Artizence Systems LLP
LLP NO: ACH 3954
Regd office :- 421 sector-10C, Vrindavan Colony ,
Lucknow, India 226029

Technical Assessment for Artizence System LLP

Position: Machine Learning Engineer / Django Developer / Full Stack AI Engineer

Candidate Requirements By Role

- AI Engineer candidates should implement Task 1 with both Django and FastAPI integration
- Backend Developer candidates should focus on completing Task 2 and Task 3
- All candidates are encouraged to attempt all three tasks for the highest chance of selection
- Backend experience is considered a significant advantage regardless of role

Project Overview

Develop a comprehensive IPL cricket prediction system that leverages machine learning to forecast match outcomes, winning team scores, losing team scores, and player-wise performance metrics. The system will incorporate reasoning capabilities and provide an interactive chatbot interface for users to gain insights into predictions and match analysis.

Data Sources

- Player statistics (individual performance metrics across matches)
- Stadium information (location, pitch conditions, weather factors)
- Team information (strength, roster, current form)
- Live match data (real-time statistics and updates)
- Historical match data (past performances and trends)
- Players' recent performance in the last 3-5 matches
- Team-wise win records and patterns

Tasks

Task 1: AI Model Development

Primary focus for AI Engineer candidates - must implement with both Django and FastAPI

1. Model Architecture
    - Design and implement an ensemble machine learning model for predicting IPL match outcomes
    - Incorporate various algorithms (e.g., gradient boosting, neural networks, etc.)
    - Create feature engineering pipelines to transform raw cricket statistics
    - Implement time-series components to account for team momentum and trends
    - Develop specific predictors for:
        - Match winner
        - Winning team's final score
        - Losing team's final score
        - Key player performance metrics (runs, wickets, economy rate, etc.)
        - Performance trends based on recent matches (last 3-5 games)
2. LLM Reasoning Integration
    - Integrate a local large language model (preferably using Ollama) to provide reasoning
    - Develop a system for the LLM to explain predictions and highlight key factors
    - Design prompt engineering methods to extract optimal insights from the LLM
    - Create a hybrid approach that combines statistical ML with LLM reasoning
3. Model Training & Evaluation
    - Train multiple model variations using historical IPL data
    - Implement cross-validation with appropriate metrics (accuracy, F1, etc.)
    - Create a robust evaluation framework for continuous performance monitoring
    - Implement techniques to address class imbalance issues
    - Develop specialized evaluation metrics for player performance predictions
4. Prediction System Design
    - Develop a system to generate both point predictions and confidence intervals

- Implement methods to account for unusual circumstances (injuries, weather, etc.)
- Create a mechanism for updating predictions with live match data
- Design a pipeline for model retraining as new data becomes available
- Create visualization tools for displaying player performance trends over recent matches

Task 2: Django Backend Development

Primary focus for Django Developer candidates

1. API Development
    - Create RESTful API endpoints using Django REST Framework
    - Implement authentication and authorization mechanisms
    - Design efficient database models for storing basketball statistics
    - Develop endpoints for model predictions and explanations
2. Database Design
    - Create models for teams, players, matches, and predictions
    - Implement efficient indexing for quick data retrieval
    - Design schema to handle time-series data effectively
    - Create relationships between different data entities
    - Design specialized structures for storing player performance across recent matches
3. Integration with ML Model
    - Develop a system to serve ML model predictions via API
    - Implement caching mechanisms for frequently requested predictions
    - Create a pipeline for model updates and versioning
    - Design error handling for ML model failures
4. API Documentation
    - Create comprehensive documentation for all API endpoints
    - Implement Swagger/OpenAPI specification
    - Develop usage examples for common scenarios
    - Design error codes and messages

Task 3: Data Collection Pipeline

Additional task for Full Stack AI Engineer candidates

1. Web Scraper Development
    - Implement Selenium-based scrapers for IPL cricket websites
    - Design scraping schedules based on match calendars
    - Create robust error handling for scraper failures
    - Develop proxy rotation and user-agent switching mechanisms
    - Create specific scrapers for player-wise performance metrics and recent match data
2. Data Processing

- Create data cleaning and normalization pipelines
- Implement data validation checks
- Design transformations to convert raw data into model features
- Create data versioning system

