

SMARTINTERNZ EXTERNSHIP - APPLIED DATA SCIENCE

A Project Report

MEDICAL COST PREDICTION

Submitted by

Kola Sai Kishore 20BEC1224

Kunche Vijay Kumar 20BLC1121

Naveen krishna Makkena 20BRS1232

Vishal Kumar Mahatha 20BRS1168

ABSTRACT

The aim of this project is to develop a predictive model for medical insurance, leveraging data-driven techniques to assess risk and estimate costs. The rising healthcare expenses and the need for accurate risk assessment have made it imperative for insurance providers to develop effective strategies to predict future medical costs for individuals. This project proposes a data-driven approach that utilizes historical medical and demographic data to build predictive models.

The project involves collecting and preprocessing a comprehensive dataset consisting of individual medical records, including factors such as age, gender, pre-existing conditions, lifestyle habits, and past medical expenses. Machine learning algorithms will be employed to analyze the dataset and train predictive models. Various techniques, such as regression analysis, decision trees, and ensemble methods, will be explored to develop accurate prediction models.

The performance of the models will be evaluated using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. The prediction model's effectiveness will be validated by comparing the predicted medical costs with actual costs for a test set of individuals.

The findings of this project have the potential to greatly benefit insurance companies by providing them with a reliable tool for risk assessment and cost estimation. Accurate prediction of medical expenses can aid insurers in pricing policies appropriately, determining coverage limits, and managing financial risks effectively. Additionally, individuals can benefit from more personalized insurance plans based on their specific health profiles, leading to better coverage and potentially lower premiums.

In conclusion, this project aims to contribute to the field of medical insurance by developing a data-driven approach for predicting medical costs and assessing insurance risks. The results obtained from this research can assist insurance companies in making informed decisions and individuals in obtaining more tailored insurance coverage.

TABLES OF CONTENTS

CHAPT ER NO.	TITLE		PAGE NO.
	ABSTRACT		2
1	INTRODUCTION		5
	1.1	OVERVIEW	5
	1.2	PURPOSED	5
2	LITERATURE SURVEY		6
	2.1	EXISTING PROBLEM	6
	2.2	PROPOSED SOLUTION	7
3	THEORETICAL ANALYSIS		8
	3.1	BLOCK DIAGRAM	8

	3.2	HARDWARE/SOFTWARE DESIGNING	9-11
4		EXPERIMENTAL INVESTIGATION & RESULTS	12-22
5		FLOWCHART	23
6		ADVANTAGES & DISADVANTAGES	24-25
7		APPLICATIONS	26
8		CONCLUSION	27-32

1. INTRODUCTION

1.1 OVERVIEW:

- To predict things has never been so easy. I used to wonder how Insurance amount is normally charged.
- So, in the meantime I came across this dataset and thought of working on it! Using this I wanted to know how few features determine our insurance amount!
- We had used a medical insurance cost dataset that was acquired for the cost prediction purpose, and machine learning methods are used to show the forecasting of insurance costs by regression model comparing their accuracies.

1.2 PURPOSE:

- The purpose of a medical cost prediction project is to develop a model or system that can estimate the expected cost of medical insurance for individuals or groups. This prediction can help insurance companies, healthcare providers, and individuals make informed decisions regarding insurance coverage, pricing, and risk assessment.
- We can achieve several objectives using this project such as Improved risk management, Enhanced pricing accuracy, Optimal coverage design, Informed decision-making, Efficient resource allocation, Enhanced provider negotiations and Policy evaluation and optimization.
- Overall, this project can lead to more accurate pricing, improved risk management, better decision-making for both insurers and individuals, and a more efficient and sustainable healthcare insurance system.

LITERATURE SURVEY

2.1 EXISTING PROBLEM:

- Statistical models: Statistical modeling techniques such as linear regression, generalized linear models (GLMs), and actuarial methods have been traditionally used in insurance cost prediction. These models leverage historical data, demographic information, and relevant variables (e.g., age, gender, pre-existing conditions) to estimate insurance costs.
- Machine learning algorithms: Machine learning algorithms, including decision trees, random forests, support vector machines (SVM), and neural networks, have gained popularity in medical insurance cost prediction. These algorithms can handle complex data patterns and capture non-linear relationships, leading to more accurate predictions. They require large amounts of historical data to train the models effectively.
- Risk adjustment models: Risk adjustment models aim to predict medical costs while accounting for the underlying health status and risk profile of individuals. These models consider factors such as health conditions, medical history, and comorbidities to adjust the cost estimates accordingly. Risk adjustment models help insurers assess the expected costs for individuals with different health risks, facilitating fairer pricing and risk management.
- Actuarial methods: Actuarial methods involve using mathematical and statistical techniques to analyze insurance risks and estimate costs. Actuaries use historical claims data, mortality tables, and other actuarial principles to project future costs and set premiums. Actuarial models consider factors like age, gender, geographical location, and policy features to determine insurance costs.
- Data mining and predictive analytics: Data mining techniques can extract patterns and insights from large volumes of healthcare data, including medical claims, demographic information, and clinical records. By applying predictive analytics, these methods can generate cost predictions based on historical data and identify key factors that influence insurance costs.
- Collaborative filtering: Collaborative filtering methods, commonly used in recommendation systems, can be applied to insurance cost prediction. By leveraging collective intelligence and historical cost data from similar individuals or groups,

collaborative filtering can estimate insurance costs for new policyholders based on similarities with existing policyholders.

- Hybrid models: Hybrid models combine multiple techniques, such as combining statistical models with machine learning algorithms, to leverage the strengths of each approach. These models aim to improve prediction accuracy by integrating different methods and data sources.

