**I will produce project that use this dataset for topic "Data-centric product with an AI add-on
" please summary report to one poster template that include at least follwing information i.e. Title, Issue and motivation, Traditional solution, AI add-on solution, Visualization.**

Pro Search

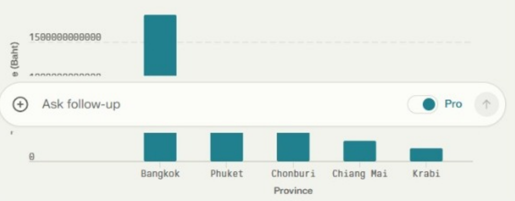
- Analyze the provided dataset to extract relevant information for the poster template including Title, Issue and motivation, Traditional solution, AI add-on solution, and Visualization
- Summarize the findings and create a poster template with the required sections

Sources

thaiiland_domestic_tourism.csv
thailand_domestic_tourism

Answer

Top 5 Provinces by Total Revenue



Poster: Data-Centric Product with AI Add-On for Domestic Tourism in Thailand

Title

Data-Centric Product with AI Add-On for Domestic Tourism in Thailand

Issue and Motivation

The dataset reveals the distribution of domestic and foreign tourists across various provinces in Thailand, along with their revenue contributions. The challenge lies in optimizing tourism strategies to maximize revenue while ensuring tourist satisfaction. Traditional methods of analyzing tourism data are time-consuming and prone to human error, limiting the ability to identify actionable insights efficiently.

Traditional Solution

- Manual analysis of tourism data to identify trends and make decisions.
- Dependence on historical data without predictive capabilities.
- Inefficient marketing strategies due to lack of personalization.

AI Add-On Solution

An AI-powered system can revolutionize tourism management by:

- **Predicting Tourist Trends**: Using machine learning models to forecast tourist inflows based on historical data, seasonality, and external factors.
- **Optimizing Marketing Strategies**: Identifying high-revenue provinces and targeting campaigns accordingly.
- **Personalized Recommendations**: Leveraging AI to provide tailored suggestions for accommodations, attractions, and activities based on tourist preferences.
- **Real-Time Insights**: Offering dashboards that visualize key metrics like revenue, tourist demographics, and satisfaction levels.

Visualization

Below is a bar chart showcasing the **Top 5 Provinces by Total Revenue (Baht)**:

Top 5 Provinces by Total Revenue

Key Insights from Visualization:

1. Bangkok generates the highest total revenue, followed by Phuket, Chonburi, Chiang Mai, and Krabi.
2. These provinces are key targets for enhancing tourism strategies due to their significant economic contributions.

This AI-enhanced product can empower stakeholders in Thailand's tourism sector to make data-driven decisions, improve tourist experiences, and boost overall revenue.

Share Rewrite

Please provide python code for "Predicting Tourist Trends: Using machine learning models to forecast tourist inflows based on historical data, seasonality, and external factors." by using streamlit and seaborn

Pro Search

- Prepare a Python code to predict tourist trends using machine learning models with the provided dataset, incorporating Streamlit for the web app interface and Seaborn for data visualization

