APPLIED DATA SCIENCE - SMARTINTERNZ

MEDICAL INSURANCE COST PREDICTION

Submitted by

N.Dhyana Sai-20BCE2956

Dasari Bhanu Harshitha-20BCT0135

Kolla Kranthi Priya-20BDS0291

K.V.Saketh-20BCI0255

1 INTRODUCTION

1.1 Overview

We live in a dangerous and unpredictable world. Different risk types are exposed to by people, homes, businesses, properties, and property. The degree of risk can also change. These risks include the potential for demise, illness, and the loss of assets or property. The greatest aspects of a person's life are life and wellbeing. But since risks can't always be avoided, the financial industry has created a variety of products to protect people and businesses against them by utilising money to cover the costs. So insurance is a strategy that lowers or eliminates the costs associated with losses brought on by various risks. It becomes crucial for insurance companies to be sufficiently accurate in measuring or quantifying the amount covered by this policy and the insurance fees that must be paid for it when considering the value of insurance in people's lives. These costs are estimated using a variety of variables. Each of these is significant. If any factor is left out when the quantities are calculated, the policy as a whole changes. As a result, it is vital that these activities be completed precisely. Because human error is possible, insurers employ experts in this field. They also employ various tools to compute the insurance premium. Data science is useful in this situation. Data Science may generalise the effort or method used to develop policy. These ml models can be taught independently. The model is trained using historical insurance data. The necessary criteria for measuring payments can then be defined as model inputs, and the model can accurately predict insurance policy costs. This reduces human work and resources while increasing the company's profitability. As a result, ml can enhance accuracy.The Insurance dataset is trained with a Random Forest Regression model, and a web application is built with Flask integration that accepts user input and forecasts insurance costs.

1.2 Purpose

- The main goal of this project is to predict insurance costs using Random Forest Regression, a machine learning model.A web application is constructed in which a user inputs data such as age, gender, BMI, smoker, children, and region, and the web page displays the predicted amount.The essential payment criteria can then be defined as model inputs, and the model can reliably anticipate insurance policy costs. This minimises human labour and resources while enhancing the company's profitability. As a result, this model may improve accuracy.

- The insurance cost is 2641 which is very close to the real value. This particular data points the medical insurance cost is 2640 dollars and the value predicted by the model is 2641 which is very very close.

- So it tells us the model is performing kind of very well.

2 LITERATURE SURVEY

2.1 Existing problem

Authors and Year (Reference)	***Title (Study)***	***Concept / Theoretical model/*** ***Framework***	***Iethodology ised/ mplementati on***	***Dataset details/ Analysis***	***Relevant Finding***
MOHAMMED HANFEY, Omar M. A. Mahmoud (2021) [1]	Predict Health Insurance Cost by using Machine Learning and DNN Regression Models	The research ises various nachine learning egression models ind deep neural ietworks to orecast charges of iealth insurance ased on specific ittributes, on nedical cost ersonal dataset rom Kaggle	shows that Stochastic Gradient Boosting offers the best efficiency, with an RMSE value of 0.380189, an MAE value of 0.17448, and an accuracy of 85.8	the data set is separated into two-part the first part called training data, and the second called test data; training data makes up about 80 percent of the total data used, and the rest for test data The training data set is applied to build a model as a predictor of medical insurance cost year and the test set will use	Machine learning (ML) for the insurance industry sector can make the wording of insurance policies more efficient. This study demonstrates how different models of regression can forecast insurance costs

				to evaluate the regression model	
Dr. Akhilesh Das Gupta (2020) [2]	Health Insurance Amount Prediction	The goal of this project is to allows a person to get an idea about the necessary amount required according to their own health status	Three regression models naming Multiple Linear Regression, Decision tree Regression and Gradient Boosting Decision Tree Regression have been used to compare and contrast the performance of these algorithms	The primary source of data for this project was from Kaggle user D'marco.The dataset consists of 1338 records with 6 attributes. Attributes are as follows: ‘age’,‘gender’,‘bmi’,‘children’, ‘smoker’ and ‘charges’.	We see that the accuracy of predicted amount was seen best i.e., 99.5% in gradient boosting decision tree regression. Other two regression models also gave good accuracies about 80%.
Saddam Hussain, Mogeeb A. A. Mosleh [3]	A Computational Intelligence Approach for Predicting Medical Insurance Cost	In this study, we used supervised ML models to demonstrate and compare the accuracy of various regression models, including Linear Regression (LR), Stochastic Gradient Boosting (SGB), XGBoost (XGB)	The proposed research approach uses Linear Regression, Support Vector Regression, Ridge Regressor, Stochastic Gradient Boosting, XGBoost, Decision Tree, Random	The medical cost personal datasets are obtained from the KAGGLE repository. This dataset contains seven attributes	Data mining (DM) and machine learning (ML) techniques are widely used for insurance cost prediction and medical fraud detection. Using the Extreme Gradient Boosting algorithm, we improved the accuracy of a decision tree

			Forest Regressor, Multiple Linear Regression, and k-Nearest Neighbours		classifier for predicting healthcare insurance fraud.
Nataliya Shakhovska1 , Valentyna Chopiyak2 and Michal Gregus ml3 [4]	An Ensemble Methods for Medical Insurance Costs Prediction Task	The paper reports three new ensembles of supervised learning predictors for managing medical insurance costs. The open dataset is used for data analysis methods development.	bagging shows its weakness in generalising the prediction. The stacking is developed using K Nearest Neighbors (KNN), Support Vector Machine (SVM), Regression Tree, Linear Regression, Stochastic Gradient Boosting	The medical insurance payments dataset [29] was selected. It consists of 7 attributes and 1338 vectors. The task is to predict individual payments for health insurance.	two feature selection techniques for the comparison of the prediction accuracy of the different machine learning algorithms were applied. The weak components for the design an ensemble models were found

2.2 Proposed solution

- RandomForestRegression is the best model, with the highest r2__score.The higher the r2_score, the better the model.By integrating numerous decision trees, you can create a dependable and exact predictive model using the Random Forest Regressor, an ensemble learning technique.The insurance data is first obtained, then preprocessed using several ways before being trained using Random Forest Regression.This model is then used to forecast the cost of insurance.
- It offers good accuracy and is simple to use and interpret.

