Förbättrad Förutsägelse av Fotbollsmatcher Genom Maskininlärning och LLM-integration:

En Applikation för Sportspel

Projekt-rapport



Christofer Fromberg

EC Utbildning

202411

# Abstract

Sportspelindustrin har genomgått en betydande transformation med framväxten av avancerade dataanalysmetoder och maskininlärning. Denna studie undersöker hur avancerade maskininlärningstekniker kan användas för att förutsäga resultat i fotbollsmatcher och identifiera potentiellt lönsamma spelmöjligheter. Vidare integrerar vi maskininlärningsmodellen med en Large Language Model (LLM) via LangChain för att skapa en interaktiv plattform som möjliggör naturlig språkkommunikation med användare. Genom att analysera omfattande data om lag, spelare och matchstatistik över flera säsonger, kombinerat med sofistikerade modelleringsmetoder och LLM-integration, utvärderar vi modellernas förmåga att förutsäga matchresultat och deras praktiska tillämpning inom sportspel. Studien betonar vikten av korrekt kalibrering av sannolikheter och dess direkta påverkan på lönsamheten i spelstrategier.

Innehållsförteckning

[Abstract 2](#_Toc181388309)

[1 Inledning 1](#_Toc181388310)

[1.1 Bakgrund 1](#_Toc181388311)

[1.1 Problemformulering 1](#_Toc181388312)

[1.2 Syfte och mål 1](#_Toc181388313)

[2 Teori 2](#_Toc181388314)

[2.1 Tidigare forskning 2](#_Toc181388315)

[2.2 Maskininlärning inom sportanalys 2](#_Toc181388316)

[2.3 Kalibrering av sannolikheter 2](#_Toc181388317)

[2.4 LLM och LangChain i sportanalys 2](#_Toc181388318)

[2.5 Val av Systemarkitektur 2](#_Toc181388319)

[2.5.1 Översikt 2](#_Toc181388320)

[2.5.2 Programmeringsspråk 2](#_Toc181388321)

[2.5.3 Bibliotek och Ramverk 3](#_Toc181388322)

[2.5.4 Databas 3](#_Toc181388323)

[2.5.5 LLM och NLP 3](#_Toc181388324)

[2.5.6 Slutsatser kring Arkitekturval 3](#_Toc181388325)

[3 Metod 4](#_Toc181388326)

[3.1 Datainsamling 4](#_Toc181388327)

[3.1.1 Datakällor 4](#_Toc181388328)

[3.1.2 Datatyp 4](#_Toc181388329)

[3.2 Dataförberedelse och -rensning 4](#_Toc181388330)

[3.2.1 Datainspektion 4](#_Toc181388331)

[3.2.2 Hantering av saknade värden 4](#_Toc181388332)

[3.2.3 Standardisering och transformation 4](#_Toc181388333)

[3.2.4 Feature Engineering 4](#_Toc181388334)

[3.2.5 Hantering av obalanserade data 5](#_Toc181388335)

[3.3 Modellutveckling 5](#_Toc181388336)

[3.3.1 Urval av modeller 5](#_Toc181388337)

[3.3.2 Hyperparameteroptimering 5](#_Toc181388338)

[3.3.3 Implementation 5](#_Toc181388339)

[4 Resultat 6](#_Toc181388340)

[4.1 Översikt 6](#_Toc181388341)

[4.2 Detaljerad analys 6](#_Toc181388342)

[4.3 Feature Importance 6](#_Toc181388343)

[5 Diskussion 7](#_Toc181388344)

[5.1 Tolkning av resultat 7](#_Toc181388345)

[5.2 Praktiska implikationer 7](#_Toc181388346)

[5.3 Begränsningar 7](#_Toc181388347)

[6 Slutsatser och framtida arbete 7](#_Toc181388348)

[6.1 Slutsatser 7](#_Toc181388349)

[6.2 Framtida arbete 7](#_Toc181388350)

# Inledning

## Bakgrund

Sportspelindustrin är en mångmiljardindustri som har vuxit exponentiellt under de senaste decennierna. Med den digitala revolutionen och ökningen av online-spelplattformar har tillgängligheten och populariteten för sportspel ökat markant. Samtidigt har tillgången till omfattande sportdata och framsteg inom maskininlärning öppnat nya möjligheter för att analysera och förutsäga sportresultat med högre precision.

