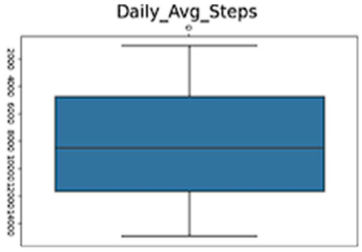
Family_History	Annual_Income	Region	Num_Dependents	Previous_Claims	Exercise_Frequency
1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
0.479000	84655.433475	1.979000	2.589000	0.508000	2.025000
0.499809	36809.780755	0.814385	1.702051	0.500186	0.825261
0.000000	20139.657164	1.000000	0.000000	0.000000	1.000000
0.000000	52550.052637	1.000000	1.000000	0.000000	1.000000
0.000000	85770.931293	2.000000	3.000000	1.000000	2.000000
1.000000	115620.424154	3.000000	4.000000	1.000000	3.000000
1.000000	149919.374605	3.000000	5.000000	1.000000	3.000000

Cholesterol_Level	Daily_Avg_Steps	Alcohol	Fat_Percentage	Insurance_Premium
1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
1.512000	8250.016000	30.012694	0.250193	3347.715679
0.500106	3998.908596	16.939729	0.086850	559.057329
1.000000	1035.000000	0.057014	0.100042	1538.602012
1.000000	4761.500000	15.846028	0.176104	2962.714409
2.000000	8488.000000	29.915199	0.249973	3357.318218
2.000000	11712.250000	43.828531	0.324704	3747.359247
2.000000	14994.000000	59.986528	0.399537	4976.643199

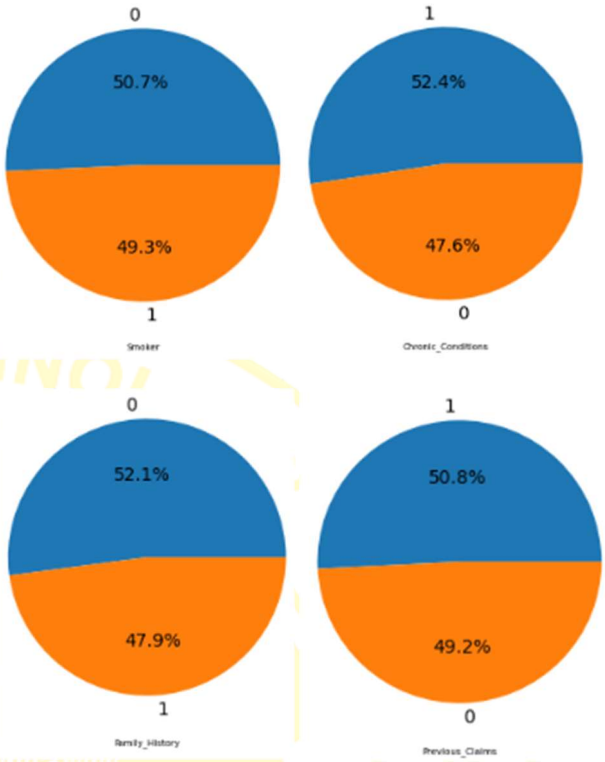
Missing Values

```
Age                    0
Health_Status          0
Smoker                 0
Lifestyle              0
BMI                    0
Chronic_Conditions     0
Family_History         0
Annual_Income          0
Region                 0
Num_Dependents         0
Previous_Claims        0
Exercise_Frequency     0
Cholesterol_Level      0
Daily_Avg_Steps        0
Alcohol                0
Fat_Percentage         0
Insurance_Premium      0
```