3 THEORETICAL ANALYSIS

The theoretical examination of the solution entails comprehending the fundamental concepts, approaches, and considerations associated with developing a predictive model for insurance cost and integrating it into a Flask web application. Here are some crucial points to consider:

a. Data Preprocessing: Before creating the model, data preprocessing techniques are used to prepare the dataset. This comprises converting categorical variables to numerical form using one-hot encoding, partitioning the dataset into independent variables (X) and dependent variables (y), and splitting the data into training and test sets for evaluation.

b. Feature Engineering: Feature engineering entails changing input features to improve the model's predictive capacity. Polynomial features are applied to the given solution using scikit-learn's PolynomialFeatures class. This enables the capture of non-linear interactions between the input variables and the target variable.

c. Feature Scaling: Feature scaling is used to normalise the input features and guarantee they are all on the same scale. The scikit-learn StandardScaler is used to standardise the features, which improves the model's performance and convergence.

d. Model Evaluation: The R-squared statistic is used to assess the predictive model's performance. R-squared calculates the proportion of the variance in the dependent variable that can be explained by the independent variables. A higher R-squared value implies a better fit of the model to the data.

e.Web Application Integration: The Flask web framework is utilised to develop the web application. Flask lets you create routes and views to manage user requests and render HTML templates. The programme collects user data via a form, sends it to the predictive model, and presents the estimated insurance cost on a result page.

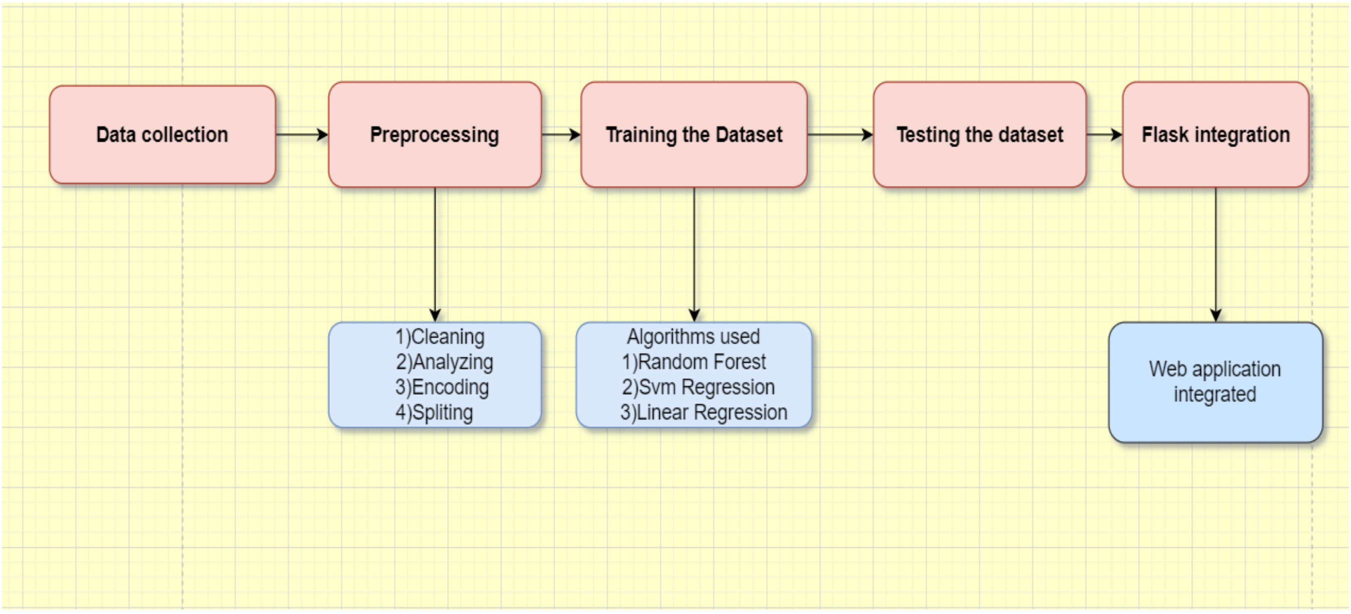
f. User Interface (UI) Design: The user interface is a critical component of any web application. The HTML templates are intended to provide a user-friendly interface for entering the relevant information and showing the prediction results. CSS styling is used to improve the visual appearance and layout of online pages.

g. Deployment and Accessibility: The Flask application is deployed on a local server and is accessible via a web browser. The application runs on the Flask development server. Users can enter the relevant information, submit the form, and check the estimated insurance cost on the result page.

h. Applications and Implications: Theoretical analysis investigates the solution's potential applications and implications in a variety of fields, including the insurance sector, healthcare planning, personal finance, insurance product creation, risk management, and research. It demonstrates how the predictive model and web

application may be used to improve decision-making, financial planning, and resource allocation in these domains.
By undertaking a theoretical analysis, we acquire a better grasp of the underlying concepts and processes used in the solution. It assists us in comprehending the prediction models and web application's capabilities, limits, and prospective use cases.