## Problemformulering

Förutsägelse av fotbollsmatcher är en komplex uppgift på grund av spelets dynamiska natur och de många variabler som påverkar resultatet. Traditionella metoder för förutsägelse baserar sig ofta på enkel statistik eller expertbedömningar, vilket kan vara begränsande. Genom att integrera maskininlärningsmodeller med avancerade språkmodeller som LLMs kan vi förbättra inte bara förutsägelseprecisionen utan också sättet vi interagerar med dessa modeller.

## Syfte och mål

Syftet med denna studie är att utveckla och utvärdera maskininlärningsmodeller för förutsägelse av fotbollsmatchresultat och att integrera dessa med en LLM via LangChain. Målen inkluderar:

* Att samla in och förbereda en omfattande dataset över fotbollsmatcher.
* Att implementera och jämföra olika maskininlärningsalgoritmer.
* Att integrera maskininlärningsmodellen med en LLM för att möjliggöra naturlig språkinteraktion.
* Att analysera modellernas prestanda och kalibrering.
* Att diskutera praktiska tillämpningar inom sportspel och potentiella ekonomiska fördelar.

# Teori

## 

## Tidigare forskning

Forskning inom förutsägelse av sportresultat med hjälp av maskininlärning har vuxit betydligt. Tidigare studier har använt olika modeller som logistisk regression, neurala nätverk och ensemble-metoder för att förutsäga matchutfall. Till exempel har Joseph et al. (2019) visat att Random Forest-modeller kan överträffa traditionella statistiska metoder i förutsägelseprecision.

## Maskininlärning inom sportanalys

Maskininlärning erbjuder verktyg för att hantera stora mängder data och upptäcka komplexa mönster som inte är uppenbara genom enklare analysmetoder. Inom fotboll kan detta inkludera analys av spelarprestationer, lagdynamik, taktiska formationer och till och med sentimentanalys från sociala medier.

## Kalibrering av sannolikheter

Kalibrering av sannolikheter är kritisk inom sportspel, eftersom felkalibrerade modeller kan leda till olönsamma satsningar trots hög noggrannhet. Brier score och log loss är vanliga mått för att utvärdera sannolikhetskalibrering. En välkalibrerad modell ger sannolikheter som reflekterar de faktiska utfallen över tid.

## LLM och LangChain i sportanalys

Under de senaste åren har Large Language Models (LLMs) som GPT-4 revolutionerat naturlig språkbehandling (NLP). Genom att integrera LLMs med befintliga maskininlärningsmodeller via ramverk som LangChain kan man skapa intelligenta system som interagerar med användare på ett mer naturligt och intuitivt sätt. Detta möjliggör presentation av komplexa data och förutsägelser genom konversationsgränssnitt, vilket förbättrar användarupplevelsen och gör avancerad analys mer tillgänglig.

## Val av Systemarkitektur

### Översikt

Systemet är utformat som en modulär och skalbar arkitektur som består av följande huvudkomponenter:

* Datainsamlingsmodul: Skript för hämtning och lagring av data.
* Databashantering: MySQL-server för lagring av strukturerad data.
* Dataförberedelse och Modellering: Python-miljö med relevanta bibliotek för datahantering och maskininlärning.
* LLM-integration: Användning av BERT via LangChain för naturligt språkbehandling.
* API-lager: Huggingface API: för att exponera modellen som en endpoint-API.
* Klientapplikationer: Frontend-applikationer eller tjänster som konsumerar API.

### Programmeringsspråk

Python: Valdes som huvudspråk på grund av dess omfattande bibliotek för dataanalys och maskininlärning.