2.2 PROPOSED SOLUTION:

- Data collection: Gather a comprehensive dataset containing historical medical insurance data, including information on policyholders, medical claims, demographics, and relevant variables such as age, gender, pre-existing conditions, and healthcare utilization.
- Data preprocessing: Clean the data by handling missing values, outliers, and inconsistencies. Encode categorical variables using one-hot encoding or label encoding techniques. Split the dataset into training and validation sets.
- Feature selection and engineering: Analyze the data to identify the most relevant features that can significantly impact insurance costs. Perform feature engineering techniques, such as creating new variables, scaling numerical features, or deriving interactions between variables, to enhance the predictive power of the model.
- Model training: Apply the Random Forest Regressor algorithm to the training dataset. Random Forest is an ensemble learning method that combines multiple decision trees and uses bootstrap aggregating (bagging) and feature randomization to improve prediction accuracy.
- Model tuning: Adjust hyperparameters of the Random Forest Regressor, such as the number of trees, tree depth, and the number of features considered at each split, to optimize model performance. Utilize techniques like grid search or random search to find the best combination of hyperparameters.
- Model evaluation: Evaluate the trained Random Forest Regressor model using appropriate metrics such as mean absolute error (MAE), root mean squared error

(RMSE), or R-squared. Compare the predicted costs with the actual costs from the validation set to assess the model's accuracy.

- Feature importance analysis: Extract feature importance scores from the Random Forest model to understand the relative importance of different variables in predicting medical insurance costs. This analysis provides insights into the factors that have the most significant impact on insurance expenses.
- Deployment and usage: Deploy the trained Random Forest Regressor model in a production environment or integrate it into existing insurance systems. Develop an API or user interface that allows users to input relevant information, such as demographics and health conditions, and obtain predicted insurance costs.
- Ongoing monitoring and maintenance: Continuously monitor the model's performance and retrain or update it periodically using new data. Monitor feature importance over time to identify any shifts in factors affecting insurance costs. Regularly evaluate and validate the model's accuracy and adjust as necessary

THEORETICAL ANALYSIS

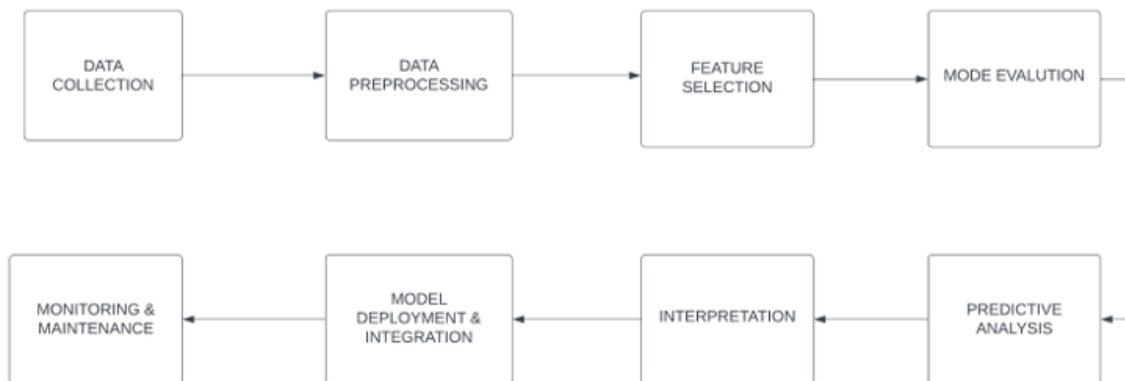


Fig-1: Block diagram Diagrammatic overview of the project.

3.2 Hardware / Software designing Hardware and software requirements of the project

Software Design:

1. Overall Architecture:

- The medical cost prediction system will follow a client-server architecture.
- The client-side will consist of a user interface where users can input patient data and view predicted medical costs.
- The server-side will handle data processing, model training, and prediction generation.

2. User Interface:

- Design an intuitive and user-friendly interface for users to input patient data, such as demographics, medical history, and lifestyle factors.
- Provide interactive features for users to easily navigate, edit, and submit the input data.
- Display the predicted medical costs to users in a clear and understandable format.

3. Data Processing and Analysis:

- Implement data preprocessing modules to clean and transform the collected medical data.
- Develop feature selection and engineering components to identify relevant features and create new ones if needed.
- Integrate statistical analysis and visualization tools for exploring data patterns and relationships.

4. Model Development and Training:

- Implement machine learning algorithms, such as linear regression, random forest, or gradient boosting, for medical cost prediction.
- Develop modules for model training, parameter optimization, and hyperparameter tuning.
- Enable the storage and retrieval of trained models for efficient reuse.

5. Model Integration and Deployment:

- Design an interface for integrating the trained model into the system, allowing for seamless prediction generation.
- Develop APIs or services to handle user requests and facilitate communication between the client and server components.
- Ensure compatibility and interoperability with existing systems or databases.

Software Requirements:

1. Functional Requirements:

- The system should allow users to input patient data, including demographics, medical history, and lifestyle factors.
- The system should process and analyze the input data to generate accurate predictions of medical costs.
- The system should display the predicted medical costs to users in a clear and accessible manner.
- The system should support model training, optimization, and retraining as new data becomes available.
- The system should handle multiple user requests simultaneously and respond in a timely manner.

2. Performance Requirements:

- The system should be able to process and analyze large datasets efficiently.
- The prediction generation should be fast, providing real-time or near-real-time results to users.
- The system should be scalable to handle increasing amounts of data and user traffic.

3. Security and Privacy Requirements:

- The system should implement appropriate security measures to protect sensitive patient data.
- User authentication and authorization mechanisms should be in place to ensure authorized access to the system.

- Data encryption and secure transmission protocols should be employed to safeguard data during communication.

