Estimation and Prediction of Hospitalization and Medical Care Costs

Naan Mudhalvan – Data Analytics Track

College: Madras Institute of Technology, Anna University (MIT, Anna University) Chrompet

Group Members:

Team Lead: Balasubramaniam N (2020506020)

Jawahar A S (2020506035)

Sarukesh S (2020506084)

Vijay N (2020506108)

1. Introduction:

Project Overview:

In this project, the aim is to develop a predictive model for estimating insurance charges based on a given dataset. By leveraging data analysis techniques, machine learning algorithms, and web development tools, we aim to create a user-friendly application that provides accurate predictions of insurance charges. The project also focuses on creating an interactive dashboard using IBM Cognos to present the analysis, findings, and model performance.

By accurately predicting insurance charges, insurance companies can optimize their pricing strategies, enhance risk assessment, and improve overall profitability. Moreover, individuals seeking insurance can benefit from a transparent and reliable estimation of charges, aiding them in selecting the most suitable insurance plans for their needs.

Through this project, we aim to showcase the process of data exploration, feature analysis, model development, and deployment using modern software tools and technologies. The results obtained will contribute to the body of knowledge in insurance prediction and demonstrate the potential of data-driven approaches in the insurance industry.

The following sections of this report will provide a detailed overview of the dataset used, the software tools and technologies employed, the methodology followed for data analysis and modeling, the results obtained, and the conclusions drawn from the project. Additionally, we will discuss the development of a Flask app for prediction and an interactive dashboard using IBM Cognos to present the project's findings effectively.

Purpose:

The insurance industry heavily relies on accurate predictions of insurance charges to determine appropriate premiums for policyholders. Predicting insurance charges involves analyzing various factors such as age, BMI, and other relevant features. Accurate estimation of insurance charges not only benefits insurance companies but also helps individuals in making informed decisions regarding their insurance coverage.

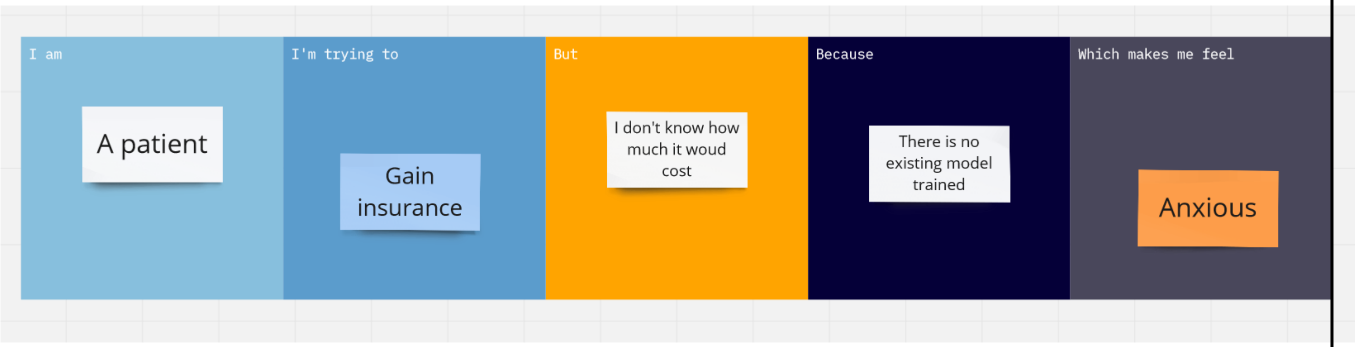
2. Ideation and Proposed Solution

2.1 Problem Statement Definition

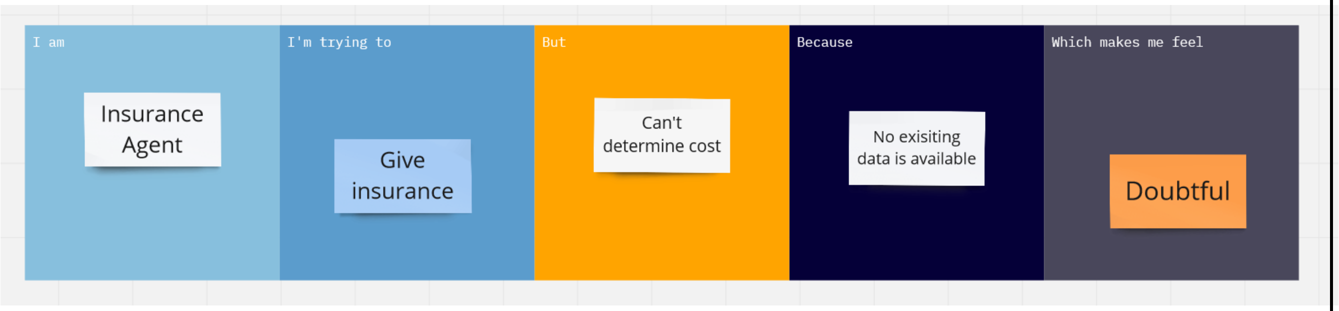
Ideation Phase
Define the Problem Statements

Date	23 April 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs
Maximum Marks	2 Marks

Customer Problem Statement:



I am	I'm trying to	But	Because	Which makes me feel
A patient	Gain insurance	I don't know how much it woud cost	There is no existing model trained	Anxious



I am	I'm trying to	But	Because	Which makes me feel
Insurance Agent	Give insurance	Can't determine cost	No exisiting data is available	Doubtful

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Patient	Gain insurance	I don't know how much it would cost	There is no existing model	Anxious

PS-2	Insurance Agent	Give insurance	Can’t determine cost	No existing data is available	Doubtful