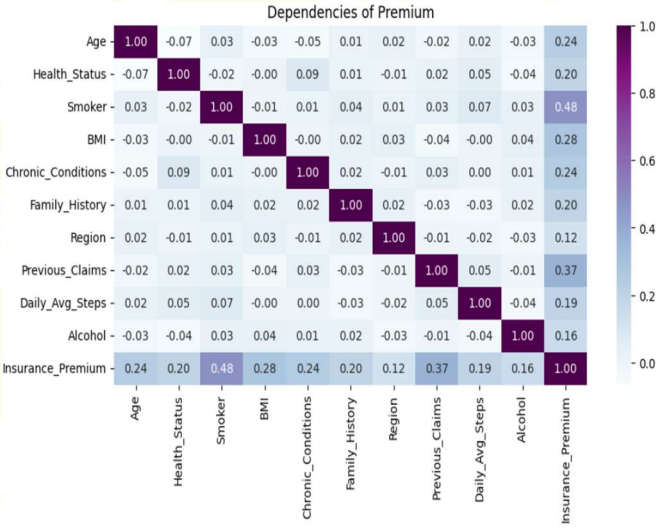
Box Plot to identify Outliers in Numeric Columns



Checking Imbalanced Data



Correlation



Premium is highly dependent on :

- **Age**
- **Health Status**
- **Smoker**
- **Chronic Conditions**
- **BMI**
- **Family History**
- **Previous Claims**
- **Daily Average Steps**
- **Alcohol Consumption**

4.3 Feature Engineering

Includes the transformation of text data into numerical data so that ML algorithms can process. Methods like LabelEncode , HotEncoder etc.

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

data['sex'] = le.fit_transform(data['sex'])
data['region'] = le.fit_transform(data['region'])
data['smoker'] = le.fit_transform(data['smoker'])
```

4.4 Model Selection and Training

1. Linear Regression

- Type: Supervised Learning, Regression
- Applicability: Predicting numerical values based on linear relationships.
- Pros: Simplicity, interpretability, and quick training.
- Cons: Assumes linear relationships, may not handle complex data well.

2. Decision Trees

- Type: Supervised Learning, Regression
- Applicability: Predicting numerical values based on decision rules.
- Pros: Handles complex, non-linear relationships. Interpretable.
- Cons: Prone to overfitting, can be sensitive to data variations.

3. Random Forest

- Type: Supervised Learning, Regression (Ensemble)
- Applicability: Predicting numerical values by aggregating multiple decision trees.
- Pros: Improved accuracy, handles overfitting, feature importance.
- Cons: Complexity, longer training time, and less interpretability compared to a single decision tree.

4. Gradient Boost

- Type: Supervised Learning, Regression (Ensemble)
- Applicability: Predicting numerical values using gradient boosting.
- Pros: High accuracy, handles complex relationships, robust to outliers.
- Cons: Can be computationally expensive, tuning parameters is critical.

4.5 Evaluation Metrics

1. Mean Absolute Error

It is a metric that calculates the average of the absolute differences between predicted and actual values, providing a measure of the model's prediction accuracy in the same units as the target variable.

2. Mean Absolute Percentage Error

It measures the average percentage difference between predicted and actual values, providing a relative assessment of prediction accuracy that's particularly useful when comparing models across different datasets or units.

3. R2 Score

It is a statistical metric that quantifies the proportion of the variance in the dependent variable explained by the independent variables in a regression model, with higher values indicating a better fit of the model to the data.

4. Adjusted R2 Score

It is a modification of the R-squared (R2) score in regression analysis that accounts for the number of predictors in the model, helping to evaluate the model's goodness of fit while penalizing excessive complexity.

4.6 Training and Results

4.6.1. Data 1 & more

```
lr=LinearRegression()
model = lr.fit(X_train,y_train)

y_predicted = model.predict(X_test)
```

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(random_state = 0)
model1 = regressor.fit(X_train, y_train)

predrandom = model1.predict(X_test)
```

```
from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor(random_state = 0)
model3 = dt.fit(X_train, y_train)

predTree = model3.predict(X_test)
```

```
from sklearn.ensemble import GradientBoostingRegressor

gb = GradientBoostingRegressor()
model4 = gb.fit(X_train, y_train)
predGb = model4.predict(X_test)
```

