

Documentation:

● API Documentation using Swagger/OpenAPI

- Implemented automatically via **FastAPI**.
- Accessible at:
 - Swagger UI: <http://127.0.0.1:8000/docs>
 - ReDoc: <http://127.0.0.1:8000/redoc>
- All API endpoints (/predict) are documented with input schema (PredictionInput) and output schema (PredictionOutput), including data types and example usage.

● Model Architecture and Training Documentation:

- **Model:** Decision Tree Classifier from sklearn.tree.
- **Features Used:**
 - team1, team2, venue, venue_team (categorical)
 - recent_wins, recent_matches (numerical)
- **Label Encoding** applied to all categorical features.
- **Training:**
 - Conducted in main.py or optionally in a separate notebook model_training.ipynb.
 - Model is saved as model.pkl after training.
- **Sample Data:** Manually created dataset with realistic IPL match scenarios.
- **Output:** Prediction includes winning team, mock score, and reasoning string.

● Data Processing Pipeline Documentation:

1. **Input Collection:** User provides team names, venue, and recent performance.
2. **Encoding:** Categorical inputs are converted to numeric values using LabelEncoder.
3. **Model Prediction:**
 - Encoded inputs are passed to the trained decision tree model.
 - Output is decoded to return a team name as the predicted winner.
4. **Mock Score Generation:**
 - Based on team performance ratio and fixed logic (e.g., 180 vs 160).
5. **Output Format:** JSON response with winner, score, and reasoning.

● Setup and Installation Instructions:

Backend (FastAPI)

```
git clone https://github.com/SandhitaG/IPL-Prediction-Model.git
```

```
cd IPL Prediction Model
```

```
# Install Python dependencies
```

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

```
# Start FastAPI server
```

```
python -m uvicorn api_fastapi.main:app --reload
```

```
# Access API docs
```

```
# Swagger: http://127.0.0.1:8000/docs
```

```
# ReDoc: http://127.0.0.1:8000/redoc
```

Frontend (React)

```
cd frontend
```

```
npm install
```

```
npm start
```