A PROJECT REPORT

ON

Car Purchase Prediction Model

Under SmartInternz

AIML Externship 2023

Submitted By:

Team: 591615

Saxam Rawat(saxamrawat2021@vitbhopal.ac.in)

Vedang Gupta(vedang.gupta2021@vitbhopal.ac.in)

Under the Guidance Of:

Saumya Mohandas

Content

1. Introduction

1.1 Project Overview

The project centers around the development of an innovative Machine Learning (ML) solution designed to predict car purchases based on comprehensive customer data. Through the strategic utilization of features such as age, income, and historical purchase patterns, our team has created a robust model that ensures high predictive accuracy. This predictive model is seamlessly integrated into a user-friendly interface, facilitating easy access and interaction for users.

1.2 Purpose

The purpose of this project is twofold: firstly, to empower potential car buyers by providing them with a reliable estimation of their likelihood to make a purchase, and secondly, to revolutionize marketing strategies within the automotive industry. By leveraging advanced ML algorithms and meticulous data preprocessing, the model offers a dependable tool for predicting purchase behavior. The user-friendly interface enhances accessibility, allowing users to input their demographics and receive precise purchase likelihoods effortlessly.

2. Literature Survey

2.1 Existing Problem

The automotive industry currently faces several challenges that hinder effective customer engagement and marketing efficiency:

- **Inaccurate Predictions:** Traditional methods of predicting car purchases often lack precision, relying on generalized demographics and historical data that may not capture individual nuances. This leads to inefficient resource allocation and missed opportunities.

- **Limited Customer Understanding:** Dealerships often struggle to fully comprehend the diverse and evolving preferences of potential buyers. This lack of insight into individual customer profiles results in generic marketing strategies that may not resonate with specific target audiences.

- **Data Overload:** The abundance of customer data available can be overwhelming for businesses. Extracting meaningful insights from vast datasets requires sophisticated tools and algorithms to avoid information overload and ensure relevant decision-making.

- **Complex Decision-Making Process:** Potential car buyers are often faced with a complex decision-making process. A lack of accessible information and personalized guidance can lead to indecision, causing delays in the purchase journey or even abandonment.

2.2 References

The advent of online portals has enabled both buyers and sellers to access relevant information determining the market price of used cars. Machine learning algorithms, such as Lasso Regression, Multiple Regression, and Regression Trees, serve as examples. Our goal is to create a statistical model predicting the value of pre-owned vehicles using past customer data and various vehicle parameters. This project seeks to compare the predictive efficiency of different models to identify the most suitable one.

Numerous prior studies have delved into predicting used car prices. In Mauritius, Pudaruth employed methods like naive Bayes, k-nearest neighbors, multiple linear regression, and decision trees, but due to a limited number of observed cars, the predictive results were suboptimal. Pudaruth concluded that decision trees and naive Bayes are ineffective for continuous-valued variables.

Noor and Jan utilized multiple linear regression for price prediction, employing a variable selection method to identify influential variables and discarding the rest. Despite using only a few variables in the model, the linear regression achieved an outstanding 98 percent R-square.

Peerun et al. assessed the performance of neural networks in predicting used car prices, finding that support vector machine regression slightly outperformed neural networks and linear regression, especially on higher-priced cars. Various approaches, from multiple linear regression and k-nearest neighbor to naive Bayes, random forest, and decision tree, have been employed in the digital realm to accurately anticipate car prices.

In an attempt to facilitate user predictions, we developed a web application where users can assess the likelihood of a specific customer purchasing a car based on factors such as gender, age, and annual salary.

2.3 Problem Statement Definition

- **Objective:** Develop a machine learning solution to accurately predict car purchases based on individual customer data, addressing the shortcomings of existing methods.
- **Specific Challenges to Address:**
- **Precision:** Enhance the precision of car purchase predictions by leveraging advanced algorithms that consider a diverse set of features beyond traditional demographics.
- **Personalization:** Provide a solution that allows businesses to understand individual customer

profiles, enabling the tailoring of marketing strategies to specific preferences and behaviors.

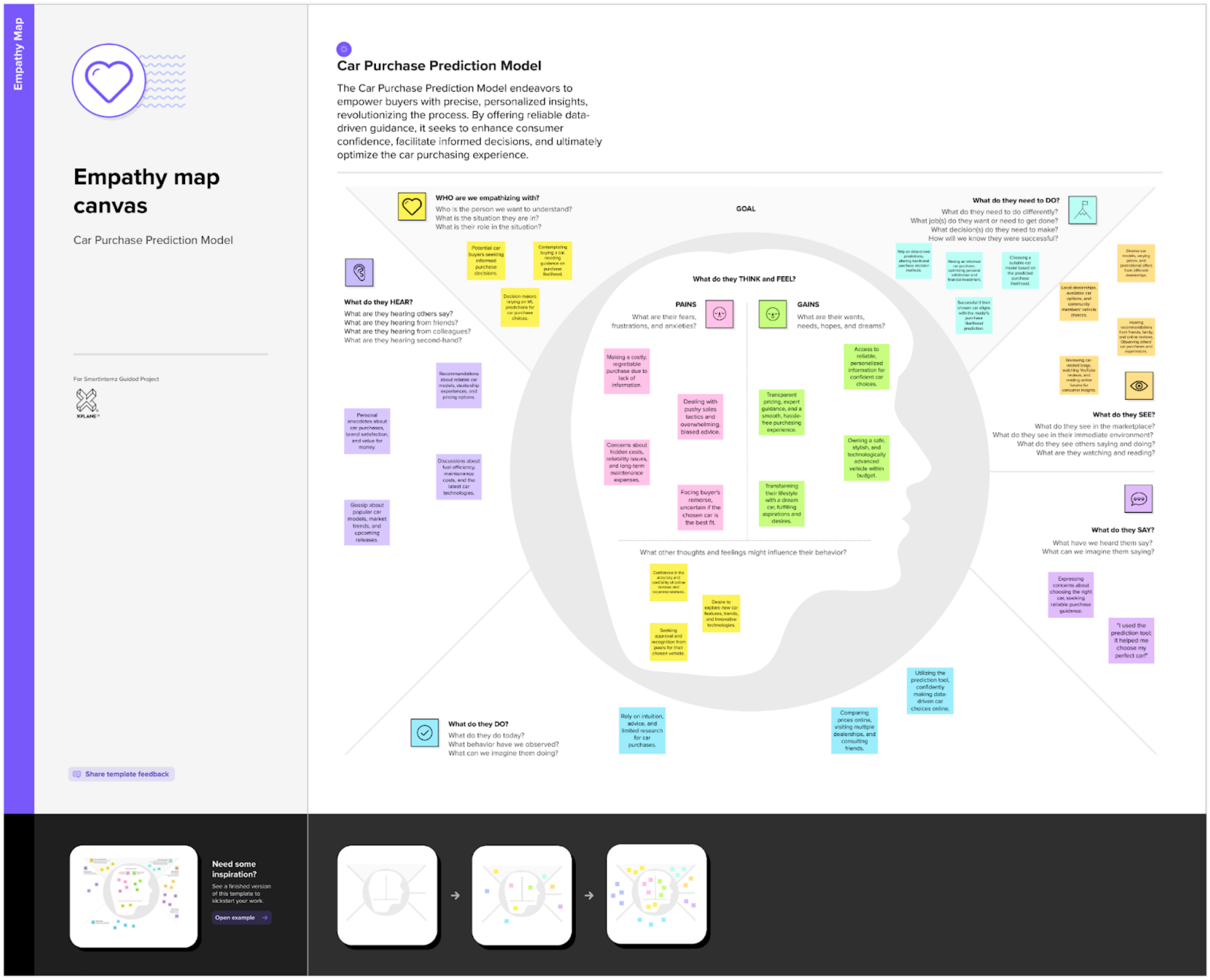
- **Data Management:** Develop tools for efficient data preprocessing and analysis to extract meaningful insights from large datasets, ensuring that businesses can make informed decisions without being overwhelmed by information.