3.1 Block diagram



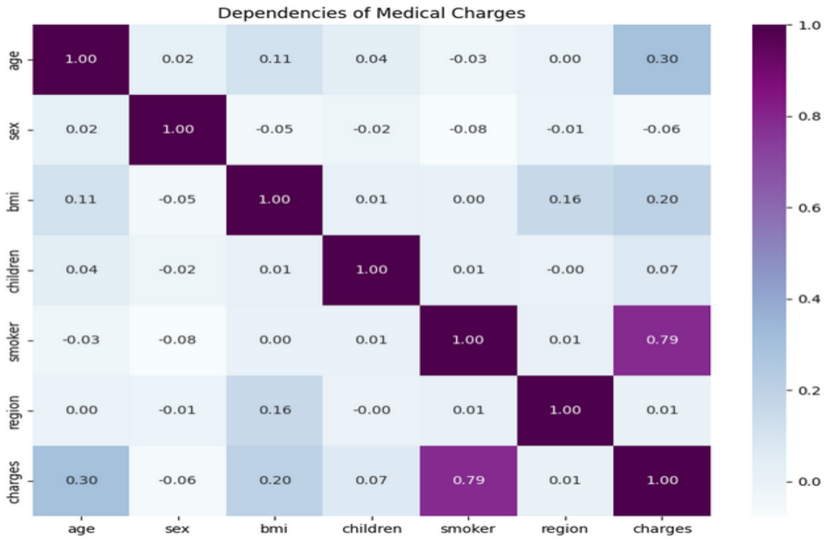
3.2 Hardware / Software designing

Hardware Components:desktop/laptop

Software Components:VScode,Jupyter Notebook,python packages and Flask micro web framework to develop web application.

4 .EXPERIMENTAL INVESTIGATIONS

Correlation:



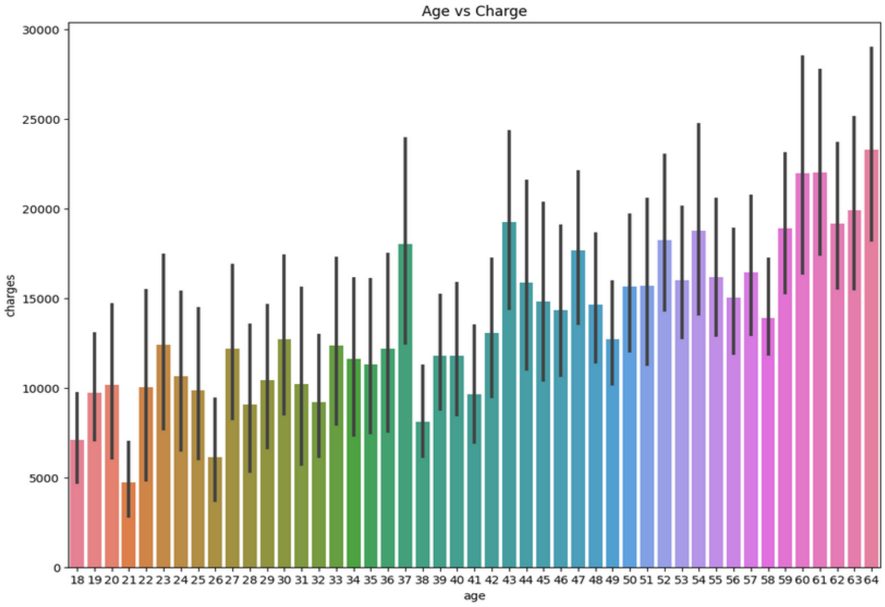
	age	sex	bmi	children	smoker	region	charges
age	1.00	0.02	0.11	0.04	-0.03	0.00	0.30
sex	0.02	1.00	-0.05	-0.02	-0.08	-0.01	-0.06
bmi	0.11	-0.05	1.00	0.01	0.00	0.16	0.20
children	0.04	-0.02	0.01	1.00	0.01	-0.00	0.07
smoker	-0.03	-0.08	0.00	0.01	1.00	0.01	0.79
region	0.00	-0.01	0.16	-0.00	0.01	1.00	0.01
charges	0.30	-0.06	0.20	0.07	0.79	0.01	1.00

most important features are Smoker,BMI and Age that determnines - Charges

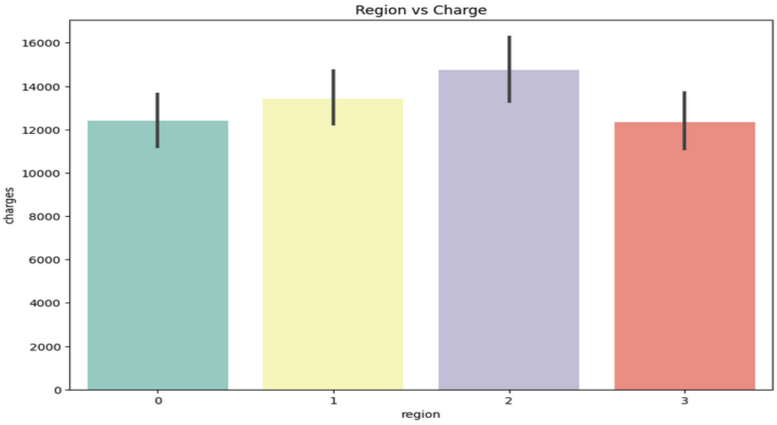
Sex, Children and Region do not affect the Charges. We might drop these 3 columns as they have less correlation

