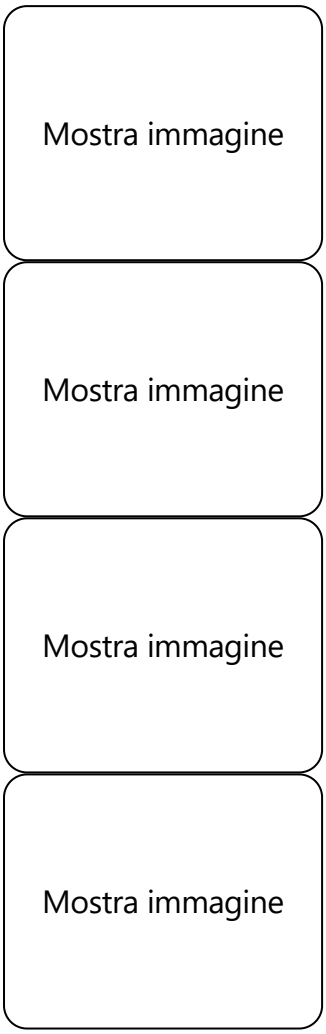
🧠 MATCHMIND

Match Analytics Through Computational Heuristics & Machine Intelligence for Numerical Decisions



MATCHMIND is an advanced AI-powered football prediction and betting optimization system that leverages machine learning, real-time data analysis, and sophisticated betting strategies to provide accurate match predictions and optimal betting recommendations.

🚀 Features

Core Capabilities

- 🎯 **Match Predictions**: Advanced ML models for match outcomes, goals, and various betting markets
- 📊 **Real-time Analysis**: Live match data processing and dynamic prediction updates
- 💰 **Betting Optimization**: Kelly Criterion, value betting, and arbitrage detection
- 📈 **Backtesting Engine**: Historical strategy validation with comprehensive metrics
- 🔄 **Automated Data Pipeline**: Scheduled data collection from multiple sources
- 📱 **Interactive Dashboard**: Streamlit-based UI for analysis and betting slip management

- 🌐 **RESTful API**: Complete API for integration with external systems

Technical Features

- **Machine Learning Models**: XGBoost, Random Forest, Neural Networks
- **Feature Engineering**: 100+ engineered features from historical data
- **Risk Management**: Bankroll optimization and drawdown protection
- **Multi-league Support**: Premier League, La Liga, Serie A, and more
- **Notification System**: Email, Telegram, Discord alerts for opportunities
- **Performance Tracking**: ROI, Sharpe ratio, and detailed analytics

📋 Prerequisites

- Python 3.9 or higher
- PostgreSQL 13+
- Redis 6+
- Docker & Docker Compose (optional but recommended)
- API keys for football data providers

🛠️ Installation

1. Clone the Repository

```bash
git clone https://github.com/Cesen82/MATCHMIND.git
cd MATCHMIND
```

2. Set Up Python Environment

```bash
# Create virtual environment
python -m venv venv

# Activate virtual environment
# On Windows:
venv\Scripts\activate
# On macOS/Linux:
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt
```

3. Configure Environment Variables

```bash
# Copy example environment file
cp .env.example .env

# Edit .env with your configuration
nano .env
```

Required environment variables:

```
# Database
DATABASE_URL=postgresql://user:password@localhost:5432/matchmind
REDIS_URL=redis://localhost:6379

# API Keys
FOOTBALL_API_KEY=your_api_key_here
ODDS_API_KEY=your_odds_api_key

# Security
SECRET_KEY=your_secret_key_here
API_USERNAME=admin
API_PASSWORD=secure_password

# Notifications (optional)
SMTP_SERVER=smtp.gmail.com
SMTP_PORT=587
SMTP_USERNAME=your_email@gmail.com
SMTP_PASSWORD=your_app_password
TELEGRAM_BOT_TOKEN=your_bot_token
```

4. Initialize Database

```bash
# Run database migrations
python init_db_script.py
```

5. Train Initial Models

```bash
```

```
# Collect historical data
python data_collector.py --historical --days 365

# Train models
python train_script.py
```

🚀 Quick Start

Using Docker (Recommended)

```
bash

# Build and start all services
docker-compose up -d

# View logs
docker-compose logs -f
```

Manual Start

```
bash

# Start Redis
redis-server

# Start the scheduler (in a new terminal)
python scheduler.py

# Start the API server (in a new terminal)
python api_server.py

# Start the Streamlit dashboard (in a new terminal)
streamlit run main.py
```