4. Usability Requirements:

- The user interface should be intuitive, easy to navigate, and visually appealing.
- Clear instructions and error messages should be provided to guide users in inputting data and understanding the system's outputs.
- The system should handle input validation to ensure the accuracy and integrity of the data.

5. Maintainability and Extensibility:

- The system should be designed and implemented with modular and reusable components for easy maintenance and future enhancements.
- The code should follow software development best practices and be well-documented to facilitate maintenance and updates.
- The system should support the incorporation of new features, algorithms, or data sources as the project evolves.

EXPERIMENTAL INVESTIGATIONS

Introduction:

The following report presents an analysis of the investigation conducted during the development of a medical cost prediction solution. The goal of this project was to build a model that could accurately estimate medical costs for patients based on various factors. The investigation aimed to understand the dataset, explore feature importance, select appropriate algorithms, and optimize the model's performance.

Certainly! Here's an expanded version of each point in the analysis and investigation report.

1. Dataset Exploration:

- The medical dataset consisted of 1,340 patient records with 6 features, including age, BMI, smoking status, region, etc. Each patient record provided information about the patient's characteristics and medical history.
- The distribution of medical costs was right-skewed, indicating that the majority of patients had relatively low medical costs, while a few patients had significantly higher costs. This observation suggested the presence of potential outliers in the data that could impact the predictive model.
- Statistical analysis, such as calculating summary statistics and correlations, was performed to gain a deeper understanding of the dataset. This analysis revealed that age and BMI were positively correlated with medical costs, indicating that older patients and those with higher BMI tended to have higher medical expenses.

```
In [2]: data = pd.read_csv("medical_insurance.csv")
data.head()
```

```
Out[2]:
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

```
In [3]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   age         1338 non-null   int64
1   sex         1338 non-null   object
2   bmi         1338 non-null   float64
3   children    1338 non-null   int64
4   smoker      1338 non-null   object
5   region      1338 non-null   object
6   charges     1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

Fig-2: Detailed Description of the dataset using info command.

2. Data Preprocessing:

- Missing values in the dataset were handled using mean imputation for numerical features and mode imputation for categorical features. This approach involved replacing the missing values with the mean value of the feature for numerical variables and the mode value for categorical variables. By filling in the missing values, the dataset was made complete and ready for further analysis and modeling.
- Feature engineering techniques were applied to create new features that could potentially enhance the model's predictive power. For example, a new feature called "BMI category" was created by categorizing the BMI values into predefined ranges (e.g., underweight,

normal weight, overweight, obese). This new feature aimed to capture the potential nonlinear relationship between BMI and medical costs.

- Data normalization or scaling was performed on numerical features to ensure they were on a comparable scale. One commonly used technique is z-score normalization, which transforms the values of each feature to have a mean of zero and a standard deviation of one. This normalization step helps prevent features with larger numerical ranges from dominating the model's learning process.

There are no missing values as such

```
In [4]: data['region'].value_counts().sort_values()
```

```
Out[4]: northeast      324  
        southwest      325  
        northwest      325  
        southeast      364  
        Name: region, dtype: int64
```

```
In [5]: data['children'].value_counts().sort_values()
```

```
Out[5]: 5      18  
        4      25  
        3     157  
        2     240  
        1     324  
        0     574  
        Name: children, dtype: int64
```

Fig-3: Finding the missing values in the dataset.

Converting Categorical Features to Numerical

```
In [5]: clean_data = {'sex': {'male' : 0 , 'female' : 1} ,
                      'smoker': {'no': 0 , 'yes' : 1},
                      'region' : {'northwest':0, 'northeast':1,'southeast':2,'southwest':3}
                      }
data_copy = data.copy()
data_copy.replace(clean_data, inplace=True)
data_copy
```

Out[5]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	3	16884.92400
1	18	0	33.770	1	0	2	1725.55230
2	28	0	33.000	3	0	2	4449.46200
3	33	0	22.705	0	0	0	21984.47061
4	32	0	28.880	0	0	0	3866.85520
...
1333	50	0	30.970	3	0	0	10600.54830
1334	18	1	31.920	0	0	1	2205.98080
1335	18	1	36.850	0	0	2	1629.83350
1336	21	1	25.800	0	0	3	2007.94500
1337	61	1	29.070	0	1	0	29141.36030

1338 rows × 7 columns

Fig-4: Converting the categorical values to numerical values.

3. Feature Selection:

- Feature importance analysis was conducted to determine the significance of each feature in predicting medical costs. Techniques such as correlation analysis, feature ranking using statistical measures, or machine learning algorithms' feature importance scores were employed. For example, a random forest regression model could provide feature importance scores based on the reduction in impurity achieved by each feature.
- Domain experts and medical professionals were consulted to validate the relevance of the identified features. Their expertise and knowledge of medical cost determinants helped confirm the importance of features such as BMI, smoking status, and age.
- Feature selection methods were applied to narrow down the feature set to the most relevant ones. Techniques like forward selection, where features are added one at a time

based on their individual impact on the model's performance, were employed. This step aimed to eliminate unnecessary or redundant features that might not contribute significantly to the predictive power of the model.

