## **Artificial Intelligence and Machine Learning**

Project Report

Semester-IV (Batch-2022)

**Case Study**: -**Insurance Cost Prediction**

[Url:-](about:blank)

[https://drive.google.com/file/d/1ZQSDNdJsthrwa7UtfLegyG\_4IR3ztDrd/view?usp=drive\_link](https://drive.google.com/file/d/1ZQSDNdJsthrwa7UtfLegyG_4IR3ztDrd/view?usp=drive_link%09)

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Rajeev Thakur Pulkit vashisht

Roll Number: -2210992594

Group - 14

**Department of Computer Science and Engineering**

**Chitkara University Institute of Engineering & Technology,**

## **Chitkara University, Punjab**

**Case Study: Insurance Cost Prediction**

* **Objective:**

The objective of this analysis is to predict insurance costs for individuals based on various factors such as age, gender, BMI, number of children, smoking habits, and region.

* **Library Used:**

Pandas library

NumPy library

Matplotlib library

Scikit-learn library

Joblib library

Tkinter library

* **Methods:**

read\_csv(): Reads a CSV file and converts it into a data frame.

head(): Displays the first few rows of the data frame.

tail(): Displays the last few rows of the data frame.

shape(): Returns the shape (number of rows, number of columns) of the data frame.

info(): Provides basic information about the data frame, such as column types and missing values.

isnull(): Returns True/False for each value in the data frame, indicating whether the value is missing or not.

describe(): Generates descriptive statistics of the data.

LabelEncoder(): Encodes categorical variables into numerical values.

train\_test\_split(): Splits the dataset into training and testing sets.

LinearRegression(): Initializes the linear regression model.

fit(): Trains the linear regression model.

predict(): Predicts the target variable using the trained model.

score(): Calculates the accuracy score of the model.

RandomForestRegressor(), SVR(), GradientBoostingRegressor(): Initializes various regression models.

dump(): Saves the trained model to a file.

load(): Loads the saved model from a file.

Entry(): Creates an entry widget for user input.

Button(): Creates a button widget with a callback function.

* **Analysis Steps:**
* Data Loading and Exploration:

Read the data from the CSV file using read\_csv() method.

Display the top and bottom rows of the dataset using head() and tail() methods.

Analyze the shape, information, and statistics of the dataset.

* Data Cleaning:

Check for missing values and handle them accordingly.

* Data Visualization:

No visualization required for this analysis.

* Preprocessing:

Encode categorical variables into numerical values using LabelEncoder().

* Feature and Target Separation:

Separate the dataset into features (independent variables) and target (dependent variable).

* Train/Test Split:

Split the dataset into training and testing sets using train\_test\_split() method.

* Model Training and Evaluation:

Train a linear regression model using LinearRegression() and evaluate its performance.

Train other regression models such as RandomForestRegressor(), SVR(), GradientBoostingRegressor() and compare their performances.

* Model Saving and Loading:

Save the best performing model using dump() method from Joblib library.

Load the saved model for future predictions using load() method.

* Prediction for New Customers:

Implement a Tkinter user interface for users to input their information.

Predict insurance costs for new customers based on their input using the trained model.

**Conclusion:**

The analysis successfully predicts insurance costs for individuals based on their age, gender, BMI, number of children, smoking habits, and region. The Tkinter user interface provides a user-friendly way for individuals to estimate their insurance costs.This structured approach facilitates a comprehensive understanding of the steps involved in predicting insurance costs and deploying the model in a user interface.