Visualisation:

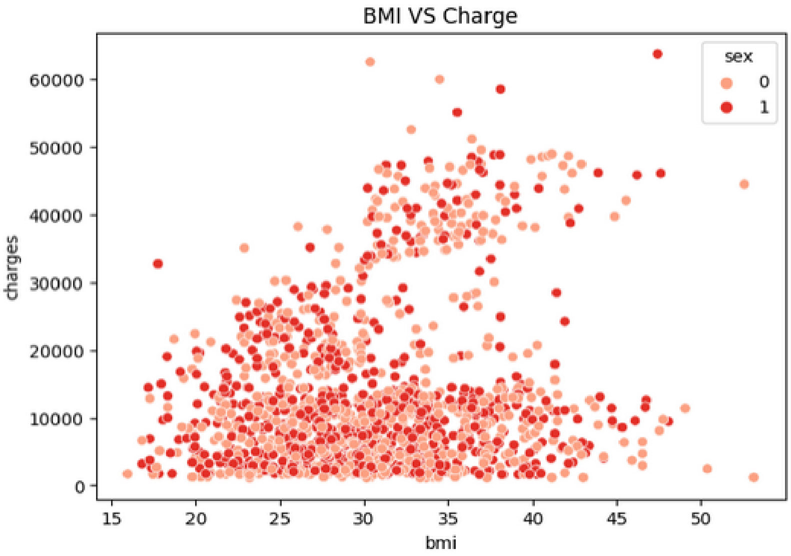
Age Vs Charges



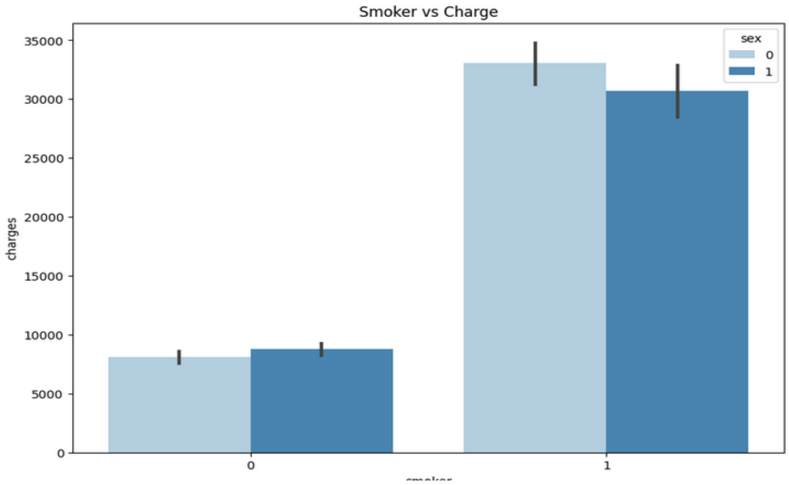
Region vs Charges



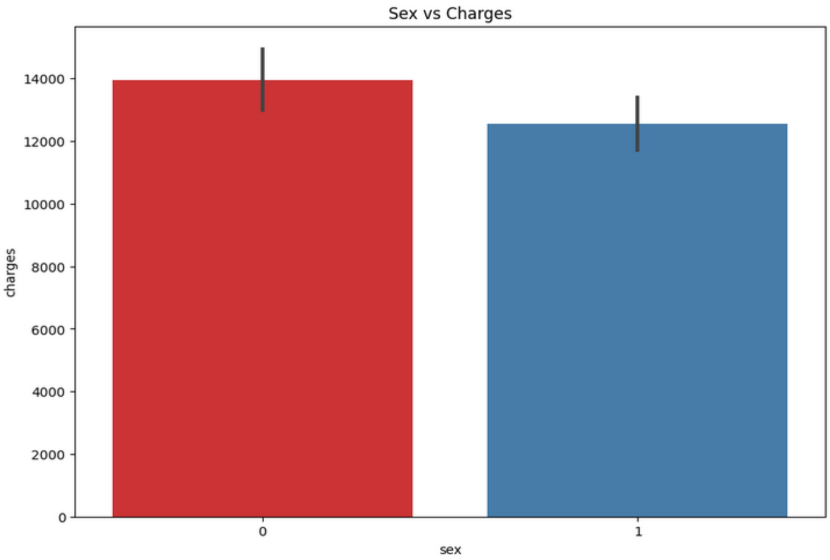
BMI vs Charges



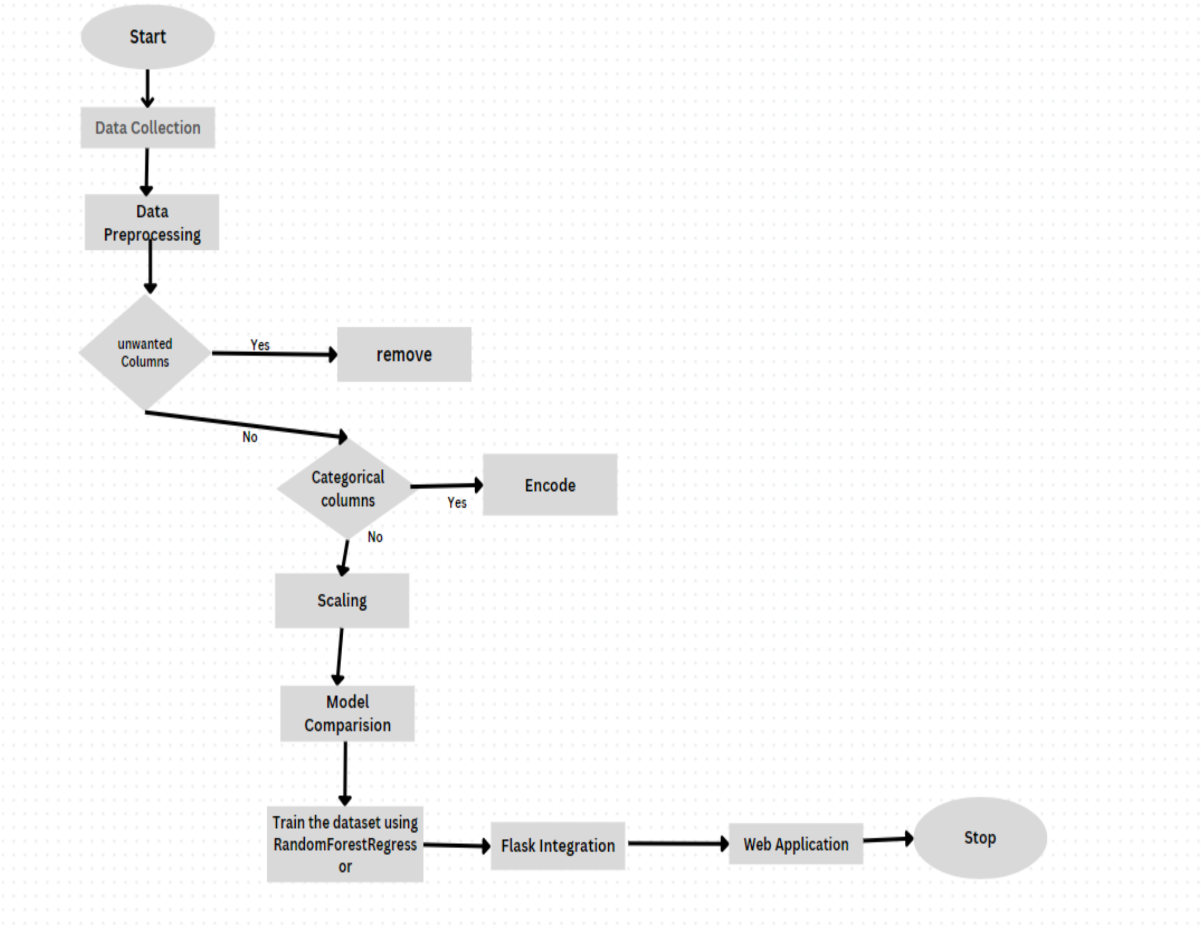
Smoker Vs Charges



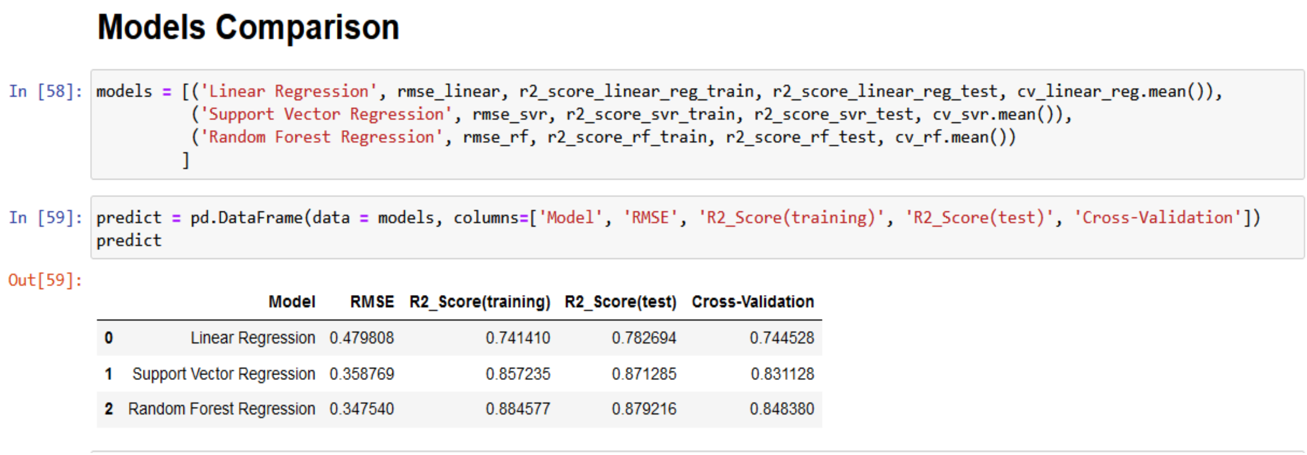
Sex Vs Charges



5 FLOWCHART



6 RESULT



Models Comparison

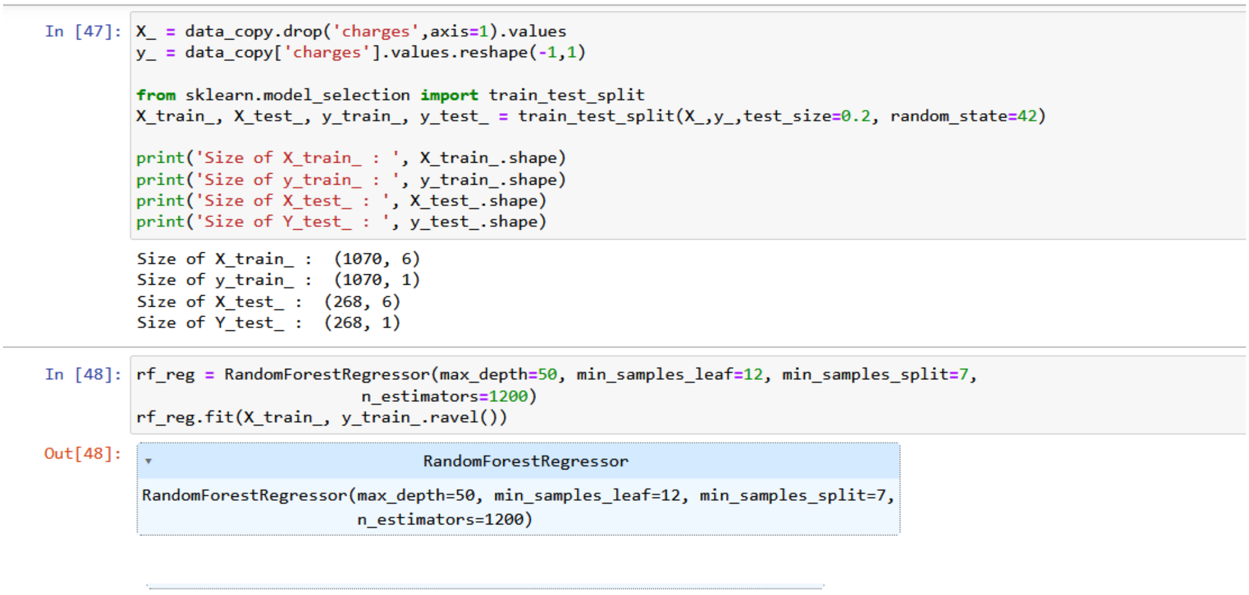
```
In [58]: models = [('Linear Regression', rmse_linear, r2_score_linear_reg_train, r2_score_linear_reg_test, cv_linear_reg.mean()),
                    ('Support Vector Regression', rmse_svr, r2_score_svr_train, r2_score_svr_test, cv_svr.mean()),