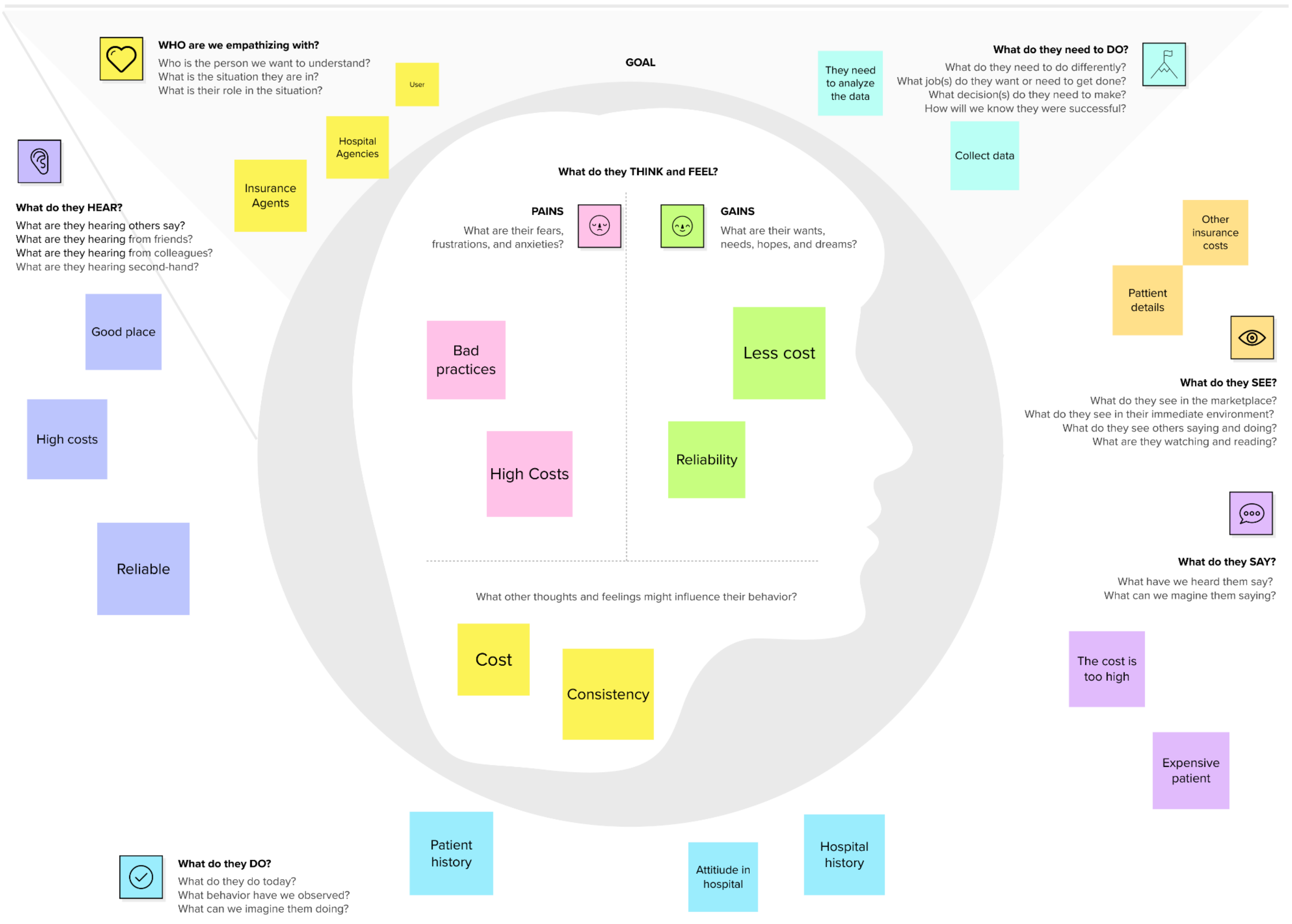
2.2 Empathy Map Canvas

Ideation Phase
Empathize & Discover

Date	23 April 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs
Maximum Marks	4 Marks

Empathy Map:

In next page



WHO are we empathizing with?

Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

User

Hospital Agencies

Insurance Agents

GOAL

What do they need to DO?

What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

They need to analyze the data

Collect data

What do they HEAR?

What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

Good place

High costs

Reliable

What do they THINK and FEEL?

PAINS

What are their fears, frustrations, and anxieties?

Bad practices

High Costs

GAINS

What are their wants, needs, hopes, and dreams?

Less cost

Reliability

What other thoughts and feelings might influence their behavior?

Cost

Consistency

What do they SEE?

What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

Other insurance costs

Pattient details

What do they SAY?

What have we heard them say?
What can we magine them saying?

The cost is too high

Expensive patient

What do they DO?

What do they do today?
What behavior have we observed?
What can we imagine them doing?