3. Storage Integration
    - Develop methods to store scraped data in Django models
    - Implement batch processing for efficient database operations
    - Create indexing strategies for optimal query performance
    - Design data archiving and retention policies
4. Monitoring System
    - Create dashboards for monitoring scraper performance
    - Implement alerts for scraper failures
    - Design logging systems for data quality issues
    - Develop metrics for data freshness and completeness

Bonus: Frontend Development

Added value for all positions

1. React.js Interface
    - Design a responsive frontend for interacting with predictions
    - Create visualizations for match predictions and analysis
    - Implement real-time updates for live matches
    - Design an intuitive chatbot interface for user interactions
    - Develop specialized visualizations for player performance trends across recent matches

Technical Requirements

1. Code Quality
    - Well-structured, documented, and tested code
    - Adherence to PEP 8 standards for Python code
    - Comprehensive unit tests with good coverage
    - Clean, modular architecture
2. Performance
    - Efficient database queries
    - Optimized ML model inference
    - Scalable architecture for handling peak loads
    - Responsive API endpoints
3. Documentation
    - Thorough API documentation
    - Clear installation and setup instructions
    - Comprehensive README for project overview
    - Inline code comments for complex logic

4. Deployment
    - Docker containerization for all components
    - Environment-specific configuration options
    - CI/CD pipeline setup
    - Monitoring and logging infrastructure

Submission Requirements

1. Repository Structure
    - Create a private GitHub repository
    - Add GitHub user akshat0098 as a collaborator
    - Organize code into logical modules and packages
    - Include a comprehensive README.md file
2. Documentation
    - API documentation using Swagger/OpenAPI
    - Model architecture and training documentation
    - Data processing pipeline documentation
    - Setup and installation instructions
3. Testing
    - Unit tests for key components
    - Integration tests for system interactions
    - Performance benchmarks
    - Test coverage reports
4. Submission Process
    - Send a formal email to akshat@artizence.com with:
        - Subject line: "IPL Prediction Model - Technical Assessment Submission"
        - Brief introduction and summary of your implementation
        - GitHub repository link (ensure akshat0098 has been added as collaborator)
        - Any special instructions or features you'd like to highlight
        - Your contact information and availability for follow-up discussions

Evaluation Criteria

1. Technical Implementation
    - Quality and efficiency of ML model
    - Structure and performance of Django/FastAPI backend
    - Robustness of data collection pipeline
    - Integration between components

    - Breadth of tasks completed (candidates who complete more tasks will receive higher consideration)
2. Code Quality
    - Clean, readable, and maintainable code
    - Proper error handling
    - Adherence to best practices
    - Comprehensive documentation
3. System Design
    - Scalability of architecture
    - Efficiency of data processing
    - Thoughtfulness of database design
    - Integration of components
4. Innovation
    - Creative approach to prediction challenges
    - Novel use of LLM reasoning capabilities
    - Unique features beyond basic requirements
    - Elegant solutions to technical problems

Preferred Technologies

- Machine Learning: Python, TensorFlow/PyTorch, scikit-learn
- LLM Integration: Ollama (local model hosting)
- Backend: Django, Django REST Framework, FastAPI (AI Engineer candidates must implement both)
- Database: PostgreSQL
- Web Scraping: Selenium, BeautifulSoup
- Frontend: React.js (bonus)
- Deployment: Docker, GitHub Actions

Timeline

- 1 week from receipt of assessment
- Progressive commits throughout the development period are encouraged

Contact

For any questions or clarifications regarding this assessment, please contact the hiring team at Artizence System LLP.

---

*Note: This technical assessment is designed to evaluate your skills in developing a complete AI system. For highest chances of selection:

- AI Engineer candidates should complete Task 1 with both Django and FastAPI implementations, and attempt Tasks 2 & 3
- Backend Developer candidates should focus on completing Tasks 2 & 3 thoroughly
- Candidates who successfully complete all three tasks will be given strong preference in the selection process
- Backend development experience (Django/FastAPI) is considered a significant advantage for all positions*

Regards,

Akshat Srivastava,
Designated Partner ,
Artizence Systems LLP
+91-9369489827