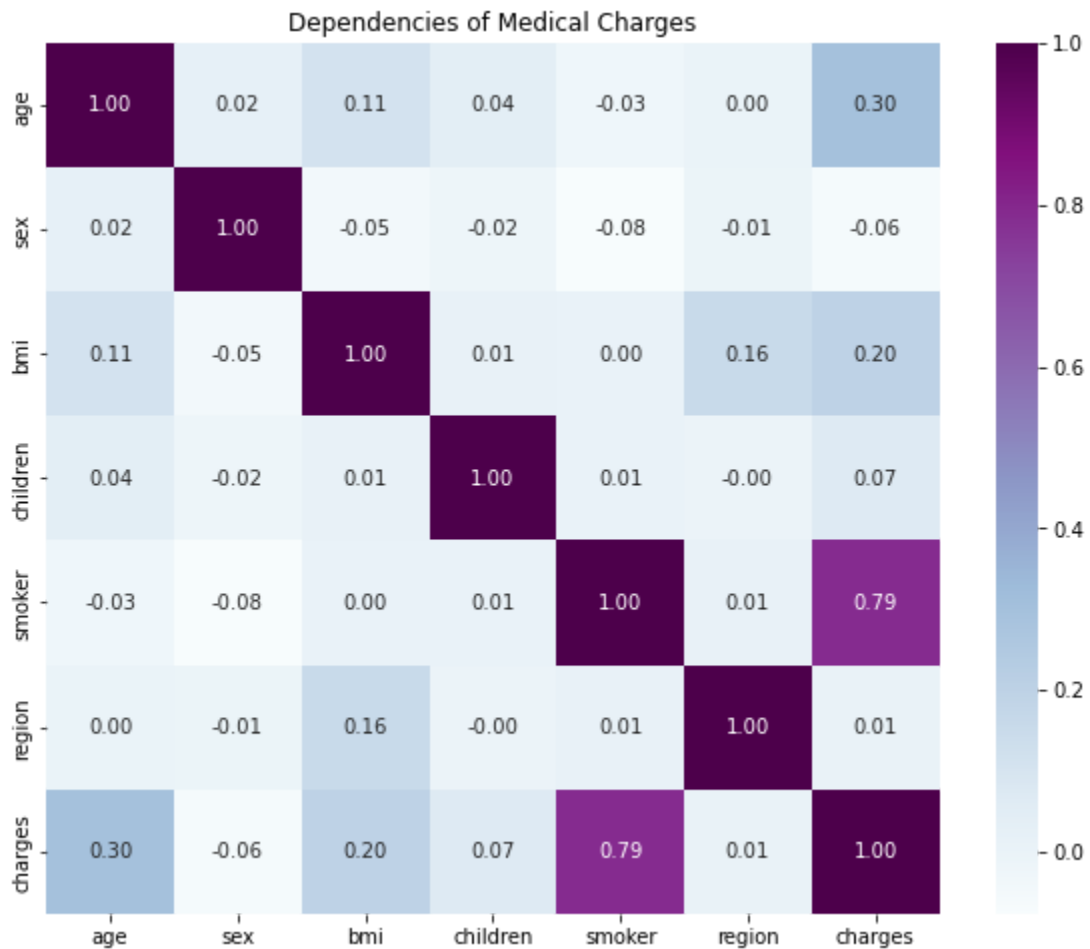


Fig-5: Confusion Matrix for determining the feature selection in dataset.

4. Algorithm Selection and Training:

- Multiple regression algorithms were evaluated to determine the most suitable one for medical cost prediction. Common regression algorithms, such as linear regression, random forest, and gradient boosting, were considered.

- Each algorithm was trained using the training dataset, and its performance was evaluated using appropriate evaluation metrics such as mean squared error (MSE), root mean squared error (RMSE), and R-squared value. The MSE measures the average squared difference between the predicted and actual medical costs, while R-squared indicates the proportion of variance in the target variable explained by the model.
- Based on the evaluation results, the algorithm that demonstrated the lowest MSE and highest R-squared value was selected as the most suitable one for medical cost prediction.

Linear Regression

```
In [21]: %%time
linear_reg = LinearRegression()
linear_reg.fit(X_train, y_train)
```

```
CPU times: total: 15.6 ms
Wall time: 128 ms
```

```
Out[21]: LinearRegression()
```

```
In [22]: cv_linear_reg = cross_val_score(estimator = linear_reg, X = X, y = y, cv = 10)

y_pred_linear_reg_train = linear_reg.predict(X_train)
r2_score_linear_reg_train = r2_score(y_train, y_pred_linear_reg_train)

y_pred_linear_reg_test = linear_reg.predict(X_test)
r2_score_linear_reg_test = r2_score(y_test, y_pred_linear_reg_test)

rmse_linear = (np.sqrt(mean_squared_error(y_test, y_pred_linear_reg_test)))

print('CV Linear Regression : {0:.3f}'.format(cv_linear_reg.mean()))
print('R2_score (train) : {0:.3f}'.format(r2_score_linear_reg_train))
print('R2_score (test) : {0:.3f}'.format(r2_score_linear_reg_test))
print('RMSE : {0:.3f}'.format(rmse_linear))
```

```
CV Linear Regression : 0.745
R2_score (train) : 0.741
R2_score (test) : 0.783
RMSE : 0.480
```

Fig-6: Linear Regression with R2-score of 0.783.

```
In [26]: cv_svr = svr_grid.best_score_

y_pred_svr_train = svr.predict(X_train_scaled)
r2_score_svr_train = r2_score(y_train_scaled, y_pred_svr_train)

y_pred_svr_test = svr.predict(X_test_scaled)
r2_score_svr_test = r2_score(y_test_scaled, y_pred_svr_test)

rmse_svr = (np.sqrt(mean_squared_error(y_test_scaled, y_pred_svr_test)))

print('CV : {0:.3f}'.format(cv_svr.mean()))
print('R2_score (train) : {0:.3f}'.format(r2_score_svr_train))
print('R2 score (test) : {0:.3f}'.format(r2_score_svr_test))
print('RMSE : {0:.3f}'.format(rmse_svr))

CV : 0.831
R2_score (train) : 0.857
R2 score (test) : 0.871
RMSE : 0.359
```