Patient history

Attitiude in hospital

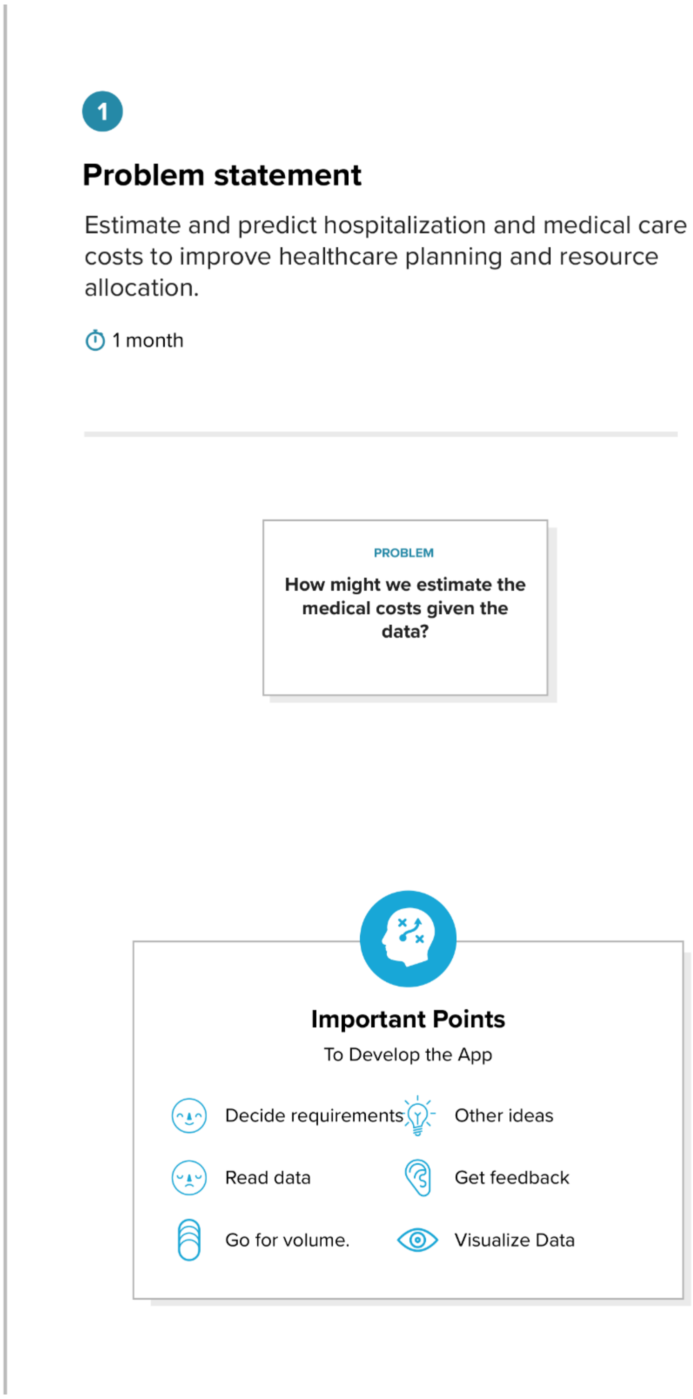
Hospital history

2.3 Ideation and Brainstorming

Ideation Phase
Brainstorm & Idea Prioritization

Date	23 April 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs
Maximum Marks	4 Marks

Step-1: Team Gathering, Collaboration and Select the Problem Statement



1

Problem statement

Estimate and predict hospitalization and medical care costs to improve healthcare planning and resource allocation.

⏱ 1 month

PROBLEM

How might we estimate the medical costs given the data?

Important Points

To Develop the App

Decide requirements

Other ideas

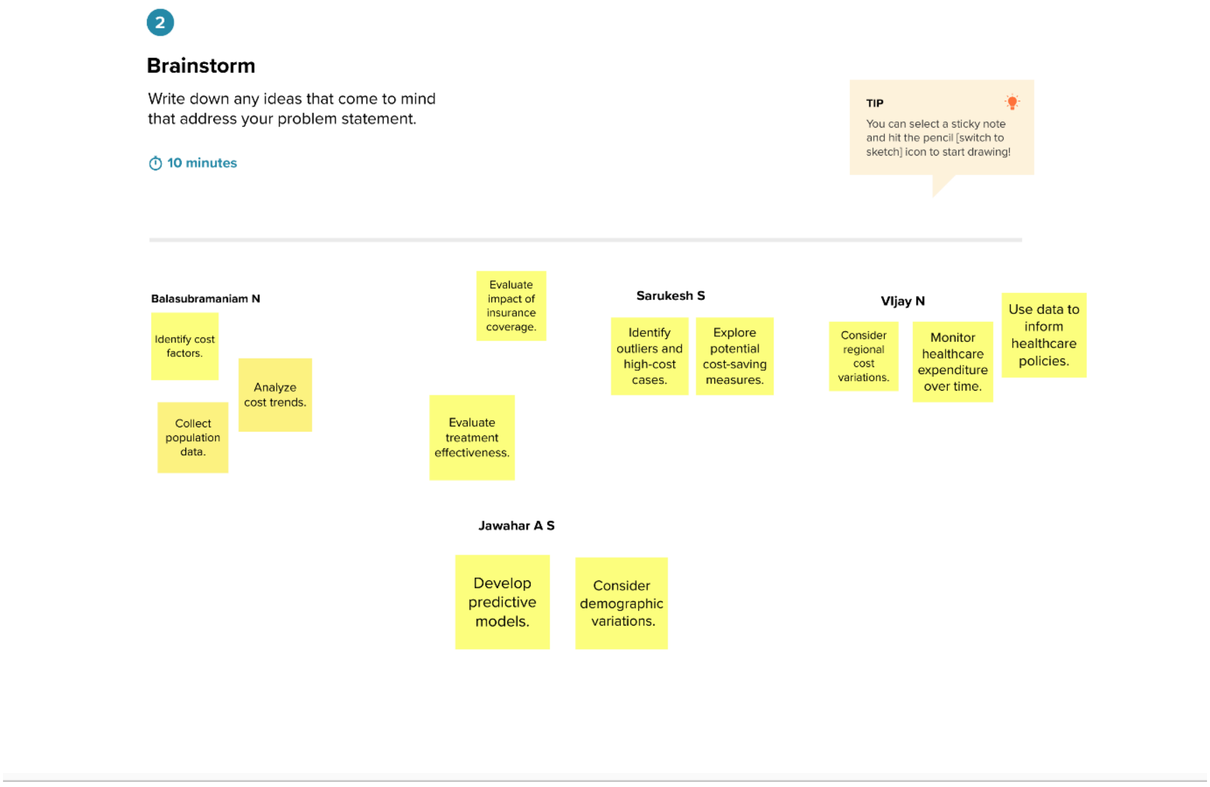
Read data

Get feedback

Go for volume.

Visualize Data

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Balasubramaniam N

- Identify cost factors.
- Analyze cost trends.
- Collect population data.
- Evaluate impact of insurance coverage.
- Evaluate treatment effectiveness.