                    ('Random Forest Regression', rmse_rf, r2_score_rf_train, r2_score_rf_test, cv_rf.mean())
                   ]
```

```
In [59]: predict = pd.DataFrame(data = models, columns=['Model', 'RMSE', 'R2_Score(training)', 'R2_Score(test)', 'Cross-Validation'])
         predict
```

Out[59]:

	Model	RMSE	R2_Score(training)	R2_Score(test)	Cross-Validation
0	Linear Regression	0.479808	0.741410	0.782694	0.744528
1	Support Vector Regression	0.358769	0.857235	0.871285	0.831128
2	Random Forest Regression	0.347540	0.884577	0.879216	0.848380

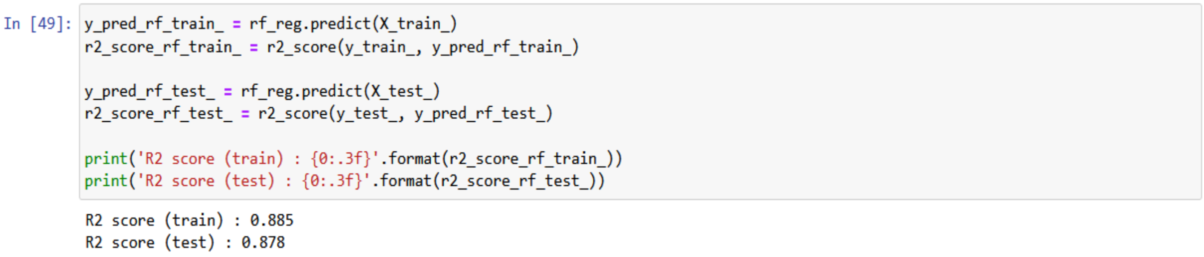
RandomForestRegressor:



```
In [47]: X_ = data_copy.drop('charges',axis=1).values
         y_ = data_copy['charges'].values.reshape(-1,1)

         from sklearn.model_selection import train_test_split
         X_train_, X_test_, y_train_, y_test_ = train_test_split(X_,y_,test_size=0.2, random_state=42)

         print('Size of X_train_ : ', X_train_.shape)
         print('Size of y_train_ : ', y_train_.shape)
         print('Size of X_test_ : ', X_test_.shape)
         print('Size of Y_test_ : ', y_test_.shape)

         Size of X_train_ :  (1070, 6)
         Size of y_train_ :  (1070, 1)
         Size of X_test_ :  (268, 6)
         Size of Y_test_ :  (268, 1)
```