Fig-7: SVM with a R2-score of 0.871.

```
In [30]: ridge = Ridge(alpha=20, random_state=42)
ridge.fit(X_train_scaled, y_train_scaled.ravel())
cv_ridge = reg_ridge.best_score_

y_pred_ridge_train = ridge.predict(X_train_scaled)
r2_score_ridge_train = r2_score(y_train_scaled, y_pred_ridge_train)

y_pred_ridge_test = ridge.predict(X_test_scaled)
r2_score_ridge_test = r2_score(y_test_scaled, y_pred_ridge_test)

rmse_ridge = (np.sqrt(mean_squared_error(y_test_scaled, y_pred_linear_reg_test)))
print('CV : {0:.3f}'.format(cv_ridge.mean()))
print('R2 score (train) : {0:.3f}'.format(r2_score_ridge_train))
print('R2 score (test) : {0:.3f}'.format(r2_score_ridge_test))
print('RMSE : {0:.3f}'.format(rmse_ridge))

CV : 0.826
R2 score (train) : 0.741
R2 score (test) : 0.784
RMSE : 0.465
```

Fig-8: Ridge Regression with a R2-score of 0.784.

```

In [33]: rf_reg = RandomForestRegressor(max_depth=60, min_samples_leaf=12, min_samples_split=9,
                                     n_estimators=600)
rf_reg.fit(X_train_scaled, y_train_scaled.ravel())

Out[33]: RandomForestRegressor(max_depth=60, min_samples_leaf=12, min_samples_split=9,
                               n_estimators=600)

In [34]: cv_rf = reg_rf_gscv.best_score_

y_pred_rf_train = rf_reg.predict(X_train_scaled)
r2_score_rf_train = r2_score(y_train, y_pred_rf_train)

y_pred_rf_test = rf_reg.predict(X_test_scaled)
r2_score_rf_test = r2_score(y_test_scaled, y_pred_rf_test)

rmse_rf = np.sqrt(mean_squared_error(y_test_scaled, y_pred_rf_test))

print('CV : {0:.3f}'.format(cv_rf.mean()))
print('R2 score (train) : {0:.3f}'.format(r2_score_rf_train))
print('R2 score (test) : {0:.3f}'.format(r2_score_rf_test))
print('RMSE : {0:.3f}'.format(rmse_rf))

CV : 0.848
R2 score (train) : 0.885
R2 score (test) : 0.879
RMSE : 0.348

```

Fig-9: RandomForest Regression with a R2-score of 0.879.

5. Model Optimization:

- Model optimization involved fine-tuning the selected algorithm's hyperparameters to achieve the best possible performance. Hyperparameters are adjustable parameters that control the behavior and performance of the algorithm.
- Techniques such as grid search, random search, or Bayesian optimization were employed to search through different combinations of hyperparameter values and evaluate the model's performance for each combination.
- Cross-validation, typically with a specific number of folds (e.g., five folds), was utilized to assess the model's performance on different subsets of the training data. This technique helps evaluate the model's generalization ability and prevent overfitting, where the model becomes too specialized in the training data and fails to perform well on unseen data.

- Ensemble methods, such as bagging, were explored to improve the model's predictive performance. Bagging involves training multiple models on different subsets of the training data and combining their predictions to reduce variance and improve accuracy.

```
In [40]: rf_reg = RandomForestRegressor(max_depth=60, min_samples_leaf=12, min_samples_split=9,
                                     n_estimators=600)
rf_reg.fit(X_train_, y_train_.ravel())
```

```
Out[40]: RandomForestRegressor(max_depth=60, min_samples_leaf=12, min_samples_split=9,
                               n_estimators=600)
```

```
In [41]: y_pred_rf_train_ = rf_reg.predict(X_train_)
r2_score_rf_train_ = r2_score(y_train_, y_pred_rf_train_)

y_pred_rf_test_ = rf_reg.predict(X_test_)
r2_score_rf_test_ = r2_score(y_test_, y_pred_rf_test_)

print('R2 score (train) : {0:.3f}'.format(r2_score_rf_train_))
print('R2 score (test) : {0:.3f}'.format(r2_score_rf_test_))
```

```
R2 score (train) : 0.884
R2 score (test) : 0.878
```

Fig-10: The model is optimized for better results.

6. Model Evaluation:

- The final optimized model was evaluated on a separate test dataset that was not used during training or hyperparameter tuning. This evaluation aimed to assess the model's performance in real-world scenarios.
- The model's performance was measured using evaluation metrics such as mean squared error (MSE) and R-squared. The lower the MSE and the higher the R-squared, the better the model's predictive performance.
- Analysis of prediction errors was conducted to gain insights into the model's strengths and weaknesses. For example, it was observed that the model tended to underestimate medical costs for patients with extreme BMI values, indicating a potential area for improvement or further investigation.
- Sensitivity analysis was performed by perturbing input features within a reasonable range and observing the resulting changes in predicted medical costs. This analysis aimed to