Sarukesh S

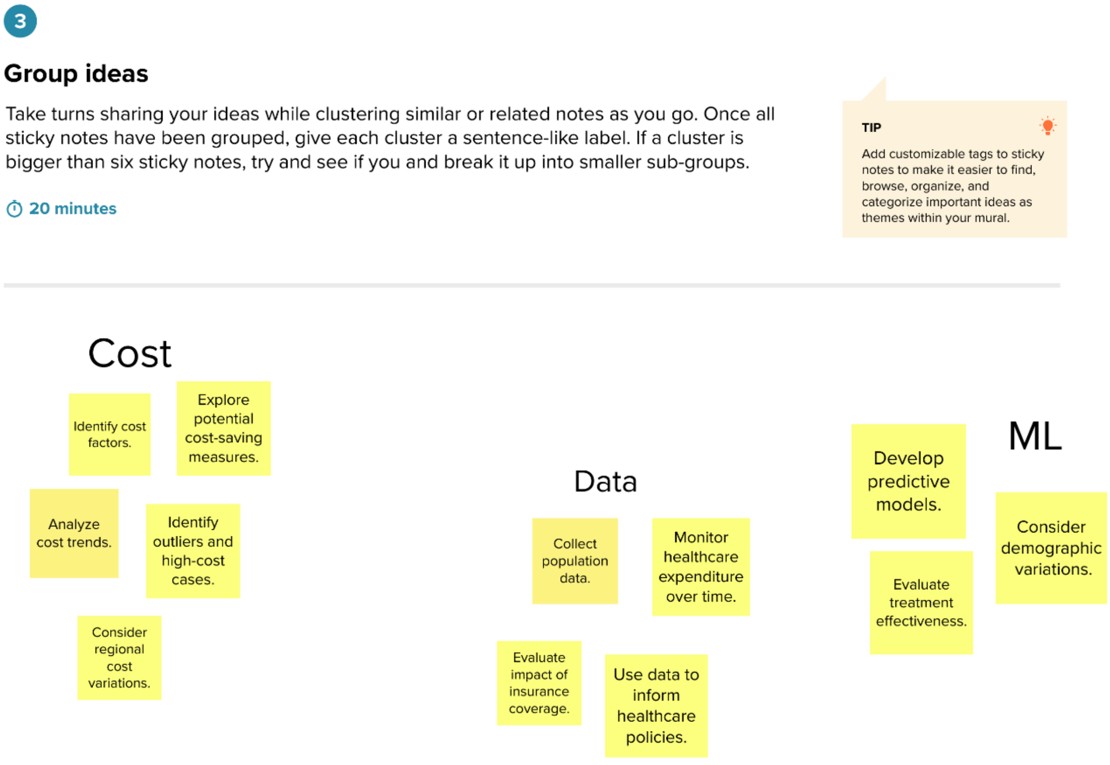
- Identify outliers and high-cost cases.
- Explore potential cost-saving measures.

VIjay N

- Consider regional cost variations.
- Monitor healthcare expenditure over time.
- Use data to inform healthcare policies.

Jawahar A S

- Develop predictive models.
- Consider demographic variations.



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Cost

- Identify cost factors.
- Explore potential cost-saving measures.
- Analyze cost trends.
- Identify outliers and high-cost cases.
- Consider regional cost variations.

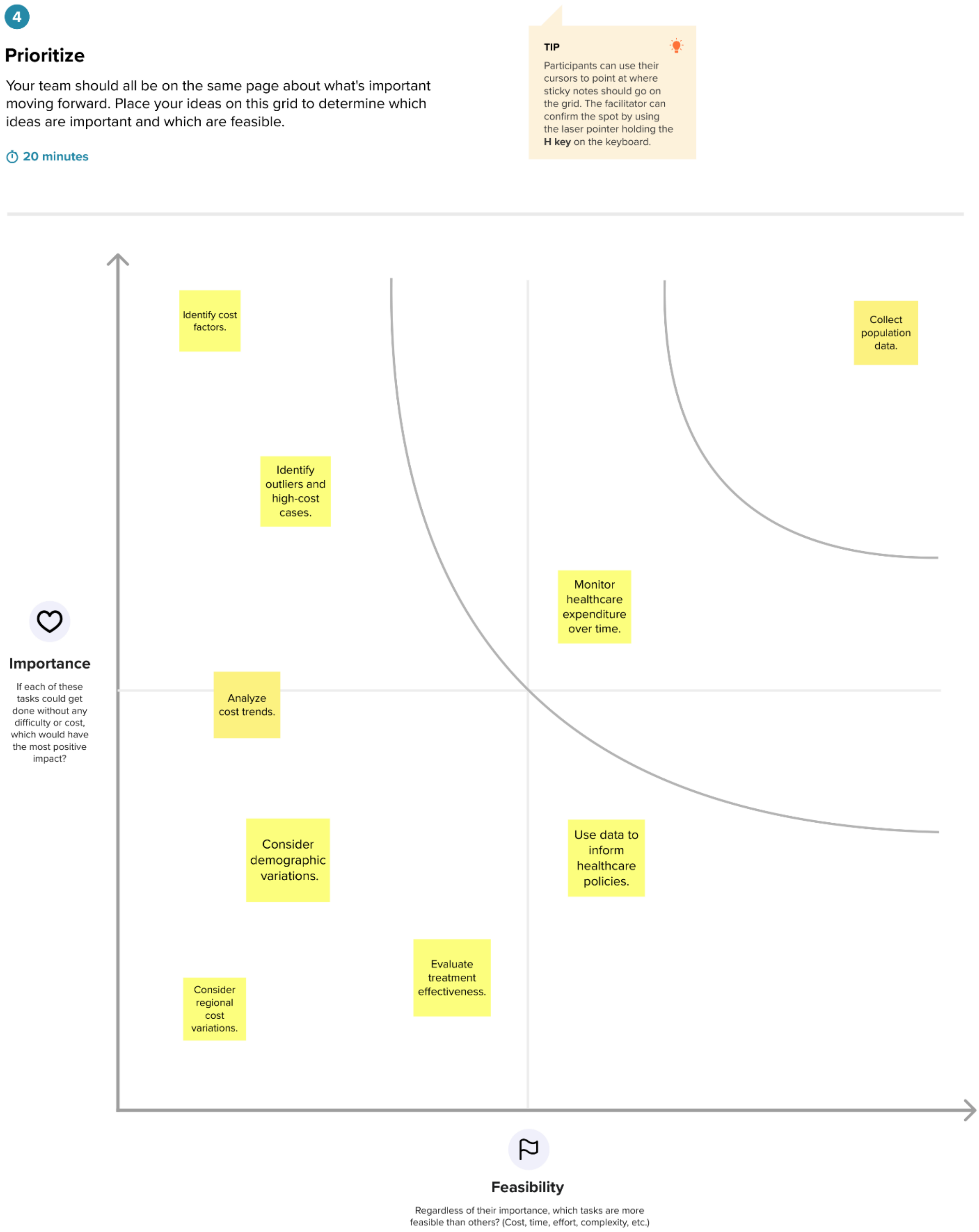
Data

- Collect population data.
- Monitor healthcare expenditure over time.
- Evaluate impact of insurance coverage.
- Use data to inform healthcare policies.

ML

- Develop predictive models.
- Consider demographic variations.
- Evaluate treatment effectiveness.