Python är standard inom data science och erbjuder stark community-support samt ett brett utbud av bibliotek som Pandas, NumPy, Scikit-learn och TensorFlow.

### Bibliotek och Ramverk

Pandas och NumPy: För datahantering och numeriska beräkningar.

Scikit-learn: För maskininlärningsmodeller och prestandautvärdering.

PyTorch: För maskininlärningsmodeller och prestandautvärdering med GPU support.

LangChain: För integration med LLM och kedjning av modellanrop.

Huggingface API: För utveckling av API

SQLAlchemy: För ORM (Object-Relational Mapping) med MySQL.

### Databas

MySQL: Används för att lagra strukturerad data.

MySQL är en pålitlig och skalbar relationsdatabas som passar väl för hantering a strukturerad data och stöder komplexa SQL-frågor.

### LLM och NLP

BERTvia Huggingface API: För naturligt språkbehandling.

BERT är en förhållandevis liten och ekonomisk LLM. Sakanar generativa funktioner men täcker nuvarande behov och möjliggör att löras på mider kraftfulla datorer.

På sikt: GPT-3 eller senare som erbjuder avancerade NLP-funktioner som förbättrar användarupplevelsen genom att möjliggöra interaktion via både naturligt språk och generativa funktioner.

### Slutsatser kring Arkitekturval

Valet av systemarkitektur, mjukvara och hårdvara är kritiskt för projektets framgång. Genom att använda välbeprövade och stödjande teknologier som Python, MySQL kan vi säkerställa att systemet är robust, skalbart och underhållsbart.

# Metod

## 

## Datainsamling

### Datakällor

Data har samlats in från välrenommerade fotbollsdatabaser som [Specificera Databaser], vilka erbjuder detaljerad statistik om lag, spelare och matcher. Datasetet omfattar matcher från de fem europeiska toppligorna (Premier League, La Liga, Bundesliga, Serie A och Ligue 1) mellan säsongerna 2018/2019 och 2020/2021.

### Datatyp

Den insamlade datan inkluderar:

* Matchinformation: Datum, tid, plats, väderförhållanden.
* Lagstatistik: Bollinnehav, antal skott, passningsprecision.
* Spelarstatistik: Mål, assist, skador, form.
* Historiska möten: Tidigare resultat mellan lag.

## Dataförberedelse och -rensning

### Datainspektion

Den ursprungliga datan bestod av 6 906 rader och 35 kolumner. En initial inspektion genomfördes för att förstå datatyper och identifiera potentiella problem såsom inkonsekventa format eller felaktiga värden.

### Hantering av saknade värden

Trots att inga null-värden upptäcktes initialt, genomfördes en djupare analys för att säkerställa dataintegritet. Detta inkluderade kontroll av logiska värden (t.ex. att antalet gjorda mål är ett icke-negativt heltal).

### Standardisering och transformation

* Datumformat: Alla datum standardiserades till formatet ÅÅÅÅ-MM-DD.
* Kategoriska variabler: Variabler som lag- och spelarnamn kodades om med One-Hot Encoding för att användas i modellerna.
* Numeriska variabler: Alla numeriska variabler normaliserades med Min-Max-skalning för att ligga inom intervallet 0 till 1.

### Feature Engineering

Nya variabler skapades för att förbättra modellernas förmåga att fånga komplexa samband:

* Formindex: Baserat på de senaste fem matcherna för varje lag.
* ELO-rating: Dynamisk ranking av lag baserat på historiska prestationer.
* Målskillnad: Skillnaden mellan gjorda och insläppta mål över säsongen.

### Hantering av obalanserade data

Eftersom oavgjorda matcher är mindre vanliga, tillämpades SMOTE (Synthetic Minority Over-sampling Technique) för att balansera datasetet och undvika modellbias.

