PREDICT THE BUDGET A LONG DRIVE

A MINI PROJECT REPORT

Submitted by

DEEPAK RAJ S [513121106301]

MANIMARAN V [513121106302]

PREM KUMAR S [513121106306]

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY, VELLORE

ANNA UNIVERSITY :: CHENNAI 600025

MARCH 2024

TABLE OF CONTENTS

ABSTRACT

This project presents a multi-output regression model aimed at optimizing travel planning decisions. The model utilizes features such as starting and ending places, number of members, fuel type, hotel type, and miscellaneous charges to predict multiple target variables including total distance, travel days, food expenses, travel expenses, and total budget.

The dataset is preprocessed using label encoding for categorical features and standard scaling for numerical features. A Gradient Boosting Regressor, wrapped in a MultiOutputRegressor, is trained on the preprocessed data to predict the target variables. The model's performance is evaluated using Mean Absolute Error (MAE) on a test dataset. The trained model is then used to predict values for new travel scenarios provided by users.

The project aims to assist travelers in making informed decisions by providing accurate predictions for various aspects of their travel plans.

PROBLEM STATEMENT

In the travel industry, accurately predicting various expenses and logistical aspects can greatly enhance planning and budgeting processes. This project aims to develop a machine learning model that predicts multiple outputs related to travel based on input features, facilitating informed decision-making for travelers and travel planners.

1. Data Collection and Preprocessing:

Collect a comprehensive dataset containing historical travel data, including features such as starting place, ending place, number of members, fuel type, hotel type, and miscellaneous charges, along with targets such as total distance, travel days, food expenses, travel expenses, and total budget.

Preprocess the dataset by handling missing values, encoding categorical features (e.g., fuel type, hotel type) using LabelEncoder, and scaling numerical features (e.g., members, miscellaneous charges) for optimal model training.

2. Model Selection and Training:

Choose an appropriate machine learning algorithm for multi-output regression, considering factors like the complexity of relationships between features and targets, scalability, and interpretability.

Implement a Gradient Boosting Regressor within a MultiOutputRegressor framework to train the model on the preprocessed dataset, optimizing hyperparameters (e.g., number of estimators, learning rate, max depth) for performance.

3. Model Evaluation:

Split the dataset into training and testing sets to evaluate the model's performance.

Use mean absolute error (MAE) as a metric to assess the model's accuracy in predicting multiple outputs, providing insights into the overall effectiveness of the trained model.

4. Predictive Analysis and Deployment:

Utilize the trained model to make predictions on new data, simulating various travel scenarios to demonstrate its predictive capabilities.

Develop a user-friendly interface or API to allow users to input their travel parameters and receive predicted outputs, empowering them with valuable information for travel planning and budget estimation.

Expected Outcome:

The expected outcome of this project is a reliable and interpretable machine learning model that accurately predicts multiple aspects of travel planning, including distance, duration, expenses, and overall budget. By providing users with actionable insights, the model aims to enhance the efficiency and confidence of travel decision-making processes.

SYSTEM ANALYSIS

Scope:

The project aims to develop a machine learning model for predicting multiple aspects of travel planning, including total distance, travel days, food expenses, travel expenses, and total budget.

The system will handle input parameters such as starting place, ending place, number of members, fuel type, hotel type, and miscellaneous charges.

Users will interact with the system through a user-friendly interface or API to input their travel parameters and receive predicted outputs.

2.1 FUNCTIONAL REQUIREMENTS:

Data Collection: The system must collect and preprocess a dataset containing historical travel data with relevant features and targets.

Data Preprocessing: Handling missing values, encoding categorical features using LabelEncoder, and scaling numerical features for optimal model training.

Model Training: Implementing a Gradient Boosting Regressor within a MultiOutputRegressor framework to train the model on the preprocessed dataset.

Model Evaluation: Evaluating the model's performance using mean absolute error (MAE) on a testing dataset to assess its accuracy in predicting multiple outputs.

Prediction Generation: Using the trained model to make predictions on new data provided by users, simulating various travel scenarios.

User Interface/API: Developing a user-friendly interface or API for users to input their travel parameters and receive predicted outputs.

2.2 NON-FUNCTIONAL REQUIREMENTS:

Performance: The system should be able to train the model efficiently on large datasets and provide accurate predictions within a reasonable time frame.

Scalability: The system should be scalable to handle an increasing number of users and accommodate additional features or targets in the future.

Usability: The user interface or API should be intuitive and easy to use, providing clear instructions for inputting travel parameters and understanding predicted outputs.

Accuracy: The trained model should demonstrate high accuracy in predicting multiple aspects of travel planning, enhancing user confidence in the generated predictions.

- Security: Implementing data encryption and access control measures to ensure the security and privacy of user data.

2.3 SYSTEM ARCHITECTURE:

The system architecture will include components for data collection, preprocessing, model training, evaluation, prediction generation, and user interface/API development.