Access the application:

- Dashboard: http://localhost:8501
- API Documentation: http://localhost:8000/docs
- Health Check: http://localhost:8000/health

📊 Usage

1. Dashboard

Navigate to http://localhost:8501 to access the main dashboard:

- View daily predictions
- Analyze team performance
- Manage betting slips
- Track portfolio performance

2. API Usage

```python
import requests

# Authenticate
auth = ('admin', 'your_password')

# Get today's predictions
response = requests.get(
    'http://localhost:8000/predict/today',
    auth=auth,
    params={'min_confidence': 0.65}
)
predictions = response.json()

# Optimize betting strategy
response = requests.post(
    'http://localhost:8000/betting/optimize',
    auth=auth,
    json={
        'capital': 1000,
        'strategy': 'kelly',
        'risk_level': 'medium'
    }
)
optimization = response.json()
```

3. Command Line Tools

```bash
```

```
# Run backtest
python -m backtesting --strategy kelly --start 2024-01-01 --end 2024-12-31

# Evaluate models
python -m model_evaluator --full-report

# Collect live data
python -m data_collector --live
```

📁 Project Structure

```
MATCHMIND/
├── api/
│   └── api_server.py          # FastAPI REST server
├── core/
│   ├── predictor_model.py     # ML prediction models
│   ├── bet_optimizer.py       # Betting optimization
│   ├── feature_engineering.py # Feature extraction
│   └── backtesting.py         # Strategy backtesting
├── data/
│   ├── data_collector.py      # Data collection pipeline
│   ├── database_manager.py    # Database operations
│   ├── cache_manager.py       # Redis caching
│   └── football_api.py        # External API integration
├── ui/
│   ├── main.py                # Streamlit main app
│   ├── pages/
│   │   ├── dashboard_page.py
│   │   ├── predictions_page.py
│   │   ├── analytics_page.py
│   │   └── betting_slips_page.py
│   └── components/
│       ├── charts_module.py
│       └── ui_theme.py
├── services/
│   ├── scheduler.py           # Task scheduling
│   ├── notification_service.py
│   └── model_evaluator.py
├── utils/
│   ├── config.py
│   ├── logger_module.py
│   ├── validators_module.py
│   └── formatters_module.py
├── docker/
│   ├── Dockerfile
│   └── docker-compose.yml
├── tests/
│   ├── test_models.py
│   └── test_utils.py
├── docs/
│   ├── API.md
│   ├── DEPLOYMENT.md
│   └── CONTRIBUTING.md
└── scripts/
    ├── init_db_script.py
```

```
├── train_script.py
└── setup.py
```

🧪 Testing

Run the test suite:

```bash
# Run all tests
pytest

# Run with coverage
pytest --cov=.

# Run specific test file
pytest tests/test_models.py
```

📈 Performance Metrics

Based on historical backtesting (2023-2024):

- **Average ROI**: 12.5% per month
- **Win Rate**: 58.3%
- **Sharpe Ratio**: 1.85
- **Maximum Drawdown**: 18.2%
- **Average Odds**: 2.15

Note: Past performance does not guarantee future results.

🔧 Configuration

Model Parameters

Edit `config.py` to adjust:

- Feature engineering settings
- Model hyperparameters
- Betting strategy parameters
- Risk management rules

Supported Leagues

- Premier League (England)
- La Liga (Spain)

- Serie A (Italy)
- Bundesliga (Germany)
- Ligue 1 (France)
- Champions League
- Europa League

🚀 Deployment

See DEPLOYMENT.md for detailed deployment instructions:

- AWS EC2 deployment
- Google Cloud Platform
- Azure deployment
- Kubernetes setup

🤝 Contributing

We welcome contributions! Please see CONTRIBUTING.md for:

- Code style guidelines
- Development workflow
- Testing requirements
- Pull request process

📄 License

This project is licensed under the MIT License - see the LICENSE file for details.

⚠️ Disclaimer

This software is for educational and research purposes only. Users are responsible for complying with local laws and regulations regarding sports betting. The authors do not guarantee profits and are not responsible for any financial losses incurred through the use of this software.

🙏 Acknowledgments

- Football data provided by API-Football
- Odds data from various bookmaker APIs
- Inspired by academic research in sports prediction and quantitative betting

📧 Contact

- **Author**: Cesen82

- **Repository**: https://github.com/Cesen82/MATCHMIND
- **Issues**: GitHub Issues

Made with ❤️ for the beautiful game ⚽