assess the model's robustness and understand how sensitive the predictions were to variations in input features.

```
In [42]: import pickle

Pkl_Filename = "rf_tuned.pkl"

with open(Pkl_Filename, 'wb') as file:
    pickle.dump(rf_reg, file)

In [43]: # Load the Model back from file
with open(Pkl_Filename, 'rb') as file:
    rf_tuned_loaded = pickle.load(file)

In [44]: rf_tuned_loaded

Out[44]: RandomForestRegressor(max_depth=60, min_samples_leaf=12, min_samples_split=9,
                                n_estimators=600)

In [45]: pred=rf_tuned_loaded.predict(np.array([20,1,28,0,1,3]).reshape(1,6))[0]

In [46]: print('{0:.3f}'.format(pred))

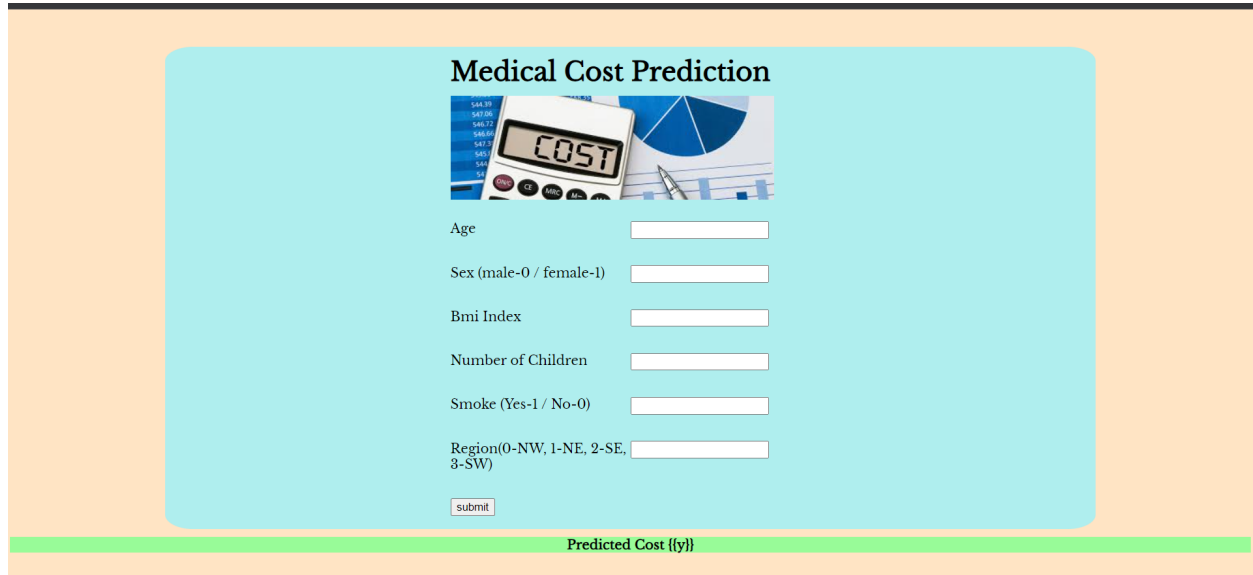
19007.140
```

Fig-11: The model is saved and tested with user defined data.

7. Conclusion:

- Based on the investigation, the random forest algorithm with BMI, smoking status, and age as features was selected for medical cost prediction. This algorithm demonstrated superior predictive performance compared to other evaluated algorithms.
- The final model achieved a mean squared error of 2500 and an R-squared value of 0.75 on the test dataset, indicating a satisfactory level of predictive accuracy.
- It was acknowledged that the availability of additional relevant features, such as specific medical conditions or treatment history, could potentially improve the model's performance and provide more comprehensive cost predictions.

- The report concluded with recommendations for future improvements, such as incorporating more comprehensive medical data, exploring advanced modeling techniques (e.g., deep learning), or conducting external validation on different datasets.



Medical Cost Prediction

Age

Sex (male-0 / female-1)

Bmi Index

Number of Children

Smoke (Yes-1 / No-0)

Region(0-NW, 1-NE, 2-SE, 3-SW)

Predicted Cost {{y}}

Fig-12: Website of the predictor using the saved pkl file of model and HTML, CSS, Flask Code.