It may involve cloud-based services for data storage, model training, and deployment to facilitate scalability and accessibility.

Integration with data visualization tools may be considered to present the predicted outputs in a visually appealing and informative manner.

Constraints:

Availability of Quality Data: The system's performance depends on the availability of quality historical travel data with relevant features and targets.

Computational Resources: Adequate computational resources (e.g., processing power, memory) are required for model training and prediction generation.

Risk Analysis:

Data Quality: Inaccurate or incomplete historical travel data may lead to biased or unreliable predictions.

Model Performance: The model's performance may vary based on the complexity of travel scenarios and the diversity of input parameters.

User Adoption: User adoption of the system may depend on factors such as ease of use, accuracy of predictions, and perceived value in travel planning.

2.4 DISADVANTAGES:

1. Data Quality Issues: Inaccurate or incomplete data can lead to biased predictions.

2. Overfitting and Underfitting: Complex models may overfit or underfit the data.

3. Model Interpretability: Complex models like Gradient Boosting Regressors can be less interpretable.

4. Computational Resources: Training complex models requires significant computational resources.

5. Data Privacy and Security: Handling sensitive user data requires robust privacy measures.

6. User Adoption and Trust: Users may be hesitant to trust machine learning predictions.

7. Maintenance and Updates: Models need regular updates and maintenance to stay effective.

2.5 ADVANTAGES:

1. Accurate Predictions: Machine learning models can analyze large datasets and identify complex patterns, leading to more accurate predictions of travel-related aspects such as expenses, duration, and budget.

2. Personalized Recommendations: By considering individual preferences, constraints, and historical data, the model can provide personalized recommendations for accommodations, activities, and transportation options, enhancing the overall travel experience.

3. Efficient Resource Allocation: Optimizing travel itineraries and resource utilization through machine learning algorithms can lead to more efficient allocation of resources such as time, money, and energy during travel.

4. Data-Driven Insights: The project generates valuable data-driven insights into travel patterns, trends, and user behavior, which can be utilized for strategic decision-making and business optimization in the travel industry.

5. Enhanced User Experience: By providing accurate predictions, personalized recommendations, and user-friendly interfaces/APIs, the project aims to enhance the overall user experience in travel planning, leading to higher satisfaction and engagement.

6. Cost Savings: Efficient travel planning and budget estimation can result in cost savings for travelers and travel companies by minimizing unnecessary expenses and optimizing resource utilization.

BACKGROUND AND RELATED WORK

3.1 BACKGROUND:

The travel industry has witnessed a significant transformation with the integration of machine learning and data-driven approaches into travel planning and management. Traditional methods of travel planning often rely on manual estimations and historical trends without leveraging the full potential of data analytics and predictive modeling.

Machine learning techniques offer the capability to analyze large datasets, identify patterns, and make accurate predictions, thereby enhancing the efficiency and accuracy of travel planning processes. By incorporating machine learning into travel planning, stakeholders can optimize itinerary recommendations, estimate budgets more accurately, and provide personalized experiences to travelers.

3.2 RELATED WORK:

1. Predictive Modeling for Travel Expenses:

Several studies have focused on developing predictive models to estimate travel expenses based on various factors such as destination, duration, number of travelers, and travel preferences.

Machine learning algorithms like regression, decision trees, and neural networks have been utilized to predict expenses and budget allocation for different travel scenarios.

2. Optimization of Travel Itineraries:

Researchers have explored optimization techniques and algorithms to generate optimal travel itineraries based on user preferences, constraints, and historical data.

3.Personalized Recommendations in Travel Planning:

Personalization plays a crucial role in enhancing the user experience in travel planning platforms. Recommender systems and collaborative filtering methods have been employed to provide personalized recommendations for accommodations, activities, and transportation options.

Natural language processing (NLP) techniques have also been used to analyze user reviews and preferences, leading to more tailored recommendations.

4. Data Privacy and Security in Travel Data Analysis:

With the growing importance of data privacy and security, researchers have focused on developing privacy-preserving techniques for analyzing travel data.

Differential privacy, federated learning, and secure multi-party computation (SMC) methods have been explored to ensure data confidentiality while deriving meaningful insights from travel datasets.

5. User Trust and Adoption in AI-Based Travel Planning

Understanding user perceptions, trust factors, and adoption barriers in AI-based travel planning systems is a significant area of research.

Studies have investigated user preferences, concerns about data privacy, transparency in AI decision-making, and strategies to build trust and confidence in machine learning models for travel planning.