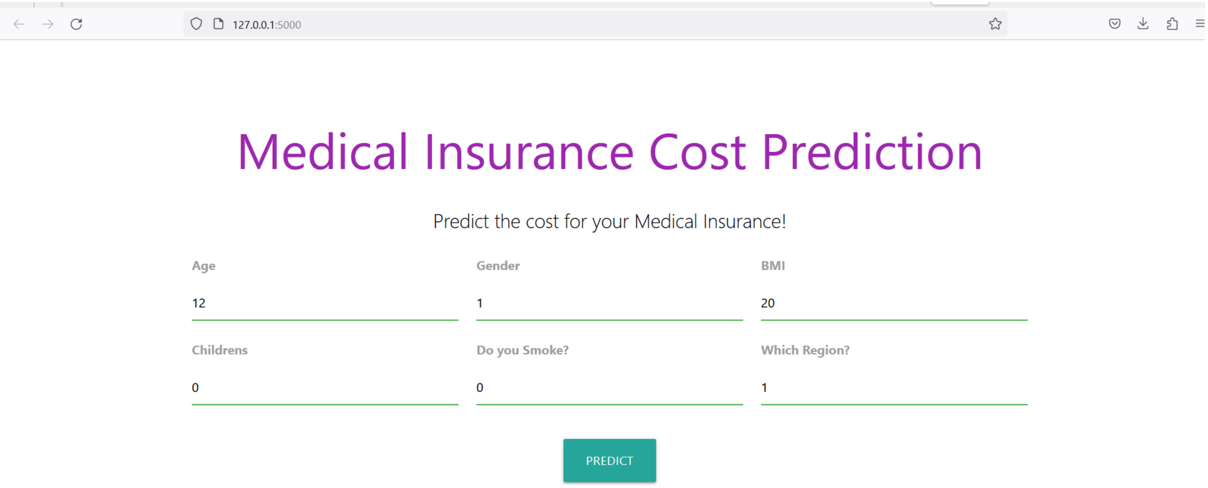
```
In [48]: rf_reg = RandomForestRegressor(max_depth=50, min_samples_leaf=12, min_samples_split=7,
                                        n_estimators=1200)
         rf_reg.fit(X_train_, y_train_.ravel())
```

```
Out[48]:                      RandomForestRegressor
         RandomForestRegressor(max_depth=50, min_samples_leaf=12, min_samples_split=7,
                               n_estimators=1200)
```



```
In [49]: 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) : 0.885
         R2 score (test) : 0.878
```

WebApplication:



Medical Insurance Cost Prediction

Predict the cost for your Medical Insurance!

Age: 12

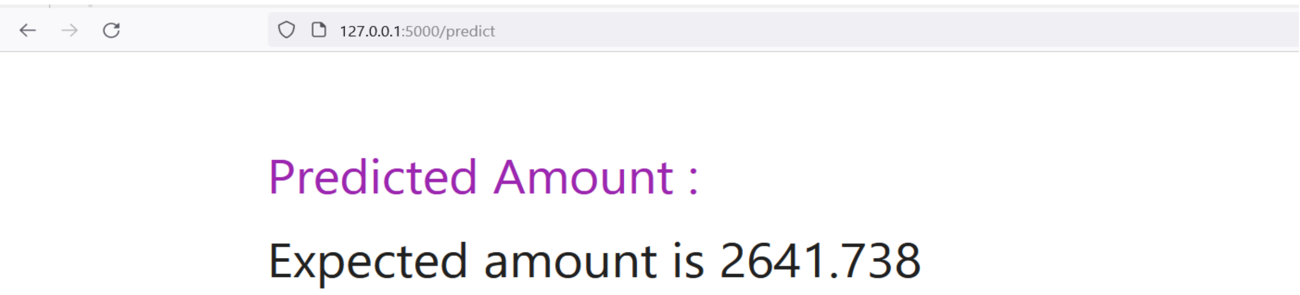
Gender: 1

BMI: 20

Childrens: 0

Do you Smoke?: 0

Which Region?: 1



Predicted Amount :

Expected amount is 2641.738

OBSERVATION:

Model	RMSE	R2_Score(training)	R2_Score(test)	Cross-Validation
Linear Regression	0.479808	0.741410	0.782694	0.744528
Support Vector Regression	0.358769	0.857235	0.871285	0.831128
Random Forest Regression	0.347540	0.884577	0.879216	0.848380

7 ADVANTAGES & DISADVANTAGES

Advantages:

The machine learning algorithm known as Random Forest Regression, which is based on the Random Forest ensemble method, offers notable advantages in a range of applications. Following are some advantages of employing Random Forest Regression:

a. Precision: Random Forest Regression is known for making precise predictions. By combining several decision trees, each trained on a distinct sample of the data, the method reduces the chance of overfitting and produces accurate predictions. It can handle both categorical and continuous data, making it flexible for many types of regression situations.

b. Deals with Non-Linearity:A non-linear relationship between the input features and the target variable can be found using Random Forest Regression. The data may contain complex correlations and nonlinear patterns, which conventional linear regression methods can find difficult to reproduce.

Random Forest Regression, which is generally robust to outliers and missing data, is largely unaffected by them. The averaging process and the use of several decision trees help to reduce the impact of outliers and handle missing values by making predictions based on the data contained in each tree.

c.Random Forest Regression: Random Forest Regression, which identifies the variables that have the largest impact on the predictions, provides a gauge of feature importance. By facilitating a better understanding and interpretation of the underlying relationships in the data, this feature aids in the selection of characteristics and the identification of the main drivers of the target variable.

d. Overfitting-resistant: The ensemble structure and randomization of the Random Forest Regression assist guard against overfitting. By creating each decision tree using a random subset of traits and samples, the method reduces the risk of memorising noise in the training data, resulting in a more generalizable model.

e. Scalability and Versatility: Random Forest Regression can work with large datasets that have a variety of properties. It is suitable for big data applications since it can be parallelized and uses computer resources efficiently. Additionally, it enables problems including both classification and regression, enabling adaptable problem-solving.

Utilising the out-of-bag (OOB) error estimation method is Random Forest Regression. It makes use of the samples omitted during training to provide performance estimates for the model without the need for cross-validation or a different validation set. This feature simplifies the model evaluation process and conserves processing resources.

f. Less Prone to Bias: Random Forest Regression is less biassed than single decision tree models. By averaging the predictions of several trees, it lessens the impact of

individual decision trees' biases and produces a more reliable and accurate overall prediction.
All things considered, Random Forest Regression is a robust and flexible algorithm that performs well when managing non-linear connections, accepting various input types, and generating accurate predictions.
Due to its resilience, feature importance analysis, and scalability, it is a preferred choice for many regression issues across various disciplines.