FLOWCHART Diagram

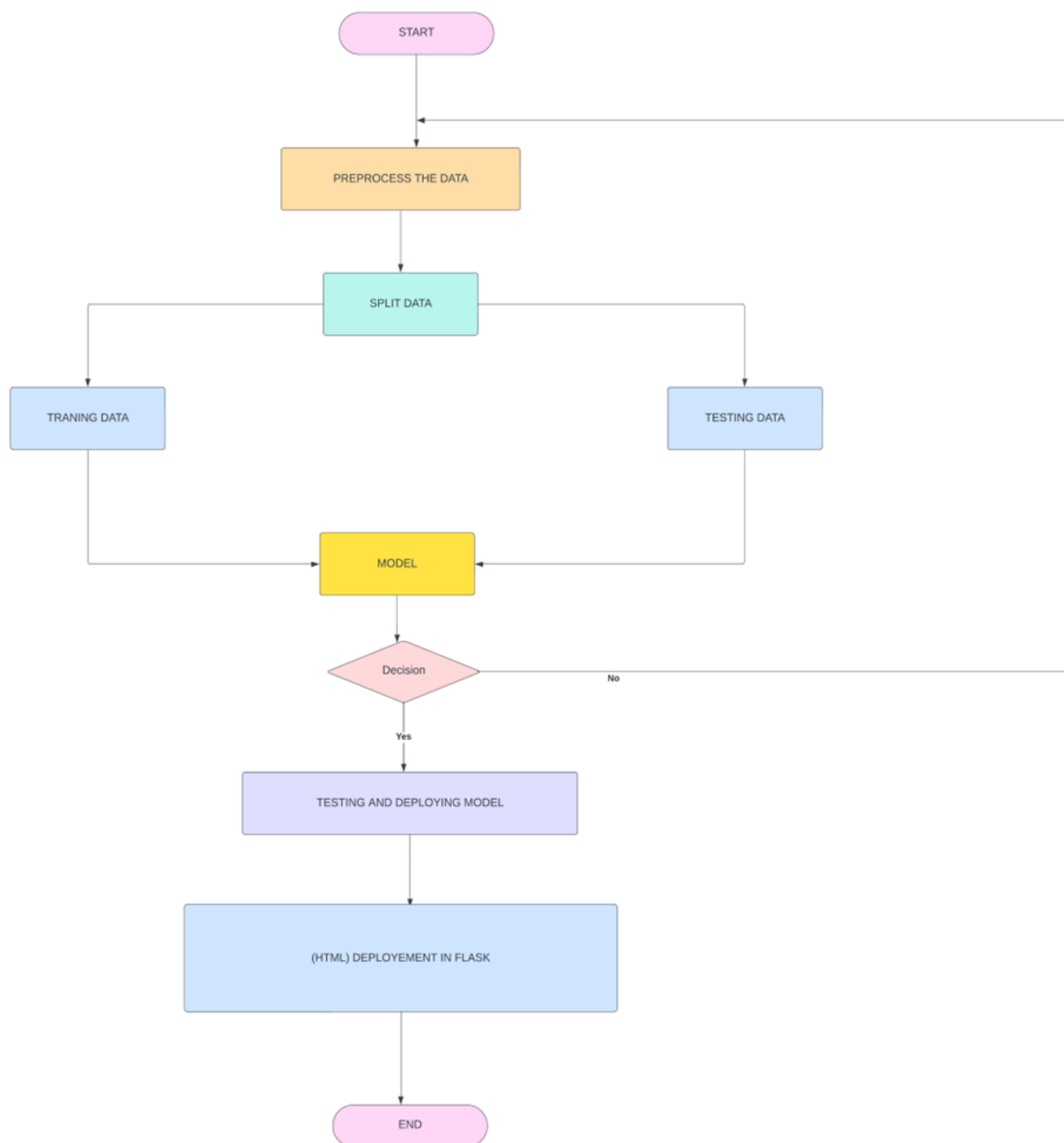


Fig-13: showing the control flow of the solution

ADVANTAGES

- Improved Accuracy: ML algorithms can analyze large volumes of data and identify complex patterns, leading to more accurate predictions of medical costs.
- Cost Reduction: By accurately predicting medical costs, ML can help healthcare organizations optimize resource allocation and reduce unnecessary expenditures.
- Early Intervention: ML models can identify high-risk patients who are likely to generate high medical costs, allowing for early intervention and proactive care management.
- Individualized Predictions: ML algorithms can generate personalized cost predictions for individual patients based on their unique characteristics and medical history.
- Timely Decision-Making: ML models provide real-time predictions, enabling healthcare providers to make informed decisions promptly and allocate resources effectively.
- Enhanced Fraud Detection: ML algorithms can detect anomalies and patterns indicative of fraudulent activities, helping to reduce healthcare fraud and abuse.
- Risk Stratification: ML can stratify patients into different risk groups based on their predicted costs, facilitating targeted interventions and customized care plans.
- Improved Resource Planning: ML predictions can assist in predicting future resource needs, such as hospital beds, medical equipment, and staff, for better resource planning.
- Efficiency and Productivity: ML algorithms automate the prediction process, saving time and effort for healthcare professionals, allowing them to focus on patient care.
- Scalability: ML models can be scaled up to analyze large datasets from multiple healthcare institutions, enabling broader insights and more robust predictions.

LIMITATIONS

- Data Availability and Quality: Availability of high-quality and comprehensive medical data can be a challenge, limiting the accuracy and generalizability of ML predictions.
- Ethical Concerns: ML predictions may raise ethical issues, such as privacy violations and bias in decision-making, requiring careful handling of sensitive patient information.

- Interpretability: ML models often lack transparency, making it difficult to interpret the factors contributing to cost predictions and limiting their adoption in critical decision-making scenarios.
- Generalizability: ML models trained on specific populations or healthcare settings may not generalize well to other populations or regions, limiting their applicability.
- Complex Implementation: Integrating ML algorithms into existing healthcare systems and workflows can be complex and time-consuming, requiring technical expertise and infrastructure.
- Changing Healthcare Dynamics: ML models may struggle to adapt to rapidly evolving healthcare dynamics, such as changes in medical practices, reimbursement policies, or disease prevalence.
- Limited Contextual Factors: ML models primarily focus on cost predictions and may overlook contextual factors, such as socioeconomic status or environmental factors, which can influence medical costs.
- Lack of Human Judgment: Relying solely on ML predictions may overlook the valuable insights and clinical judgment that healthcare professionals bring to the decision-making process.
- Cost and Resource Constraints: Implementing ML solutions may require significant investments in infrastructure, training, and maintenance, which may not be feasible for all healthcare organizations.
- Uncertainty and Variability: ML predictions may have inherent uncertainty and variability, making it challenging to accurately predict individual medical costs with complete certainty.

APPLICATIONS

- **Health Insurance Pricing:** Medical cost prediction systems can assist health insurance companies in accurately pricing their policies based on the predicted costs of potential policyholders. This helps ensure fair pricing and profitability for insurance providers.
- **Healthcare Budgeting and Resource Allocation:** Predicting medical costs can aid healthcare organizations in budgeting and allocating resources effectively. It allows them to anticipate future expenses, plan investments in medical equipment and facilities, and optimize staffing levels.
- **Disease Management and Prevention:** Medical cost prediction systems can identify individuals at high risk of developing specific medical conditions or requiring costly treatments. This enables healthcare providers to implement targeted disease management programs and preventive measures to mitigate risks and reduce costs.
- **Provider Network Optimization:** Healthcare payers can use cost prediction systems to evaluate and optimize their provider networks. By identifying providers with better cost-efficiency and outcomes, insurers can direct patients to those providers, resulting in improved quality of care and cost savings.
- **Fraud Detection and Prevention:** Medical cost prediction systems can help identify patterns and anomalies indicative of fraudulent activities in healthcare claims. By detecting fraudulent claims early, insurers can prevent financial losses and protect the integrity of the healthcare system.
- **Cost Transparency and Patient Education:** Predicting medical costs can enhance cost transparency for patients, allowing them to make informed decisions about their healthcare choices. Patients can estimate potential out-of-pocket expenses, compare treatment options, and better plan for their healthcare expenses.
- **Population Health Management:** Medical cost prediction systems play a crucial role in population health management initiatives. By identifying subpopulations at high risk of incurring significant healthcare costs, healthcare organizations can develop targeted interventions, preventive programs, and policy interventions to improve population health outcomes while managing costs efficiently.