Similarly for all models and Data sets

4.6.2 Results

1. Dataset 1

MODELs	MAE (Mean Absolute Error)		MAPE (Mean absolute percentage error)		R2 Score		Adjusted R2 Score	
	Train	Test	Train	Test	Train	Test	Train	Test
Linear Regression	4205.07	4138.88	0.439	0.4252	0.745	0.7545	0.741	0.75
Decision Tree	13.68	918.94	0.004	0.0836	0.999	0.915	0.99	0.91
Random Forest	569.29	1429.57	0.074	0.153	0.989	0.927	0.98	0.92
Gradient Boost	2051.28	2214.9	0.259	0.258	0.899	0.878	0.89	0.87

2. Dataset 2

MODELs	MAE (Mean Absolute Error)		MAPE (Mean absolute percentage error)		R2 Score		Adjusted R2 Score	
	Train	Test	Train	Test	Train	Test	Train	Test
Linear Regression	2672.92	2782.35	0.114	0.119	0.627	0.631	0.621	0.626
Decision Tree	12.36	1275	0.0004	0.049	0.998	0.663	0.99	0.658
Random Forest	478.27	1415.5	0.019	0.058	0.965	0.773	0.96	0.769
Gradient Boost	1156.3	1585.01	0.046	0.062	0.878	0.79	0.87	0.788

3. Dataset 3

MODELs	MAE (Mean Absolute Error)		MAPE (Mean absolute percentage error)		R2 Score		Adjusted R2 Score	
	Train	Test	Train	Test	Train	Test	Train	Test
Linear Regression	265.4	262.01	0.081	0.081	0.685	0.670	0.6804	0.666
Decision Tree	0.0	414.4	0.0	0.128	1.0	0.106	1.0	0.092
Random Forest	111.97	291.1	0.035	0.091	0.94	0.565	0.939	0.559
Gradient Boost	189.04	278.34	0.058	0.08	0.83	0.61	0.827	0.606

5. Result Comparison

MODELs	Dataset-1 (Accuracy)		Dataset-2 (Accuracy)		Dataset-3 (Accuracy)	
	Train	Test	Train	Test	Train	Test
Linear Regression	0.745	0.7545	0.627	0.631	0.685	0.670
Decision Tree	0.999	0.915	0.998	0.663	1.0	0.106
Random Forest	0.989	0.927	0.965	0.773	0.94	0.565
Gradient Boost	0.899	0.878	0.878	0.79	0.83	0.61

6. Conclusion

In summary, our project on using machine learning to predict insurance premiums has shown that we can make better guesses about how much people should pay for insurance.

By using lots of different information and different datasets, we created models that can give fairer prices to customers. This helps people get insurance that suits them better and also helps insurance companies make smarter decisions.

However, we need to be careful about keeping people's data safe and following the rules. As technology gets better, our project is a step toward making insurance prices even more accurate and fair.

7. Future Direction and Scope

In the future, the main aim is to make the insurance premium predictions even better by tailoring insurance plans to suit each person's unique needs.

We can use up-to-the-minute information and adjust prices as things change. We also can make use of ChatBots to have quick and helpful assistance.

To improve accuracy, we can look into other models like XGBoost and Neural Networks. These changes will not only make user experience better but also help in better management of risks and follow the rules.

Additionally, we can work on keeping the data extra safe using a technology called blockchain and collaborate with experts from different fields to get more insights.

8. Reference Papers

[1] K. Bhatia, S. S. Gill, N. Kamboj, M. Kumar and R. K. Bhatia, "Health Insurance Cost Prediction using Machine Learning," *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824201.

[2] Sahu, Ajay and Sharma, Gopal and Kaushik, Janvi and Agarwal, Kajal and Singh, Devendra, Health Insurance Cost Prediction by Using Machine Learning (February 22, 2023). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022

[3] Hanafy, Mohamed. (2021). Predict Health Insurance Cost by using Machine Learning. International Journal of Innovative Technology and Exploring Engineering. Volume-10. 137. 10.35940/ijitee.C8364.0110321

9. DataSets

- https://www.kaggle.com/datasets/harishkumardatalab/medical-insurance-price-prediction
- https://www.kaggle.com/datasets/tejashvi14/medical-insurance-premium-prediction
- https://www.kaggle.com/datasets/Dataglod/insurance-dataset