Traffic Prediction Application Documentation

# Project Overview

The Traffic Prediction Application is a web-based tool built using Streamlit for analyzing and predicting traffic data. It includes functionalities for loading, preprocessing, and visualizing traffic data, along with interactive features for user customization.

# Features

## Data Loading and Preprocessing:

- Load traffic data from a CSV file (uncleaned\_traffic.csv).  
- Clean and preprocess the data by handling missing values, ensuring numeric columns, removing duplicates, and sorting the data.  
- Extract DateTime features such as year, month, day, hour, and day of the week.  
- Add cyclical features for hour and day of the week.  
- Incorporate a holiday feature based on US federal holidays.  
- Apply feature scaling to the "Vehicles" column using StandardScaler.

## User Interface:

- Sidebar filters for selecting specific date, year, and hour.  
- Display images of traffic police and department logo.  
- Checkbox for filtering by date.  
- Interactive elements for customizing visualizations and settings.

## Visualizations:

- Time series plots and other visual representations of traffic data.  
- Adjustable granularity and color schemes for visualizations.

## Session Management:

- Manage uploaded photos and save the state across sessions.

# Requirements

- Python 3.7+  
- Streamlit  
- Pandas  
- NumPy  
- Scikit-learn  
- Plotly  
- PIL (Pillow)

# Installation

## Clone the Repository:

```bash  
git clone https://github.com/your-repo/traffic-prediction-app.git  
cd traffic-prediction-app  
```

## Install Dependencies:

```bash  
pip install -r requirements.txt  
```

## Run the Application:

```bash  
streamlit run traffic\_prediction\_app.py  
```

# Code Structure

## Data Loading and Preprocessing

```python  
import pandas as pd  
import numpy as np  
from pandas.tseries.holiday import USFederalHolidayCalendar as calendar  
from sklearn.preprocessing import StandardScaler  
import streamlit as st  
  
@st.cache(allow\_output\_mutation=True)  
def load\_data():  
 df = pd.read\_csv("uncleaned\_traffic.csv")  
 df = df.dropna(subset=['DateTime', 'Junction', 'Vehicles', 'ID'])  
 df = df.drop\_duplicates()  
 df['DateTime'] = pd.to\_datetime(df['DateTime'])  
 df['Vehicles'] = pd.to\_numeric(df['Vehicles'], errors='coerce')  
 df = df.dropna(subset=['Vehicles'])  
 df['Junction'] = df['Junction'].astype(int)  
 df['ID'] = df['ID'].apply(lambda x: '{:.0f}'.format(x))  
 df = df.sort\_values(by=['Junction', 'DateTime'])  
 df.set\_index('DateTime', inplace=True)  
 df.fillna(method='ffill', inplace=True)  
 df['Year'] = df.index.year  
 df['Month'] = df.index.month  
 df['Day'] = df.index.day  
 df['Hour'] = df.index.hour  
 df['DayOfWeek'] = df.index.dayofweek  
 df['Hour\_sin'] = np.sin(2 \* np.pi \* df['Hour'] / 24)  
 df['Hour\_cos'] = np.cos(2 \* np.pi \* df['Hour'] / 24)  
 df['DayOfWeek\_sin'] = np.sin(2 \* np.pi \* df['DayOfWeek'] / 7)  
 df['DayOfWeek\_cos'] = np.cos(2 \* np.pi \* df['DayOfWeek'] / 7)  
 cal = calendar()  
 holidays = cal.holidays(start=df.index.min(), end=df.index.max())  
 df['Holiday'] = df.index.isin(holidays).astype(int)  
 scaler = StandardScaler()  
 df['Vehicles\_scaled'] = scaler.fit\_transform(df[['Vehicles']])  
 return df  
```

## Main Application Logic

```python  
import streamlit as st  
from PIL import Image  
import plotly.express as px  
  
def main():  
 st.set\_page\_config(page\_title="Traffic Data Analysis", page\_icon=":car:")  
 df = load\_data()  
 col1, col2, \_ = st.columns([10, 1, 1])  
 with col1:  
 st.image("traffic-police.jpg", width=150)  
 with col2:  
 st.image("logo.jpg", width=150)  
   
 st.sidebar.title("Traffic Data Filters")  
 st.sidebar.title("Traffic Police")  
 traffic\_police\_image = Image.open("traffic-police.jpg")  
 st.sidebar.image(traffic\_police\_image, caption='Traffic Police')  
 department\_logo\_image = Image.open("logo.jpg")  
 st.sidebar.image(department\_logo\_image, caption='Department Logo')  
   
 show\_date\_checkbox = st.sidebar.checkbox("Filter by Date")  
 if show\_date\_checkbox:  
 st.sidebar.title("Date Filters")  
 year = st.sidebar.selectbox("Select Year", df['Year'].unique())  
 # Additional filtering logic here...  
  
 # Example visualization  
 st.title("Traffic Data Analysis")  
 fig = px.line(df, x=df.index, y='Vehicles\_scaled')  
 st.plotly\_chart(fig)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()  
```

# Usage

## Launch the Application:

```bash  
streamlit run traffic\_prediction\_app.py  
```

## Interact with the Interface:

- Use the sidebar to filter data by date, year, and hour.  
- View and download uploaded photos.  
- Customize visualization settings.

## Analyze Traffic Data:

- Explore traffic patterns through interactive visualizations.  
- Adjust granularity and color schemes to suit your analysis needs.

# Conclusion

This documentation covers the essential aspects of the Traffic Prediction Application. The application allows users to analyze traffic data interactively, providing useful insights through visualizations and customizable settings.