- **User-Friendly Interface:** Create a user-friendly interface that simplifies the decision-making process for potential buyers, offering them personalized guidance and information to facilitate a smoother and more informed car purchase journey.

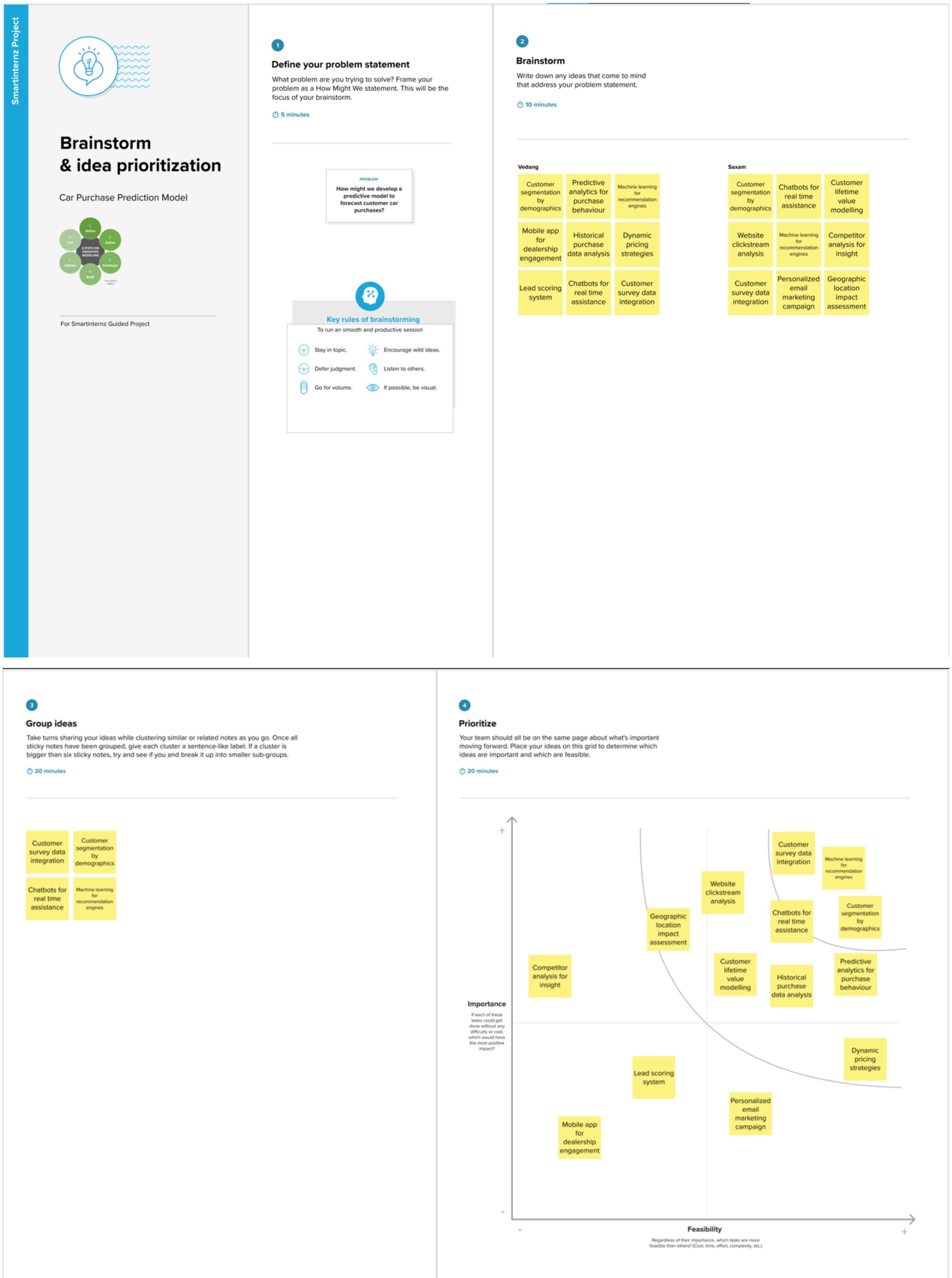
- By addressing these specific challenges, the proposed machine learning solution aims to revolutionize the prediction of car purchases, offering a more accurate, personalized, and user-friendly approach that benefits both customers and businesses in the automotive industry.