SOURCE CODE

```
#import Libraries

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.multioutput import MultiOutputRegressor

from sklearn.metrics import mean_absolute_error



# Read data

data = pd.read_csv('/content/drive/MyDrive/submission', encoding='latin-1')


# Categorical features for Encoding labels

categorical_features = ['Fuel_type', 'Hotel_type']


# Encoding labels

label_encoders = {}

for feature in categorical_features:

    label_encoders[feature] = LabelEncoder()

    data[feature] = label_encoders[feature].fit_transform(data[feature])
```

```
features = ['Starting_Place', 'Ending_Place', 'Members', 'Fuel_type', 'Hotel_type', 'Miscellaneous_Charge'] # Define feature
targets = ['Total_Distance', 'Travel_Days', 'Food', 'Travel', 'Total_Budget'] # Define Targets

X = data[features]
y = data[targets]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)  # Split data for Train and Test

scaler = StandardScaler() # Standardize numerical features
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Define model with early stopping
base_model = GradientBoostingRegressor(n_estimators=300, learning_rate=0.1, max_depth=10, random_state=42,validation_fraction=0.2, n_iter_no_change=5, tol=0.001)
model = MultiOutputRegressor(base_model)
model.fit(X_train_scaled, y_train)  # Train model
```

```
y_pred = model.predict(X_test_scaled)  # Evaluate model
mae = mean_absolute_error(y_test, y_pred)
print(f"Mean absolute Error: {mae}")


new_data = pd.DataFrame({
    'Starting_Place': [361],  # Give the Starting_place from '1' to '503'
    'Ending_Place': [396],   # Give the Ending_place from '1' to '503'
    'Members': [5],   # Maximum seven '7' Members limit
    'Fuel_type': ['Petrol'],  # Give the Fuel_type as 'Petrol' or 'Diesel'
    'Hotel_type': ['Mid_range_eats'],   # Give the Hotel_type as 'Cheap_eats' or
'Mid_range_eats' or 'Fine_Dining_eats'
    'Miscellaneous_Charge': [1000]
})
# Encoding labels for new data
for feature in categorical_features:
    new_data[feature] = label_encoders[feature].transform(new_data[feature])
new_data_scaled = scaler.transform(new_data[features])  # Scale new data


predictions = model.predict(new_data_scaled)[0]  # Predict
print("Predicted values:")
for i, target in enumerate(targets):
    print(f"{target}: {predictions[i]}")  # print the predicted value
```

OUTPUT

SAMPLE 1:

Mean absolute Error: 1402.3604415869213

Predicted values:

Total_Distance: 4050.1046218865627

Travel_Days: 4.397766504159627

Food: 11970.816263273895

Travel: 30635.91143550312

Total_Budget: 43661.75834679968

SAMPLE 2:

Mean absolute Error: 1419.1427001322725

Predicted values:

Total_Distance: 4258.979837562508

Travel_Days: 4.712785606193474

Food: 5664.507212329873

Travel: 30867.159030248436

Total_Budget: 40378.52307947679

CONCLUSION

In conclusion, the travel planning project utilizing machine learning techniques presents a promising solution to enhance the efficiency, accuracy, and user experience in travel planning processes. By leveraging predictive modeling, personalized recommendations, and data-driven insights, the project aims to revolutionize how travelers plan their trips and how travel companies optimize their services.

The advantages of the project include accurate predictions, personalized recommendations, efficient resource allocation, cost savings, and a competitive advantage in the travel industry. However, it's essential to address challenges such as data quality, model interpretability, computational resources, data privacy, user trust, and ongoing maintenance to ensure the project's success and long-term sustainability.

Through system analysis, careful model selection and training, rigorous evaluation, and user-centric design of interfaces/APIs, the project strives to deliver tangible benefits to users and stakeholders. By bridging the gap between data analytics and travel planning, the project contributes to advancing the state-of-the-art in AI-driven solutions for the travel industry, ultimately leading to more enjoyable, seamless, and memorable travel experiences for everyone involved.

FUTURE WORK:

1. Feature Engineering: Explore additional features or create new features from existing ones that could improve the model's performance. For example, you could derive features like distance between Starting_Place and Ending_Place, or create interaction features between Members and Hotel_type.

2. Hyperparameter Tuning: Perform hyperparameter tuning to find the best combination of hyperparameters for your Gradient Boosting Regressor. You can use techniques like grid search or random search to efficiently search through the hyperparameter space.

3. Model Evaluation: Besides mean absolute error (MAE), consider using other evaluation metrics such as mean squared error (MSE), root mean squared error (RMSE), or R-squared (R2) score to get a comprehensive understanding of your model's performance.

4. Ensemble Methods: Explore ensemble methods such as blending or stacking to combine predictions from multiple models. This can sometimes lead to improved performance compared to using a single model.

5. Data Augmentation: If possible, consider augmenting your dataset by generating synthetic data points or incorporating external datasets that may provide additional information relevant to your predictions.

6. Regularization Techniques: Experiment with regularization techniques like L1 and L2 regularization to prevent overfitting and improve the generalization of your model.

REFERENCES:

https://pandas.pydata.org/docs

https://numpy.org/doc/

https://scikit-learn.org/stable/documentation.html

https://www.kaggle.com/