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⏱ 20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Identify cost factors.
Collect population data.
Identify outliers and high-cost cases.
Monitor healthcare expenditure over time.
Analyze cost trends.
Consider demographic variations.
Use data to inform healthcare policies.
Evaluate treatment effectiveness.
Consider regional cost variations.

2.4 Proposed Solution

Project Design Phase-I

Proposed Solution Template

Date	30 April 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs
Maximum Marks	4 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	Estimate and predict hospitalization and medical care costs to improve healthcare planning and resource allocation.
2.	Idea / Solution description	Develop a data-driven predictive model that takes into account key factors such as age, gender, medical history, and illness severity to estimate and predict hospitalization and medical care costs for a given population.
3.	Novelty / Uniqueness	The solution provides a more accurate and comprehensive understanding of healthcare expenditure, enabling healthcare providers and policymakers to make informed decisions and improve resource allocation.
4.	Social Impact / Customer Satisfaction	The solution can help reduce healthcare costs for patients and improve the quality of care they receive, leading to greater customer satisfaction and improved health outcomes.
5.	Business Model (Revenue Model)	The solution can be offered as a subscription-based service to healthcare providers, insurers, and policymakers, generating revenue through data analytics and consulting services.
6.	Scalability of the Solution	The solution can be scaled to serve a wide range of healthcare providers and policymakers, providing valuable insights into healthcare expenditure across different populations and regions.

3. Requirement Analysis

3.1 Functional Requirement

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	10 May 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Error handling
FR-2	User Login	Login only once registered Otherwise, direct to the login page Check database for data Error handling
FR-3	Inputting data	Input data using UI Print output

3.2 Non-Functional Requirements

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The app should have a user-friendly interface that is easy to navigate and understand for users.
NFR-2	**Security**	The app should ensure the confidentiality, integrity, and availability of user data and prevent unauthorized access, modification, or deletion of data.
NFR-3	**Reliability**	The app should be dependable and consistently perform its intended functions without errors or failure.
NFR-4	**Performance**	The app should provide fast and responsive user experience with minimal latency and load times.
NFR-5	**Availability**	The app should be available and accessible to users 24/7 with minimal downtime or maintenance windows.
NFR-6	**Scalability**	The app should be able to handle increasing amounts of user traffic and data without impacting performance or availability.

4. Project Design

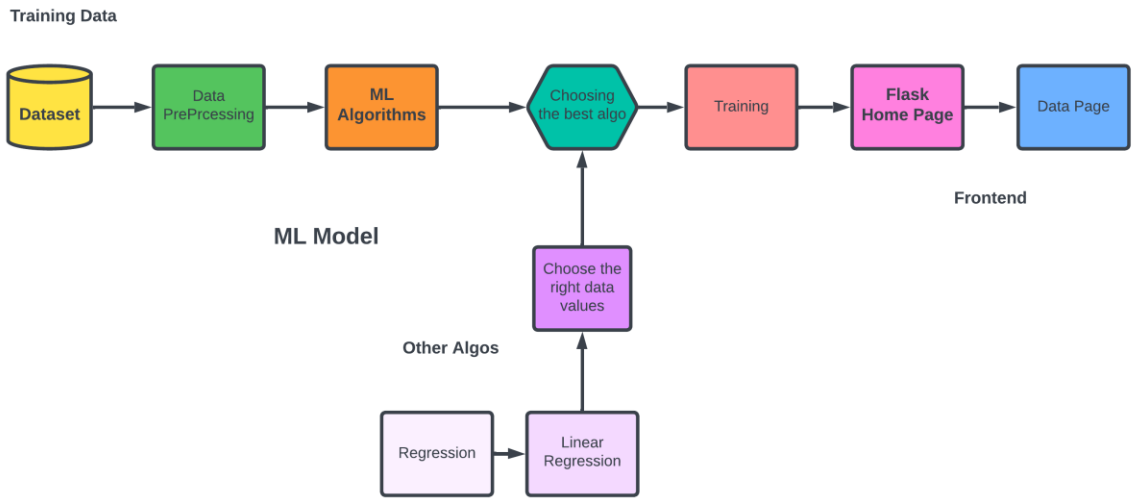
4.1 Data Flow Diagrams

Project Design Phase-II
Data Flow Diagram & User Stories

Date	10 May 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs
Maximum Marks	4 Marks

Data Flow Diagrams:

Given below is the dataflow Diagram for the given Estimation project.



4.2 Solution and Technical Architecture

Project Design Phase-I
Solution Architecture

Date	30 April 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs
Maximum Marks	2 Marks

Solution Architecture:

- Data collection from multiple sources

- Preprocessing and analysis of data using ML algorithms
- Development and deployment of a web application for cost prediction
- Visualization tools for identifying cost-saving measures and healthcare expenditure trends
- Deploying the model

Solution Architecture Diagram:

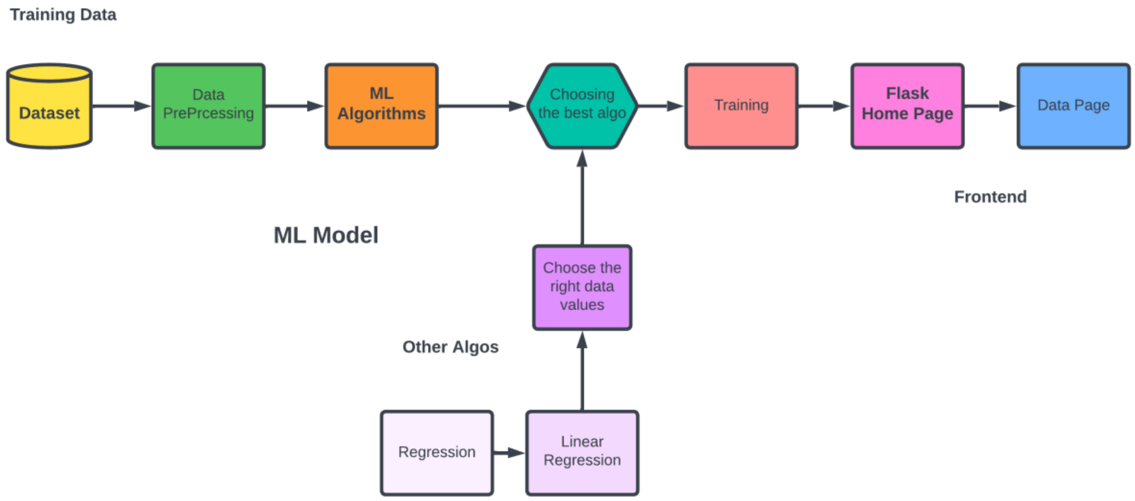


Figure 1: Architecture Model

4.3 User Stories

User Stories

Below are some of the stories for the project

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Individual	Estimate medical costs for a specific treatment	USN-1	As an individual, I want to be able to estimate the cost of medical care for a specific treatment so that I can plan for my expenses accordingly.	Access and website and predict charges	High	Balasubramaniam N
Employer	Predict medical costs for employee health insurance	USN-2	As an employer, I want to be able to predict the medical costs for my employees' health insurance so that I can budget accordingly and offer competitive benefits.	Estimate the costs	Medium	Balasubramaniam N
Insurance Provider	Estimate hospitalization costs for a specific patient	USN-3	As an insurance provider, I want to be able to estimate the hospitalization costs for a specific patient so that I can determine appropriate coverage and premiums.	Estimate the costs on the website	Medium	Jawahar A S
Government Health Agency	Predict healthcare costs for a population	USN-4	As a government health agency, I want to be able to predict healthcare costs for a population so that I can allocate resources appropriately and make informed policy decisions.	Predict the cost of the population	Medium	Sarukesh S
Medical Researcher	Estimate medical costs for a clinical trial	USN-5	As a medical researcher, I want to be able to estimate the medical costs for a clinical trial so that I can design studies that are feasible and financially viable.	Use research data to predict	Low	Vijay N

5. Coding and Solutioning

5.1 Feature 1 (Flask App):

Code:

```
# flask application for loading ml model and predicting the value

import os
import pickle
import numpy as np
from flask import (
    Flask,
    render_template,
    url_for,
    request,
    session,
    redirect,
    flash,
    url_for,
)
from pymongo import MongoClient
from dotenv import load_dotenv

app = Flask(__name__)
load_dotenv()

app.secret_key = "secret key"
mongoURI = os.getenv("MONGO_URI")
MONGODB_URI = mongoURI
client = MongoClient(MONGODB_URI)
db = client["insurance-prediction"]

# load the model from disk
```

```
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def index():
   return render_template('login.html')

@app.route("/signup", methods=["GET", "POST"])
def signup():
   if request.method == "POST":
      email = request.form["email"]
      pid = request.form["pid"]
      password = request.form["password"]
      db.users.insert_one({"email": email, "pid": pid, "password": password})
      return redirect(url_for("home"))
   return render_template("signup.html")

@app.route("/home")
def home():
   if "pid" in session:
      return render_template("index.html")
   else:
      return redirect("/login")

@app.route("/analysis")
def analysis():
   if "pid" in session:
      return render_template("analysis.html")
   else:
      return redirect("/login")

@app.route("/login", methods=["GET", "POST"])
```

```
def login():
    if request.method == "POST":
        pid = request.form["pid"]
        password = request.form["password"]
        user = db.users.find_one({"pid": pid, "password": password})
        if user is not None:
            session["pid"] = pid
            return redirect(url_for("home"))
        else:
            flash("Invalid Patient ID or Password. Please Try again.")
    return render_template("login.html")

@app.route("/logout")
def logout():
    session.pop("pid", None)
    return redirect(url_for("login"))

@app.route('/predict',methods=['POST'])
def predict():
    # get the data from the POST request
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    # predict the value
    prediction = model.predict(final_features)
    output = round(prediction[0])

    return render_template('result.html', prediction=output)

if __name__ == "__main__":
    app.run(debug=True)
```

This is a Flask application that loads a machine learning model (`model.pkl`) and uses it to make predictions based on user input. Here's a breakdown of the application's components and functionality:

1. **Flask Setup**: The necessary imports for Flask, MongoDB, and other dependencies are imported at the beginning of the code.

2. **App Configuration**: An instance of the Flask application is created, and the secret key is set to "secret key". The MongoDB URI is loaded from the environment variables.

3. **Model Loading**: The machine learning model is loaded from the `model.pkl` file using the `pickle.load()` function. The model will be used later for predictions.

4. **Routing and Views**: The Flask application defines several routes and corresponding views to handle different URL endpoints.

- The root URL ("/") renders the `login.html` template, which displays a login form.

- The "/signup" route handles both GET and POST requests. It allows users to sign up by entering their email, patient ID (pid), and password. Upon submission, the user data is stored in the MongoDB collection named "users".

- The "/home" route displays the `index.html` template, which serves as the main page after successful login. It checks if the user's patient ID (pid) is stored in the session and redirects to the login page if not.

- The "/analysis" route displays the `analysis.html` template, which could contain further analysis or additional functionality.

- The "/login" route handles both GET and POST requests for user login. It validates the provided patient ID and password against the stored data in the MongoDB collection. If the login is successful, the patient ID is stored in the session, and the user is redirected to the home page. Otherwise, an error message is flashed, and the login form is displayed again.

- The "/logout" route clears the patient ID from the session and redirects the user to the login page.

5. **Prediction Route**: The "/predict" route is responsible for handling the POST request containing the user input for prediction. It extracts the input data from the form, converts it to a numpy array, and then passes it to the model's `predict()` method. The predicted value is rounded, and the result is rendered using the `result.html` template.

6. **Application Run**: Finally, the application is run by calling `app.run(debug=True)`, which starts the Flask development server.

This Flask application provides basic user authentication, allows users to sign up and log in, and makes predictions using the loaded machine learning model. It uses MongoDB to store user data and sessions.