3. Ideation and Proposed Solution

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming



Smartinternz Project
Brainstorm
& idea prioritization
Car Purchase Prediction Model
1. Define
2. Gather
3. Prototype
4. Build
5. Validate
6. Use
6 STEPS FOR PREDICTIVE MODELLING
For Smartinternz Guided Project
1
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes
PROBLEM
How might we develop a predictive model to forecast customer car purchases?
Key rules of brainstorming
To run an smooth and productive session
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.
2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Vedang
Customer segmentation by demographics
Predictive analytics for purchase behaviour
Machine learning for recommendation engines
Mobile app for dealership engagement
Historical purchase data analysis
Dynamic pricing strategies
Lead scoring system
Chatbots for real time assistance
Customer survey data integration
Saxam
Customer segmentation by demographics
Chatbots for real time assistance
Customer lifetime value modelling
Website clickstream analysis
Machine learning for recommendation engines
Competitor analysis for insight
Customer survey data integration
Personalized email marketing campaign
Geographic location impact assessment
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
Customer survey data integration
Customer segmentation by demographics
Chatbots for real time assistance
Machine learning for recommendation engines
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
Customer survey data integration
Machine learning for recommendation engines
Website clickstream analysis
Chatbots for real time assistance
Customer segmentation by demographics
Geographic location impact assessment
Competitor analysis for insight
Customer lifetime value modelling
Historical purchase data analysis
Predictive analytics for purchase behaviour
Dynamic pricing strategies
Lead scoring system
Personalized email marketing campaign
Mobile app for dealership engagement
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.)

4. Requirement Analysis

4.1 Functional Requirement

4.1.1 Data Collection and Preprocessing:

4.1.1.1 Data Sources:

The system should integrate with data sources containing relevant customer information, including age, income, and historical purchase patterns.
The system must ensure the accuracy and integrity of the data sources.

4.1.1.2 Data Preprocessing:

Implement preprocessing steps to handle missing data, outliers, and ensure consistency in data formats.
Feature scaling and normalization should be applied to ensure uniformity across different features.

4.1.2 Machine Learning Model:

4.1.2.1 Algorithm Selection:

Utilize advanced machine learning algorithms suitable for predictive analysis, considering factors such as accuracy and interpretability.

4.1.2.2 Model Training:

Develop a training pipeline to iteratively train and validate the model using historical data.
Implement techniques such as cross-validation to assess the model's generalization performance.

4.1.3 User Interface:

4.1.3.1 Interface Design:

Create an intuitive and user-friendly interface for users to input their demographics and receive purchase likelihood estimates.
Ensure accessibility and responsiveness for a seamless user experience across devices.

4.1.3.2 Guidance and Interpretability:

Provide clear guidance and interpretation of the model's predictions to users.
Include visualizations or explanations that aid users in understanding how various factors contribute to their purchase likelihood.

4.2 Non-Functional Requirements:

4.2.1 Performance:

4.2.1.2 Accuracy:

Achieve a high level of accuracy in predicting car purchases to ensure the reliability of the model.

4.2.1.3 Response Time:

The system should provide predictions in a timely manner to enhance user experience.

4.2.2 Security:

4.2.2.1 Data Encryption:

Implement encryption measures to secure sensitive customer data during transmission and storage.

4.2.2.2 Access Control:

Enforce role-based access control to restrict system access based on user roles and responsibilities.

4.2.3 Scalability:

4.2.3.1 Model Scalability:

Design the system to handle an increasing volume of customer data and adapt to potential future enhancements.

4.2.3.2 User Interface Scalability:

Ensure the user interface can accommodate a growing user base and varying levels of concurrent usage.

5. Project Design

5.1 Proposed Solution

Sr No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	The automotive industry faces a challenge in accurately predicting car purchases based on customer data, hindering targeted marketing efforts and efficient resource allocation. Potential buyers often lack personalized guidance, leading to uncertainty in their purchasing decisions. Traditional methods fall short in providing precise forecasts, resulting in suboptimal customer engagement strategies for businesses.
2.	Idea / Solution description	We propose an innovative machine learning solution that leverages advanced algorithms and meticulous data preprocessing techniques to predict car purchases with high accuracy. By analyzing key features such as age, income, and historical purchase patterns, our solution provides potential buyers with tailored insights into their likelihood of making a purchase. This is achieved through a user-friendly interface where users input their demographics, receiving precise purchase likelihoods and informed recommendations. The solution aims to empower automotive businesses with data-driven insights to optimize their marketing strategies and enhance customer engagement.

3.	Novelty / Uniqueness	Our approach stands out due to its integration of advanced machine learning algorithms and thorough feature engineering, ensuring dependable predictions. The novelty lies in the precise estimation of purchase likelihoods for individual customers, enabling a highly personalized user experience. Additionally, the seamless user interface facilitates effortless interaction, enhancing accessibility for users of varying technical backgrounds.
4.	Social Impact / Customer Satisfaction	This innovative solution enhances customer experiences by providing them with valuable information, enabling more informed and confident car purchasing decisions. By empowering individuals with personalized insights, our project contributes to a more efficient and satisfying car-buying process, ultimately fostering consumer trust in the automotive industry. Additionally, by promoting informed choices, the solution indirectly supports environmental sustainability efforts by encouraging more thoughtful and efficient car purchases.
5.	Business Model (Revenue Model)	The business model revolves around providing our predictive analytics platform as a service to automotive companies. These businesses can subscribe to our solution to gain access to the predictive algorithms and user-friendly interface. Revenue generation can occur through subscription fees, tiered service packages, or pay-per-use models. Additionally, partnerships with dealerships and marketing agencies can be explored, offering them tailored insights and marketing strategies to enhance their customer engagement efforts. Continuous updates and customer support services can be offered as value-added options, ensuring long-term customer satisfaction and retention.

6.	Scalability of the Solution	Our machine learning solution, while accurate, owes its success to its scalability. A sturdy data processing system, refined algorithms, and user-friendly interface enable it to handle growing user bases and vast data sets. This adaptability ensures consistent performance, vital for businesses in the automotive sector. We're humbled by our contribution, empowering businesses of all sizes and enhancing customer experiences in this competitive industry. Our modest aim is to assist, learn, and grow alongside the businesses we support, fostering meaningful relationships and driving positive change in the automotive landscape.

5.2 Solution Architecture

1. Data Collection:

- Collect or create the dataset containing customer demographics and historical purchase patterns.

2. Data Visualization and Analysis:

- Perform univariate, bivariate, and multivariate analysis for insights into the dataset.
- Conduct descriptive analysis to understand data distributions.

3. Data Pre-processing:

- Check for null values and handle them appropriately.
- Address outliers using suitable techniques.
- Handle categorical data through encoding methods.
- Split the data into training and testing sets for model development.

4. Model Building:

● Import necessary libraries for Decision Tree Classifier and SVM Classifier.
● Initialize both classifiers and train them using the pre-processed data.
● Evaluate model performance using appropriate metrics (e.g., accuracy, F1-score).
● Save the trained models for future use.

5. Application Building:

● Create an HTML file for the user interface.
● Develop Python code to integrate the trained models into the HTML interface.
● Allow users to input data, process predictions using the models, and display purchase likelihood results.