DISADVANTAGES:

Despite the fact that random forest regression models have numerous advantages, they also have certain disadvantages. Employing a random forest regression model has the following drawbacks:
a.Inability to be interpreted: Random forests are referred to as "black box" models, which make it challenging to comprehend the underlying decision-making process.
a.Inability to be comprehended. Although they provide accurate forecasts, it may be difficult to understand the precise variables and connections that affect the projections.
b.Computer complexity: Random forests can be difficult to compute, particularly when dealing with large datasets or a densely packed forest. A random forest regression model may need a lot of time and processing power to construct and train.
c.Potential for overfitting: Although random forests are designed to prevent overfitting in comparison to individual decision trees, there is still a chance of overfitting, particularly if the model is not well calibrated. If the number of trees in the forest or the depth at which it was trained is too vast, a model may start capturing noise and oddities in the training data, which will have poor generalisation to new data.
d.Random forests can be affected by data noise, outliers, and irrelevant attributes. Noisy or unimportant variables may have an impact on specific trees within a forest's structure, leading to biassed forecasts.
e.Memory requirements: Random forests retain the split points and thresholds of each tree in the forest. The quantity of training data and the number of trees both cause an increase in memory requirements. This might be a problem in circumstances where memory is scarce.
f.Unbalanced datasets are difficult to handle since random forests typically predict the class that is in the majority. If the distribution of the class is skewed, this could lead to biassed predictions and poor performance.
g. Limited extrapolation capability: Random forests often aren't appropriate for extrapolation tasks, which involve making predictions outside the parameters of the training data. Beyond the reported range, the model usually relies on interpolating between known data points rather than offering precise forecasts.

It's crucial to remember that despite these issues, random forests are still widely used and quite effective in many applications. These limitations may or may not be a huge concern depending on your specific requirements.

8 APPLICATIONS

The solution, which entails utilising Random Forest to create a prediction model for insurance pricing and incorporating it into a Flask web application, can be used in a variety of contexts. Several potential uses include:

a. Insurance Industry: Insurance providers can utilise the solution to give prospective clients fast estimations of the cost of insurance. By giving quick pricing information based on input criteria like age, gender, BMI, number of children, smoker status, and geography, it can assist speed the quoting process and enhance the client experience.

b. Healthcare Planning: The model can be used for budgeting and planning in the healthcare industry. Policymakers and healthcare professionals can assess the financial impact of various population segments and make educated decisions about resource allocation, insurance coverage, and healthcare planning by projecting insurance costs based on demographic and lifestyle parameters.

c. Personal Finance and Budgeting: Users of the web programme can use it to calculate their insurance expenses and factor them into their personal budgeting. It can help people set aside money for insurance costs and make wise insurance coverage decisions based on their unique needs.

d. Insurance Product Development: Insurance companies can use the predictive model to assess how various factors affect insurance costs and create new insurance products that are specialised for particular consumer groups. The model can help with risk analysis, pricing strategies, and underwriting standards for new insurance products.

e. Risk Management: By offering insights into the elements that contribute to greater insurance prices, the solution can be used in risk management procedures. Based on their demographic and lifestyle traits, it can assist in identifying high-risk people or groups, allowing for tailored risk mitigation methods and specialised insurance options.

f. Research and Analysis: The model and web application can be used as a tool by researchers and analysts who are looking into how different factors affect insurance costs. In order to help scholarly study, market analysis, and policy evaluations, it can be used to analyse patterns, trends, and correlations in insurance data.

These are just a few examples of how this solution may be used. The predictive model's adaptability and integration with a web application open up possibilities for use in a variety of fields where insurance cost prediction, financial planning, risk assessment, and decision-making are involved.

9 CONCLUSION

A model that forecasts insurance costs is described in detail by the proposed system.The predicted quantity is displayed once a user enters details regarding variables like age, gender, BMI, children, and region.The model is trained using a random forest regressor after the collection of insurance data and several preprocessing methods.The trained model is compared to linear regression and SVM. The results of the r_score were most accurately produced by the RandomForestRegressor.The flask integration of the model results in the creation of a web application.With the use of a web application, predicting the cost of medical insurance was shown to be a practical and cheap option. Consequently, accuracy is increased.As a result, the company's profitability increases while using fewer resources and people.

10 FUTURE SCOPE

The user interface (UI) of the web application can be modified in the future to improve user engagement and experience. Think about incorporating user-friendly forms, interactive visualisations, and extra features that offer individualised advice, price comparisons, or insightful information about insurance prices.In the future, it will be possible to create and use apps.

11 BIBLIOGRAPHY

[1] Hanafy, M., & Mahmoud, O. M. A. (2021). Predict Health Insurance Cost by Using Machine Learning and DNN Regression Models. *Int. J. Innov. Technol. Explor. Eng*, *10*, 137-143.

[2] Bhardwaj, N., Anand, R., & Gupta, A. D. (2020). Health Insurance Amount Prediction International Journal of Engineering Research & Technology. *(IJERT)*, *9*(05).

[3] ul Hassan, C. A., Iqbal, J., Hussain, S., AlSalman, H., Mosleh, M. A., & Sajid Ullah, S. (2021). A computational intelligence approach for predicting medical insurance cost. *Mathematical Problems in Engineering*, *2021*, 1-13.

[4] Shakhovska, N., Melnykova, N., & Chopiyak, V. (2022). An Ensemble Methods for Medical Insurance Costs Prediction Task. *Computers, Materials & Continua*, *70*(2).

APPENDIX

A. Source Code

https://github.com/BhanuHarshitha15/Medical-Insurance-Cost-Prediction