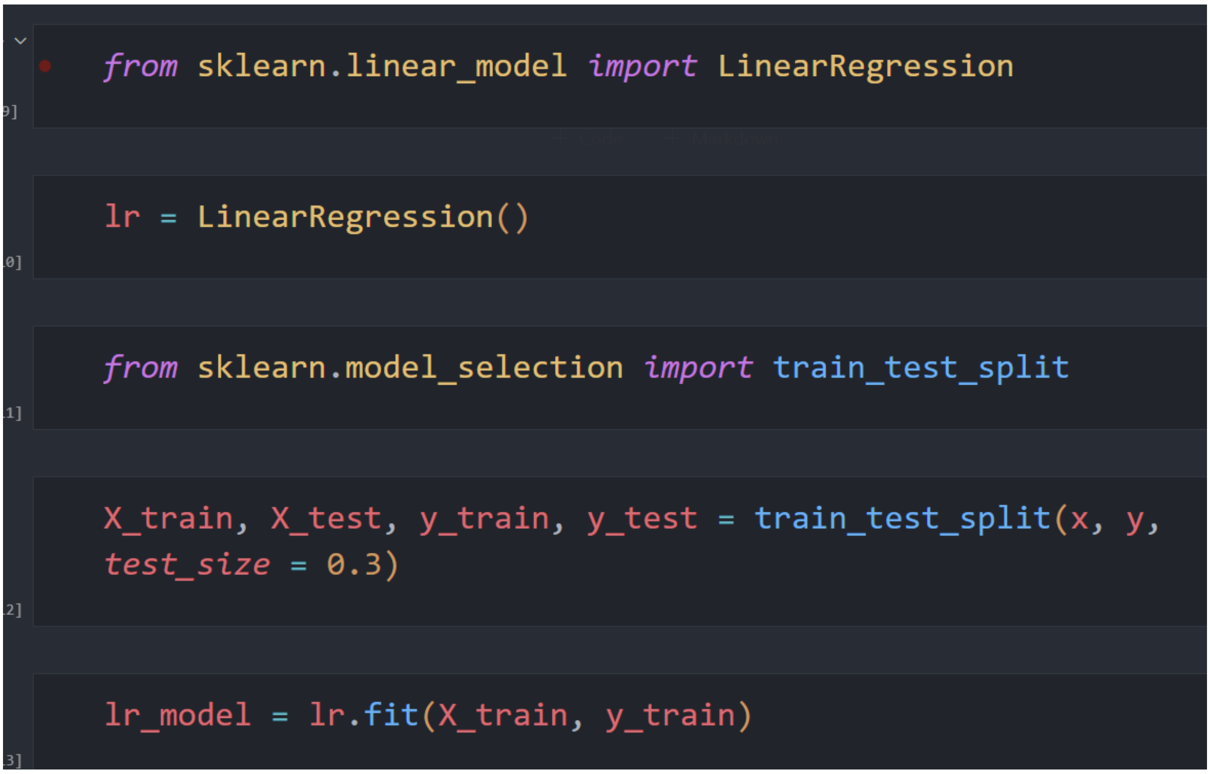
```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Insurance Premium Prediction</title>
    <style>
        /* style for form container */
        .form-container {
            margin: auto;
            width: 400px;
            padding: 20px;
            background-color: #f2f2f2;
            border-radius: 5px;
            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
        }

        /* style for form label */
        label {
            display: block;
            margin-bottom: 10px;
            font-weight: bold;
        }

        /* style for form input */
```

```
input[type="number"],
select {
   display: block;
   width: 100%;
   padding: 10px;
   margin-bottom: 20px;
   border-radius: 5px;
   border: none;
   box-shadow: 0px 0px 5px rgba(0, 0, 0, 0.2);
   font-size: 16px;
   font-family: Arial, sans-serif;
}

/* style for submit button */
input[type="submit"] {
   display: block;
   width: 100%;
   padding: 10px;
   margin-top: 20px;
   background-color: #4CAF50;
   color: white;
   border: none;
   border-radius: 5px;
   font-size: 16px;
   font-family: Arial, sans-serif;
   cursor: pointer;
}

/* style for error message */
.error {
   color: red;
```

```
            margin-top: 10px;
        }
    </style>
</head>

<body>
    <div class="form-container">
        <h2>Insurance Premium Prediction</h2>
        <form method="POST" action="/predict">
            <label for="age">Age:</label>
            <input type="number" id="age" name="age" required>

            <label for="sex">Sex:</label>
            <select id="sex" name="sex" required>
                <option value="0">Female</option>
                <option value="1">Male</option>
            </select>

            <label for="bmi">BMI:</label>
            <input type="number" id="bmi" name="bmi" step="0.01" required>

            <label for="children">Number of children:</label>
            <input type="number" id="children" name="children" required>

            <label for="smoker">Smoker:</label>
            <select id="smoker" name="smoker" required>
                <option value="0">No</option>
                <option value="1">Yes</option>
            </select>

            <label for="region">Region:</label>
            <select id="region" name="region" required>
                <option value="0">Northeast</option>
                <option value="1">Northwest</option>
                <option value="2">Southeast</option>
                <option value="3">Southwest</option>
            </select>

            <input type="submit" value="Submit">
        </form>
        <a href="/analysis">Analysis</a>



    {% if prediction %}
        <h2>Model Prediction</h2>
        <p>Insurance cost: {{ prediction }}</p>
    {% endif %}
    </div>
</body>

</html>
```

5.2 Feature 2 (Templates and ML):

```
def predict_charges(features):
    # convert the dictionary into a 2D array
    X = [[features['age'][0], features['sex'][0], features['bmi'][0], features['children'][0], features['smoker'][0], features['region'][0]]]
    prediction = lr.predict(X)
    return prediction[0]
```



```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(x, y,
test_size = 0.3)
```

```
lr_model = lr.fit(X_train, y_train)
```

5.3 Database Schema:

```
const userSchema = new Schema({
  pid: {
    type: String,
    required: true,
    unique: true
  },
  email: {
    type: String,
    required: true,
    unique: true
  },
  password: {
    type: String,
    required: true
  }
});

const User = mongoose.model('User', userSchema);
```

The `userSchema` specifies the structure of the document in the collection. It consists of three fields: `pid`, `email`, and `password`.

- `pid` represents the unique identifier for each user. It is of type `String`, and the `required: true` option ensures that a `pid` value is always present.
- `email` represents the email address of the user. It is also of type `String` and is required. The `unique: true` option ensures that each email address is unique within the collection.

- `password` represents the user's password. It is of type `String` and is required.