6. Deployment and Integration:

● Deploy the application on a web server.
● Integrate the application with the backend system housing the trained models.
● Ensure seamless communication between the frontend (HTML interface) and the backend(Python code and machine learning models).

7. User Interaction:

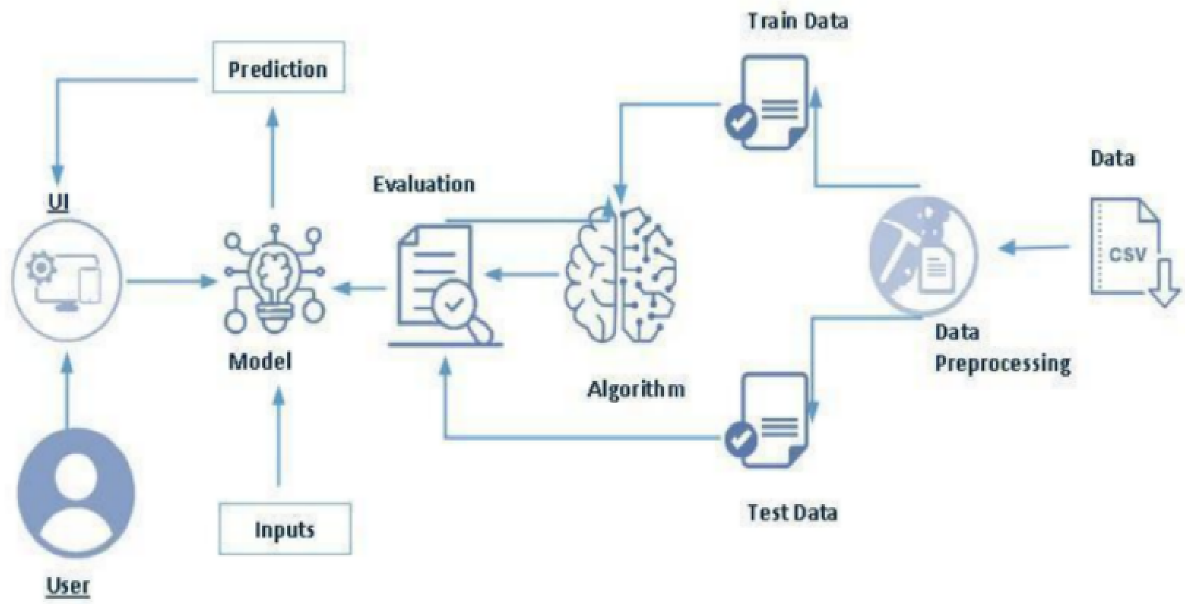
● Users input their demographics via the HTML interface.
● Python code processes the input using the trained Decision Tree Classifier and SVM Classifier.
● Display the purchase likelihood results back to the users via the HTML interface.

8. Monitoring and Maintenance:

● Implement monitoring tools to track user interactions and system performance.
● Conduct regular maintenance to update the models with new data and enhance prediction accuracy.
● This solution architecture outlines a streamlined process from data collection and analysis to model building, application development, and deployment. The Decision Tree Classifier and

SVM Classifier are key components in the machine learning model building phase, integrated into a user-friendly web application for predicting car purchases.

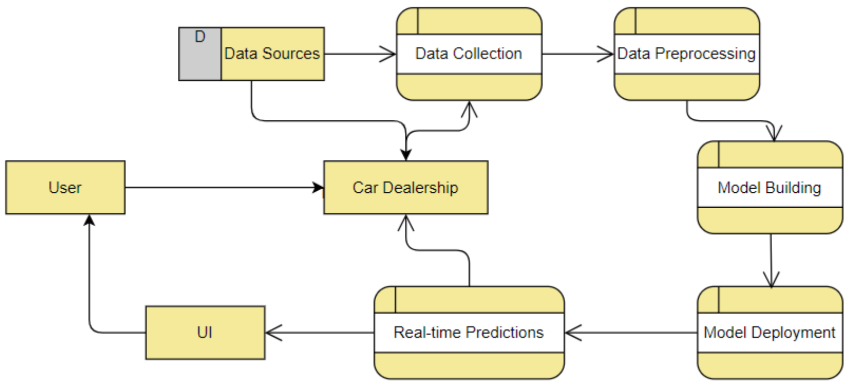
Solution Architecture Design:



5.3 Data Flow Diagram and User Stories

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Dashboard	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer Care Executive	Registration	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Web user)	Registration	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
Administrator	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1