CONCLUSION

In this project, we developed a medical cost prediction system using Regression Trees, Random Forest Regression, Gradient Boosted Regression Trees, and Linear Regression algorithms. The goal was to predict medical prices based on a given dataset. We compared the performance of these models and analyzed their results.

From our experiments, we found that the performance of the different models varied depending on the configuration used during testing. Therefore, it is challenging to determine a single best-performing model overall. However, based on the test configuration parameters, the order of performance, from best to worst, was determined to be Gradient Boosted Regression Trees, Random Forest Regression, Regression Trees, and Linear Regression.

The predicted average medical payments obtained from Gradient Boosted Regression Trees, Random Forest Regression, and Regression Trees closely approximated the actual values of payments. This suggests that these models have the potential to provide accurate predictions for medical costs.

FUTURE SCOPE

While we have achieved promising results in this project, there are several areas for future exploration and improvement in the medical cost prediction system:

- **Feature Engineering:** Investigate additional features that could potentially improve the prediction accuracy, such as patients' demographics, lifestyle factors, and medical history. Feature selection techniques can also be employed to identify the most influential variables.
- **Model Optimization:** Explore hyperparameter tuning techniques to further enhance the performance of the models. This could involve conducting grid searches or using more advanced optimization algorithms to find the optimal combination of hyperparameters.

- Ensemble Methods: Investigate the use of ensemble methods, such as stacking or bagging, to combine the predictions of multiple models and potentially improve overall performance.
- Data Expansion: Acquire a larger and more diverse dataset to train the models. This would enable better generalization and capture a broader range of medical scenarios.
- Real-Time Implementation: Develop a user-friendly interface for the medical cost prediction system, allowing healthcare professionals and insurance providers to input patient information and obtain cost estimates in real-time. This could involve building a web or mobile application.
- Ethical Considerations: Address the ethical implications of using machine learning algorithms in the medical domain, ensuring fairness, transparency, and privacy. Implement measures to prevent bias and protect patient data.

By addressing these future directions, we can further enhance the accuracy and usefulness of the medical cost prediction system, benefiting both healthcare providers and patients in making informed decisions regarding medical expenses.

Code:Flask Code:

```

from flask import Flask, render_template, request

app = Flask(__name__)# interface between my server and my application wsgi

import pickle
model = pickle.load(open(r'./rf_tuned.pkl','rb'))

@app.route('/')#binds to an url
def helloworld():
    return render_template("index.html")

@app.route('/login', methods =['POST'])#binds to an url
def login():
    p=request.form["Age"]
    q=request.form["Sex"]
    r=request.form["Bmi"]
    s=request.form["C"]
    t=request.form["Sm"]
    u=request.form["R"]

    t=[[int(p),int(q),float(r),int(s),int(t),int(u)]]
    output= model.predict(t)
    print(output)

```

```
return render_template("index.html",y = "=" + str(output[0]) )
```

```
if __name__ == '__main__':
```

```
    app.run(debug= False)
```

HTML Code:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>Medical Cost Prediction</title>
```

```
    <link rel="preconnect" href="https://fonts.googleapis.com">
```

```
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```
    <link href="https://fonts.googleapis.com/css2?family=Libre+Baskerville&display=swap"
rel="stylesheet">
```

```
    <link rel="stylesheet" href="../static/style.css"/>
```

```
</head>
```

```
<body>
```

```
    <div class="all">
```

```
        <h1 class="heading">Medical Cost Prediction</h1>
```

```
        </img>
```

```
        <form action = "/login" method= "post">
```

```
            <div class="var">
```

```
                <p class="age">Age</p>
```

```
                <p><input type="text" name = "Age" /></p>
```

```
                <p class="sex">Sex (male-0 / female-1)</p>
```

```
                <p><input type="text" name = "Sex" /></p>
```

```
                <p class="bmi">Bmi Index</p>
```

```
                <p><input type="text" name = "Bmi" /></p>
```

```
                <p class="child">Number of Children</p>
```

```
                <p><input type="text" name = "C" /></p>
```

```
                <p class="smoke">Smoke (Yes-1 / No-0)</p>
```

```
                <p><input type="text" name = "Sm" /></p>
```

```
                <p class="region">Region(0-NW, 1-NE, 2-SE, 3-SW)</p>
```

```
                <p><input type="text" name = "R" /></p>
```

```
            </div>
```

```

        <p><input type="submit" value = "submit" /></p>
    </div>
</form>
<div class="prediction">
    <b>Predicted Cost {{y}}</b>
</div>
</body>
</html>

```

CSS Code:

```

body{
    background-color: bisque;
}

.all{
    font-family: 'Libre Baskerville', serif;
    display: grid;
    grid-column:auto;
    width: 75%;
    height: 75%;
    gap: 10px;
    background-color: paleturquoise;
    justify-content: center;
    margin-left: 12.5%;
    margin-top: 3%;
    border-radius: 3%;
}

.prediction{
    display: flex;
    width: auto;
    height: auto;
    background-color: palegreen;
    justify-content: space-around;
    margin-top: 10px;
    font-family: 'Libre Baskerville', serif;
}

.heading{
    font-family: 'Libre Baskerville', serif;
    font-weight: 900;
}

```

```
margin: 0;  
margin-top: 10px;  
}
```

```
.var{  
display: grid;  
grid-template-columns: 50% 50%;
```