6. Results

6.1 Performance Metrics

Project Development Phase
Performance Test

Date	16 May 2023
Team ID	PBL-NT-GP--7627-1681100533
Project Name	Estimation and Prediction of Hospitalization and Medical Care Costs

Model Performance Testing:

Project team shall fill the following information in the performance testing template.

S.No.	Parameter	Screenshot / Values
1.	Dashboard design	No of Visualizations / Graphs - 12
2.	Data Responsiveness	The data is very responsive and can be viewed in mobile as well
3.	Utilization of Data Filters	Data Filters were used to filter the data in IBM Cognos
4.	Effective User Story	No of Scene Added - 5
5.	Descriptive Reports	No of Visulizations / Graphs - 3

7. Advantages and Disadvantages

Advantages of the Project:

1. Accurate Insurance Charge Predictions: By developing a predictive model, the project aims to provide accurate estimations of insurance charges. This can help insurance

companies optimize their pricing strategies and individuals make informed decisions when selecting insurance plans.

2. Data-Driven Approach: The project leverages data analysis techniques and machine learning algorithms to derive insights and predictions. This data-driven approach allows for evidence-based decision-making and can lead to more accurate results compared to traditional methods.
3. User-Friendly Application: The project aims to create a user-friendly application using Flask, making it easy for users to interact with the predictive model and obtain insurance charge predictions. This enhances the accessibility and usability of the solution.
4. Interactive Dashboard: The inclusion of an interactive dashboard using IBM Cognos allows for the effective visualization and presentation of analysis, findings, and model performance. This enables stakeholders to gain a comprehensive understanding of the project's outcomes.
5. Potential for Industry Impact: The results obtained from the project can contribute to the body of knowledge in insurance prediction and demonstrate the potential of data-driven approaches in the insurance industry. The insights and methodologies developed can be applied in real-world scenarios to improve pricing strategies and risk assessment.

Disadvantages of the Project:

1. Data Limitations: The accuracy and reliability of the predictive model heavily depend on the quality and representativeness of the dataset used. If the dataset is limited in size or lacks diversity, it may lead to biased or less accurate predictions.
2. Model Generalization: The developed predictive model may perform well on the given dataset but might face challenges when applied to new and unseen data. Ensuring the model's generalizability is crucial for its practical applicability.
3. Development Complexity: Developing a predictive model, creating a Flask application, and integrating an interactive dashboard using IBM Cognos can be complex tasks, requiring expertise in various areas. Managing and coordinating different components of the project may present challenges and require proficient technical skills.
4. Resource and Time Requirements: The project may require substantial computational resources and time for data processing, model training, and application development. Adequate resources and efficient project management are essential to meet project timelines and deliverables.
5. User Adoption: The success of the application depends on user adoption and acceptance. Convincing insurance companies and individuals to use the application and trust its predictions may pose a challenge, especially if they have existing systems or skepticism towards data-driven approaches.

8. Conclusion

In conclusion, this project aimed to predict insurance charges using a dataset containing features such as age, BMI, and other relevant factors. The analysis involved descriptive, bivariate, univariate, and multivariate techniques to gain insights into the dataset. A linear regression model was implemented to estimate charges accurately. The model was deployed using a Flask app, enabling users to obtain predicted charges based on their input. Additionally, an IBM Cognos dashboard, report, and story were created to present the analysis, model performance, and overall project findings. This project provides a comprehensive solution for insurance charge prediction and offers valuable insights for the insurance industry.

9. Future Work

Here are some potential future work ideas for the project:

1. **Enhancing the Predictive Model**: Continuously improving the predictive model by experimenting with different algorithms, feature engineering techniques, and model architectures. Exploring ensemble methods or advanced machine learning techniques like gradient boosting or deep learning may further enhance the model's accuracy and generalization capabilities.
2. **Feature Expansion**: Consider expanding the feature set used in the predictive model. Explore additional relevant features that could potentially contribute to better predictions, such as medical history, lifestyle factors, or additional demographic information. Careful feature selection and domain expertise will be important in determining the most valuable features to include.
3. **Model Evaluation and Validation**: Conduct thorough model evaluation and validation to ensure its reliability and performance. Employ cross-validation techniques, assess different evaluation metrics, and perform rigorous testing on unseen data or a hold-out dataset to validate the model's accuracy and generalizability.
4. **Performance Optimization**: Optimize the application's performance by implementing caching mechanisms, load balancing techniques, or parallel processing to handle larger volumes of requests efficiently. Monitor and analyze the application's performance metrics to identify bottlenecks and areas for improvement.
5. **Integration with External Data Sources**: Explore the possibility of integrating external data sources, such as healthcare databases, socio-economic indicators, or environmental data, to enrich the predictive model and capture more comprehensive insights. This can lead to a more holistic understanding of the factors affecting insurance charges.
6. **Security and Privacy Enhancements**: Implement robust security measures to protect user data and ensure privacy compliance. Employ encryption techniques, secure authentication protocols, and adopt best practices for secure data handling to build user trust and confidence in the application.
7. **User Feedback and Iterative Improvements**: Gather feedback from users, insurance professionals, and stakeholders to understand their experience with the application and identify areas for improvement. Iterate on the application based on user feedback and incorporate valuable suggestions to enhance its usability and functionality.
8. **Real-Time Updates and Monitoring**: Implement a mechanism to update the predictive model periodically with new data to keep it up to date. Develop monitoring systems to track the model's performance over time and detect any degradation or concept drift that may require model retraining or adjustments.
9. **Expand to Multiple Insurance Types**: Consider extending the project to cover multiple types of insurance, such as health insurance, auto insurance, or property

insurance. This would involve expanding the dataset, developing specific predictive models for each insurance type, and tailoring the application to accommodate different insurance domains.

10. **Collaboration with Insurance Companies**: Collaborate with insurance companies to validate the predictive model's effectiveness, gather real-world data, and assess its impact on their pricing strategies and risk assessment processes. This collaboration can provide valuable insights, improve the model's practical applicability, and foster industry adoption.

10. Appendix

All the code snippets and other pdfs, and the project report is present in the below GitHub link:

https://github.com/naanmudhalvan-SI/PBL-NT-GP--7627-1681100533

Video link:

https://drive.google.com/file/d/1zbLyYCfHZKuy_tLZXCsAzm-_Btflr9re/view?usp=sharing