6. Project Planning and Scheduling

6.1 Technical Architecture

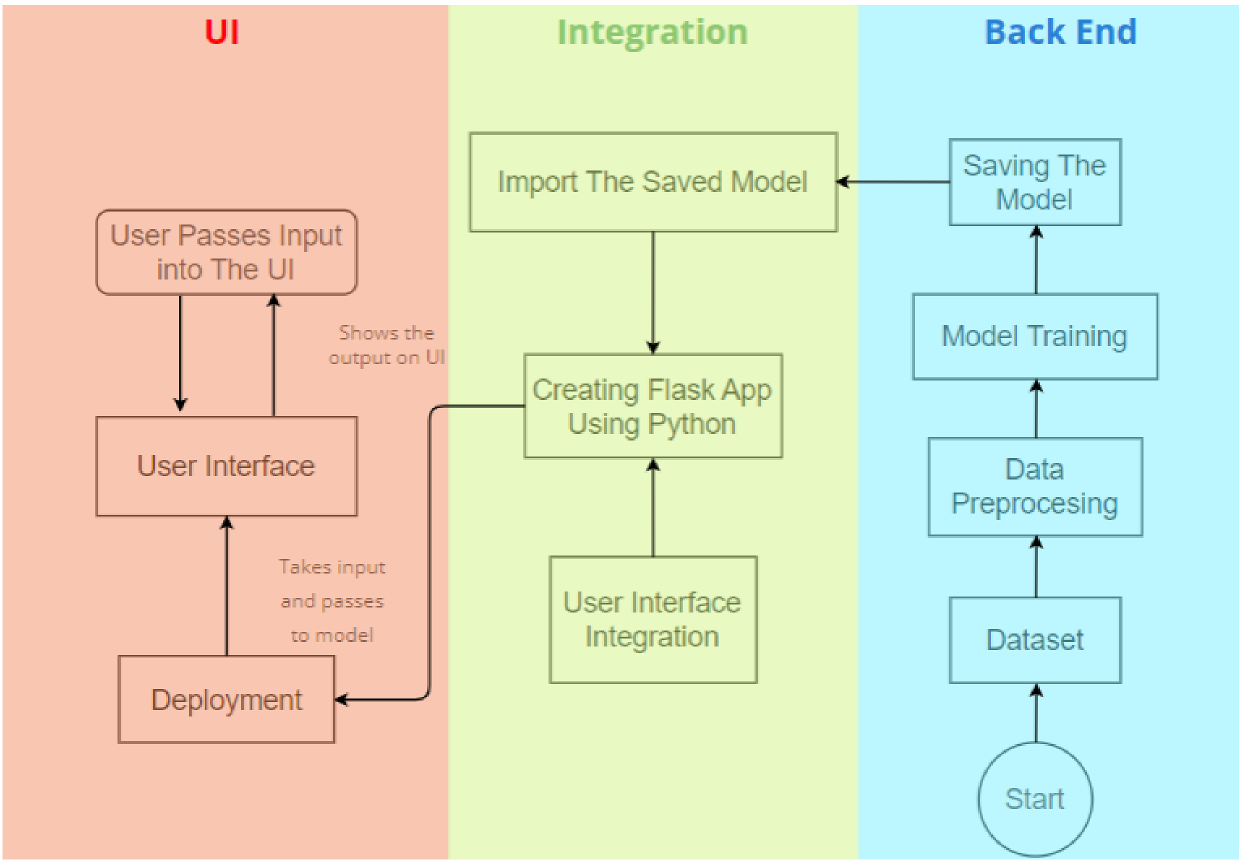


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Database	Collect the Dataset Based on the Problem Statement	File Manager, MySQL, NoSQL, etc.
4.	File Storage/ Data	File storage requirements for Storing the dataset	Local System, Google Drive Etc
5.	Frame Work	Used to Create a web Application, Integrating Frontend and Back End	Python Flask, Django etc
6.	Deep Learning Model	Purpose of Model	CNN, Transfer Learning etc.
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python's Flask
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

6.2 Sprint Planning and Architecture

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project Setup & Environment	USN-1	Set up the development environment with the required tools and frameworks to start the Car Purchase Prediction Model.	1	High	Saxam
Sprint-1	Data Collection	USN-2	Gather a diverse dataset of features such as age, income, and historical purchase patterns for accurate forecasts.	2	High	Vedang
Sprint-2	Data Preprocessing	USN-3	Preprocess the collected data.	2	High	Vedang
Sprint-2	Model Development	USN-4	Explore and evaluate different deep learning architecture(e.g. CNNs) to select the most suitable model for car purchase prediction.	3	High	Saxam
Sprint-3	Model Development	USN-5	Train the selected deep learning model using the preprocessed dataset and monitor its performance on the validation set.			
Sprint-3	Training	USN-6	Implement data augmentation techniques(e.g. Rotation, flipping) to improve the model's robustness and accuracy.	6	Medium	Saxam
Sprint-4	Model deployment & Integration	USN-7	Deploy the trained deep learning model as an API or web service to make it accessible for garbage classification. integrate the model's API into a user-friendly web interface for users to upload images and receive garbage classification results.	1	Medium	Vedang
Sprint-5	Testing & quality assurance	USN-8	Conduct thorough testing of the model and web interface to identify and report any issues or bugs. fine-tune the model hyperparameters and optimize its performance based on user feedback and testing results.	1	Medium	Saxam

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Start Date	End Date	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	3	2 Days	28 Oct 2023	29 Oct 2023	20	27 Oct 2023
Sprint-2	5	2 Days	30 Oct 2023	31 Oct 2023		
Sprint-3	10	2 Days	1 Nov 2023	2 Nov 2023		
Sprint-4	1	2Days	3 Oct 2023	4 Oct 2023		
Sprint-5	1	3 Days	5 Oct 2023	6 Oct 2023		

Velocity:

Imagine we have a 29-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per
iteration unit (story points per day)

AV= 11/20 = 0.55

7. Coding and Solutioning

7.1 Decision Tree Model

Note: We tested two models, but eventually went with DT Model because of its superior accuracy

```
#importing libraries
from sklearn.tree import DecisionTreeClassifier
import sklearn.tree as tree
from sklearn.model_selection import cross_val_score
```

```
#using GridSearchCV to find out the best parameters
from sklearn.model_selection import GridSearchCV
dt_classifier = DecisionTreeClassifier()

param_grid = {
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2']
}

grid_search = GridSearchCV(dt_classifier, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)

print("Best Hyperparameters:", grid_search.best_params_)

best_dt_classifier = grid_search.best_estimator_
accuracy = best_dt_classifier.score(X_test, y_test)
print(f"Test Accuracy with Best Hyperparameters: {accuracy:.2f}")
```

```
Best Hyperparameters: {'criterion': 'gini', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}
Test Accuracy with Best Hyperparameters: 0.91
```

```
#building the model

model1 = DecisionTreeClassifier(criterion = 'entropy', max_depth = None, max_features = 'log2', min_samples_leaf = 1, min_samples_split = 10, splitter = 'best')
model1
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_features='log2',
                       min_samples_split=10)
```

```
# Perform 5-fold cross-validation
cross_val_scores = cross_val_score(model1, X, y, cv=10)

# Print the cross-validation scores
print("Cross-validation scores:", cross_val_scores)

# Calculate and print the average accuracy across all folds
print("Average accuracy: {:.2f}".format(cross_val_scores.mean()))
```

```
Cross-validation scores: [0.87 0.85 0.89 0.87 0.9  0.91 0.86 0.92 0.85 0.87]
Average accuracy: 0.88
```