Sources

thailand_domestic_tourism.csv
thailand_domestic_tourism

Answer

Below is the Python code for creating a Streamlit app to predict tourist trends using a machine learning model. The app uses historical data, external factors, and seasonality to forecast tourist inflows. It also visualizes results with Seaborn.

```python
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
@st.cache
def load_data():
    url = "https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-
files/21082887/3ff040c2-cd56-40a6-86db-
67592c1cc761/thailand_domestic_tourism.csv"
    data = pd.read_csv(url)
    data['travel_date'] = pd.to_datetime(data['travel_date'])
    data['month'] = data['travel_date'].dt.month
    return data

data = load_data()

# Sidebar for user input
st.sidebar.header("User Input Parameters")
province = st.sidebar.selectbox("Select Province",
data['province_eng'].unique())
month = st.sidebar.slider("Select Month (1-12)", 1, 12, 1)

# Filter data by province
province_data = data[data['province_eng'] == province]

# Prepare features and target variable
X = province_data[['month', 'no_tourist_foreign', 'no_tourist_thai']]
y = province_data['no_tourist_all']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Train Random Forest model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict and evaluate the model
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
st.write(f"Model Mean Squared Error: {mse:.2f}")

# Forecast for user-selected month
forecast_input = pd.DataFrame({'month': [month], 'no_tourist_foreign':
[X['no_tourist_foreign'].mean()],
                               'no_tourist_thai':
[X['no_tourist_thai'].mean()]})
forecast = model.predict(forecast_input)[0]
st.write(f"Predicted Tourist Inflow for {province} in Month {month}:
{forecast:.0f}")

# Visualization: Actual vs Predicted
st.subheader("Actual vs Predicted Tourist Inflows")
fig, ax = plt.subplots()
sns.scatterplot(x=y_test, y=y_pred, ax=ax)
ax.set_xlabel("Actual")
ax.set_ylabel("Predicted")
ax.set_title("Actual vs Predicted Tourist Inflows")
st.pyplot(fig)

# Visualization: Monthly Tourist Trends
st.subheader(f"Monthly Tourist Trends for {province}")
monthly_trends = province_data.groupby('month')
['no_tourist_all'].mean().reset_index()
fig2, ax2 = plt.subplots()
sns.lineplot(data=monthly_trends, x='month', y='no_tourist_all',
marker='o', ax=ax2)
ax2.set_title(f"Monthly Trends for {province}")
ax2.set_xlabel("Month")
ax2.set_ylabel("Average Tourists")
st.pyplot(fig2)
```

Key Features:

1. **Data Loading and Preprocessing**:
    - The dataset is loaded from a CSV file and preprocessed to extract features like `month`.
2. **Model Training**:
    - A `RandomForestRegressor` is trained to predict total tourist inflows based on historical data.
3. **User Interaction**:
    - Users select a province and month to forecast tourist inflows interactively.
4. **Visualizations**:
    - Scatter plot of actual vs predicted values.
    - Line chart showing monthly trends for the selected province.

To run this app:

1. Save the code in a file named `app.py`.
2. Install required libraries: `streamlit`, `pandas`, `numpy`, `scikit-learn`, `seaborn`, `matplotlib`.
3. Run the app using the command: `streamlit run app.py`.

Share Rewrite

Why you should RandomForestRegressor model.

Pro Search

Explain the advantages of using the RandomForestRegressor model for predicting tourist trends

Sources

thailand_domestic_tourism.csv

thailand_domestic_tourism

Answer

RandomForestRegressor is an excellent choice for this predictive modeling task for several reasons:

1. Handling Non-Linear Relationships: Random forests can capture complex, non-linear relationships between features and the target variable, which is likely the case with tourism data that may have intricate patterns.
2. Feature Importance: The model provides feature importance scores, allowing us to understand which factors (e.g., month, foreign tourists, Thai tourists) have the most significant impact on total tourist numbers.
3. Robustness to Outliers: Random forests are less sensitive to outliers compared to linear models, making them suitable for tourism data that may contain extreme values due to events or seasonal fluctuations.
4. Ensemble Learning: As an ensemble method, random forests combine multiple decision trees, reducing overfitting and improving generalization to new data.
5. Handling Mixed Data Types: The model can work well with both numerical and categorical features, which is useful for incorporating various types of tourism-related data.
6. No Assumption of Normality: Unlike some statistical methods, random forests do not assume a normal distribution of the target variable, making them flexible for different data distributions.
7. Parallelization: Random forests can be easily parallelized, allowing for faster training on large datasets, which is beneficial when working with extensive historical tourism data.

By using RandomForestRegressor, we can create a robust predictive model that captures the complexities of tourism patterns and provides valuable insights for forecasting tourist inflows.

Share Rewrite

Related

What makes RandomForestRegressor stand out from other regression models +

How does RandomForestRegressor handle feature interactions +

What are the advantages of using RandomForestRegressor for tourism trend predictions +

How does RandomForestRegressor perform with large datasets +

What are the key hyperparameters to tune in RandomForestRegressor +