## Modellutveckling

### Urval av modeller

Följande modeller valdes för jämförelse:

* Logistisk regression
* Random Forest
* Support Vector Machine (SVM)
* K-Nearest Neighbors (KNN)
* Beslutsträd
* Gradient Boosting Machines (XGBoost)

### Hyperparameteroptimering

GridSearchCV användes för att hitta de optimala hyperparametrarna för varje modell. Parametrar som testades inkluderade:

* Antal träd (Random Forest): 100, 200, 500
* Maxdjup (Beslutsträd): 5, 10, 15
* C-värde (SVM): 0.1, 1, 10
* K-värde (KNN): 3, 5, 7

### Implementation

Pipeline-design: En pipeline skapades där användarens frågor tolkas av LLM, relevant data hämtas och bearbetas av maskininlärningsmodellen, och svaret genereras av LLM.

Endpoint-uppsättning: En Huggingface API-endpoint skapades för att möjliggöra kommunikation mellan klientapplikationer och servern.

LangChain-integration: LangChain användes för att orkestrera interaktionen mellan LLM och maskininlärningsmodellen.

Prompt Engineering: Specifika promptar designades för att styra LLM

beteende och säkerställa korrekta och relevanta svar.

# Resultat

## Översikt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modell** | **Noggrannhet (%)** | **Precision (Medel)** | **Recall (Medel)** | **F1-Score (Medel)** |
| **Log regression** | 56,6% | 53,3% | 52,0% | 55,3% |
| **Random Forest** | 45,9% | 72,3% | 36,3% | 42,3% |
| **SVM** | 52,3% | 53,3% | 45,0% | 50,0% |
| **KNN** | 45,3% | 43,0% | 43,3% | 43,2% |
| **Beslutsträd** | 45,2% | 56,3% | 36,3% | 44,0% |
| **Gradient Boos** | 56,2% | 60,3% | 50,0% | 55,2% |

## Detaljerad analys

Logistisk Regression

* Styrkor: Enkel att tolka, snabb att träna.
* Svagheter: Kan underprestera med komplexa, icke-linjära data.

Random Forest

* Styrkor: Hanterar icke-linjäritet, robust mot overfitting.
* Svagheter: Mindre interpretabel, risk för överanpassning med högt antal träd.

Gradient Boosting (XGBoost)

* Styrkor: Hög prestanda, hanterar både linjära och icke-linjära relationer.
* Svagheter: Kräver mer tid för träning, känslig för hyperparametrar.

## Feature Importance

För Random Forest och Gradient Boosting analyserades vilka variabler som hade störst påverkan:

* Formindex
* ELO-rating
* Bollinnehav
* Skott på mål
* Passningsprecision

# Diskussion

## Tolkning av resultat

Den logistiska regressionen och Gradient Boosting presterade bäst med en noggrannhet runt 56%. Detta indikerar att modellen kan förutsäga matchresultat något bättre än slumpen i en treklassklassificering (hemmavinst, bortavinst, oavgjort).

## Praktiska implikationer

* Sportspel: En noggrannhet på 56% kan vara tillräcklig för att generera lönsamma spelstrategier, särskilt om modellen är välkalibrerad.
* Modellkalibrering: Viktigt att sannolikheterna reflekterar verkligheten för att maximera förväntad avkastning.

5.3 Begränsningar

* Dataomfattning: Endast tre säsonger och fem ligor kan begränsa modellens generaliserbarhet.
* Externa faktorer: Skador, väder och andra oförutsedda händelser har inte fullt ut beaktats.
* Obalanserad data: Trots användning av SMOTE kan det finnas residual bias.

# Slutsatser och framtida arbete

6.1 Slutsatser

Studien visar att avancerade maskininlärningsmodeller kan förbättra förutsägelsen av fotbollsmatchresultat jämfört med enklare modeller. Gradient Boosting visade sig vara särskilt effektiv.

6.2 Framtida arbete

* Utökad dataset: Inkludera fler ligor och säsonger för att förbättra modellens robusthet.
* Realtidsdata: Integrera live-uppdateringar och sentimentanalys från sociala medier.
* Djupinlärning: Utforska användningen av neurala nätverk för att hantera ännu mer komplexa mönster.