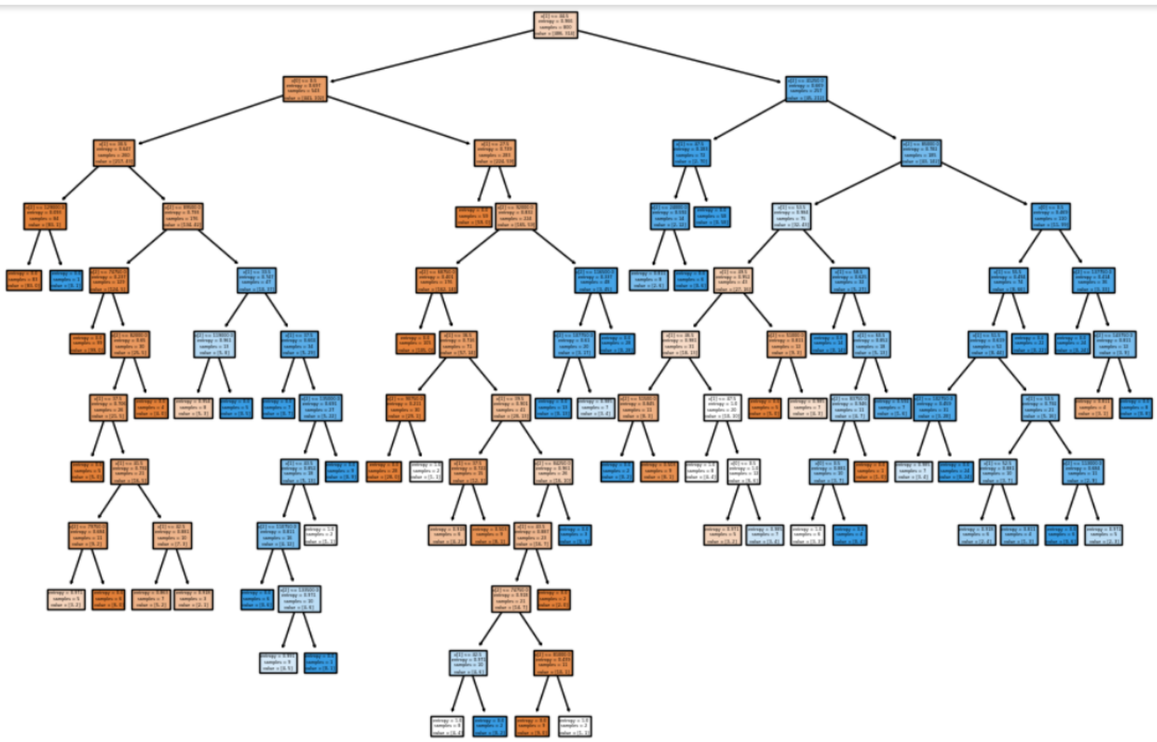
```
#fitting the model

model1.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_features='log2',
                       min_samples_split=10)
```

```
#predicting for both train and test sets
model1pred_train = model1.predict(X_train)
model1pred_test = model1.predict(X_test)
```

```
#visualizing the decision tree
fig, ax = plt.subplots(figsize=(12, 8))
tree.plot_tree(model1, filled=True, ax=ax)
mplcursors.cursor(hover=True)
mplcursors.cursor(ax, multiple=True)
plt.show()
```



7.2 Flask Application

7.2.1 Structure Layout

index.html app.py result.html CarPurchasePredictionModel.ipynb

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Car Purchase Prediction</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }

        h1 {
            text-align: center;
            color: #333;
        }

        form {
            max-width: 400px;
            margin: 20px auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        label {
            display: block;
            margin-bottom: 8px;
            font-weight: bold;
        }

        input, select {
            width: 100%;
            padding: 8px;
            margin-bottom: 16px;
            box-sizing: border-box;
        }

        input[type="submit"] {
            background-color: #4caf50;
            color: white;
            cursor: pointer;
        }

        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <h1>Car Purchase Prediction</h1>
    <form action="/predict" method="post">
        Age: <input type="text" name="age" required><br>
        Gender: <select name="gender" required>
            <option value="Male">Male</option>
            <option value="Female">Female</option>
        </select><br>
        Salary: <input type="text" name="salary" required><br>
        <input type="submit" value="Predict">
    </form>

</body>
</html>
```

index.html | app.py | result.html | CarPurchasePredictionModel.ipynb

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prediction Result</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }

        h1 {
            text-align: center;
            color: #333;
        }

        p {
            max-width: 400px;
            margin: 20px auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        a {
            display: block;
            text-align: center;
            margin-top: 20px;
            text-decoration: none;
            color: #4caf50;
        }

        a:hover {
            color: #45a049;
        }
    </style>
</head>
```

7.2.2 Application Logic

index.html | app.py | result.html | CarPurchasePredictionModel.ipynb

```python
from flask import Flask, render_template, request, jsonify
import pickle
import numpy as np
from sklearn import preprocessing
import os

```

```python
app = Flask(__name__)

# Define the absolute path to the model file
model_path = os.path.join(os.path.dirname(__file__), 'DTModel.pkl')

# Load the model
with open(model_path, 'rb') as model_file:
    model = pickle.load(model_file)
```

```python
@app.route('/')
def home():
    return render_template('index.html')
```

```python
@app.route('/predict', methods=['POST'])
def predict():
    # Get user input from the form
    age = float(request.form['age'])
    gender = request.form['gender']
    salary = float(request.form['salary'])

    # Encode gender using the LabelEncoder
    gender_encoded = le_Sex.transform([gender])[0]

    # Create a NumPy array with the input values
    input_data = np.array([[age, gender_encoded, salary]])

    # Make the prediction
    prediction = model.predict(input_data)

    # Display the prediction
    result = "Will Buy a Car" if prediction[0] > 0.5 else "Will Not Buy a Car"
```

8. Performance Testing

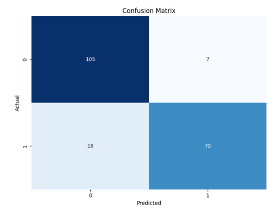
8.1 Performance Metrics

Note: Zoom in to clearly see the metrics

<table>
<tr><th>S.No</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Metrics</td><td>Classification Model:
Decision Tree

Confusion Matrix

Accuracy Score

Classification Report</td><td>

DecisionTrees's Accuracy: 0.875

Train Set Classification Report:
precision recall f1-score support
0 0.95 0.98 0.96 486
1 0.97 0.92 0.94 314
accuracy 0.96 800
macro avg 0.96 0.95 0.95 800
weighted avg 0.96 0.96 0.96 800

Test Set Classification Report:
precision recall f1-score support
0 0.85 0.94 0.89 112
1 0.91 0.80 0.85 88
accuracy 0.88 200
macro avg 0.88 0.87 0.87 200
weighted avg 0.88 0.88 0.87 200</td></tr>
<tr><td>2.</td><td>Tune The Model</td><td>Hyperparameter Tuning: Grid Search CV

Validation Method:
5 fold cross validation.</td><td>from sklearn.model_selection import GridSearchCV
# Create a Decision Tree classifier
dt_classifier = DecisionTreeClassifier()

# Define the hyperparameters and their possible values to tune
param_grid = {
'criterion': ['gini', 'entropy'],
'splitter': ['best', 'random'],
'max_depth': [None, 10, 20, 30],
'min_samples_split': [2, 5, 10],
'min_samples_leaf': [1, 2, 4],
'max_features': ['auto', 'sqrt', 'log2']
}

# Use GridSearchCV to search for the best hyperparameters
grid_search = GridSearchCV(dt_classifier, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Print the best hyperparameters
print("Best Hyperparameters:", grid_search.best_params_)

# Evaluate the model on the test set with the best hyperparameters
best_dt_classifier = grid_search.best_estimator_
accuracy = best_dt_classifier.score(X_test, y_test)
print(f"Test Accuracy with Best Hyperparameters: {accuracy:.2f}")

Best Hyperparameters: {'criterion': 'entropy', 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'best'}
Test Accuracy with Best Hyperparameters: 0.84

# Perform 5-fold cross-validation
cross_val_scores = cross_val_score(model1, X, y, cv=10)

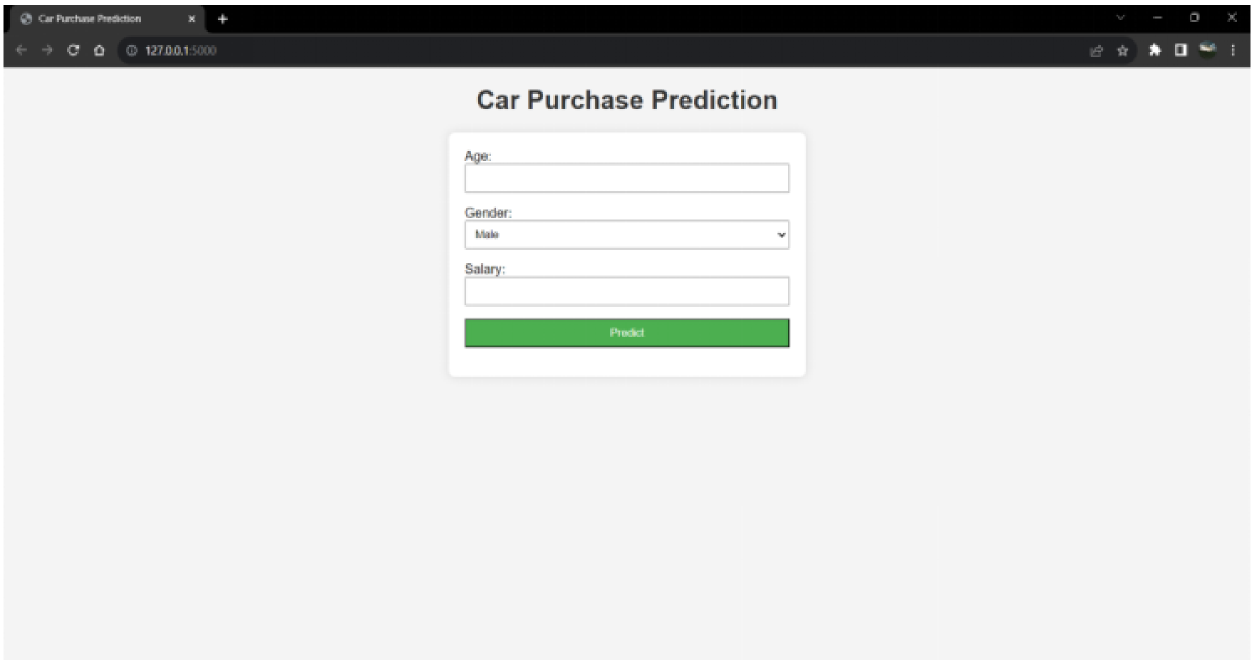
# Print the cross-validation scores
print("Cross-validation scores:", cross_val_scores)

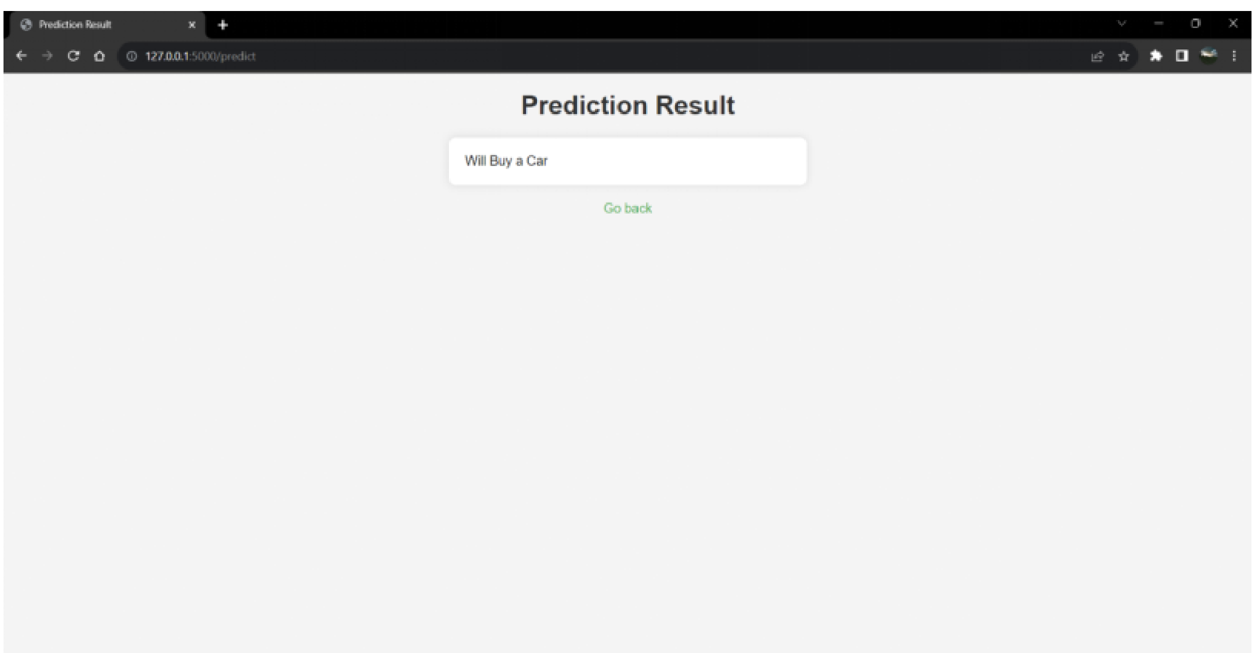
# Calculate and print the average accuracy across all folds
print("Average accuracy: {:.2f}".format(cross_val_scores.mean()))

Cross-validation scores: [0.92 0.81 0.92 0.88 0.91 0.91 0.84 0.87 0.9 0.94]
Average accuracy: 0.89</td></tr>
</table>

9. Results

9.1 Output Screenshots





10. Advantages & Disadvantages

10.1 Advantages:

1. High Predictive Accuracy:

   The use of advanced machine learning algorithms and thorough data preprocessing contributes to a high level of accuracy in predicting car purchases, providing dependable insights for users.

2. Comprehensive Data Utilization:

   The inclusion of diverse features such as age, income, and historical purchase patterns ensures a more comprehensive understanding of customer behavior, leading to more nuanced predictions.

3. User-Friendly Interface:

   The seamless integration of the predictive model into a user-friendly interface enhances the overall user experience, making it accessible and easy for potential buyers to interact with the system.

4. Tailored Marketing Strategies:

   By offering insights into individual customer profiles, the project enables automotive businesses to tailor marketing strategies, resulting in more targeted and effective customer engagement.

5. Informed Decision-Making:

The model facilitates informed choices for potential buyers, contributing to a more transparent and trustworthy relationship between customers and dealerships.

6. Revolutionizing Automotive Industry Practices:

The project represents a groundbreaking application of machine learning, driving data-powered decisions and setting a new standard for predictive analytics in the automotive sector.

10.2 Disadvantages:

1. Data Privacy Concerns:

The collection and utilization of customer data raise privacy concerns. It is crucial to implement robust security measures to protect sensitive information and comply with data protection regulations.

2. Dependency on Data Quality:

The accuracy of predictions is highly dependent on the quality and relevance of the input data. Inaccuracies or biases in the dataset may lead to skewed predictions.

3. Algorithm Complexity:

The use of advanced machine learning algorithms may result in a complex model. Understanding and maintaining such complexity might pose challenges, particularly for non-technical stakeholders.

4. Limited Scope Without Continuous Training:

The predictive model's accuracy may diminish over time without continuous training and adaptation to evolving customer behaviors and market dynamics.

5. Potential User Resistance:

Some potential buyers may be resistant to providing personal information, even if it is for predictive analysis. This could impact the accuracy of predictions if user participation is low.

6. Resource Intensiveness:

Developing and maintaining a sophisticated predictive model, integrating it into a user-friendly interface, and ensuring data security can be resource-intensive in terms of time, expertise, and costs.

11. Conclusion

In conclusion, our predictive car purchase model stands as an innovative fusion of advanced machine learning and the automotive industry. By accurately predicting car purchases through diverse features like age, income, and historical purchase patterns, the project has the potential to reshape marketing strategies and enhance user experiences.

The project's advantages lie in its high predictive accuracy, user empowerment for informed decisions, and a user-friendly interface. While addressing potential challenges like data privacy concerns and dependency on data quality, the model emphasizes the need for continuous monitoring and adaptation to remain effective.

Moving forward, regular updates, user education, and a proactive stance toward emerging issues will be vital for success. The project not only predicts car purchases but also aims to adapt to dynamic market conditions, ensuring relevance in the ever-evolving automotive landscape.

In summary, our predictive car purchase model represents a transformative application of machine learning, offering tangible benefits for both consumers and automotive businesses. Through technological innovation and user-centric design, the project sets a new standard for data-powered decision-making in the automotive sector.

12. Future Scope

The predictive car purchase model lays a strong foundation for future enhancements and expansions, positioning itself as a dynamic tool for the evolving automotive industry. The following aspects outline the potential future scope of the project:

12.1 Enhanced Feature Set:

12.1.1 Integration of Additional Features:

Consider expanding the feature set by incorporating additional customer data, such as online behavior, preferences, and social media interactions, to refine predictions further.

12.1.2 Real-Time Data Feeds:

Explore possibilities for incorporating real-time data feeds, enabling the model to adapt swiftly to changing market trends and customer behaviors.

12.2 Continuous Model Improvement:

12.2.1 Dynamic Model Training:

Implement a system for continuous model training to adapt to evolving customer preferences and market dynamics, ensuring the model's sustained accuracy over time.

12.2.2 Feedback Mechanism:

Integrate a feedback mechanism that allows users to provide insights on the accuracy of predictions, fostering a user-driven improvement cycle.

12.3 Expanded Market Applications:

12.3.1 Model Generalization:

Explore opportunities to generalize the predictive model for applications beyond car purchases, such as predicting preferences for various automotive features or accessories.

12.3.2 Industry Collaboration:

Consider collaboration with other industries to apply similar predictive models to different consumer markets, leveraging the robustness and adaptability of the existing framework.

12.4 AI-driven Personalization:

12.4.1 Personalized Marketing Strategies:

Develop advanced algorithms to tailor marketing strategies at an individual level, providing a hyper-personalized experience for potential buyers.

12.4.2 Predictive Customer Lifetime Value:

Extend the model to predict the lifetime value of customers, enabling businesses to focus on long-term customer relationships and loyalty.

12.5 Blockchain Integration for Data Security:

12.5.1 Enhanced Data Security Measures:

Explore the integration of blockchain technology to enhance data security, providing users with transparent and immutable control over their personal information.

12.5.2 Smart Contracts:

Investigate the use of smart contracts to manage data sharing agreements, ensuring privacy compliance while maintaining the integrity of the predictive model.

12.6 Global Market Expansion:

12.6.1 Localization Strategies:

Develop strategies to localize the predictive model for different global markets, considering cultural, economic, and regulatory variations.

12.6.2 Multilingual Interface:

Implement a multilingual interface to cater to diverse linguistic preferences, expanding the reach of the predictive car purchase model.

12.7 Collaboration with Automotive Manufacturers:

12.7.1 Integration with Manufacturer Systems:

Explore collaboration opportunities with automotive manufacturers to integrate the predictive model directly into their systems, streamlining the decision-making process for dealerships and manufacturers alike.

12.7.2 Vehicle Customization Predictions:

Extend predictions to include potential interest in specific vehicle customizations or packages, providing manufacturers with valuable insights for product development.

13. Appendix

13.1 Source Code:

https://github.com/smartinternz02/SI-GuidedProject-596384-1697379517/blob/main/8.%20CarPurchasePredictionModel.ipynb

13.2 Github and Demo Link:

13.2.1 Github Link:

https://github.com/smartinternz02/SI-GuidedProject-596384-1697379517

13.2.2 Demo Link:

https://drive.google.com/file/d/1A6KOXgXiErKfG50sAk1fNxN7U2UP7hV0/view?usp=sharing