Análisis CRISP-DM: NBA Machine Learning Sports Betting

Repositorio: https://github.com/kyleskom/NBA-Machine-Learning-Sports-Betting

1. Comprensión del Negocio

Contexto

El proyecto se enfoca en la aplicación de machine learning para predecir resultados de apuestas deportivas en la NBA, específicamente para determinar si apostar por encima o por debajo del spread establecido por las casas de apuestas.

Objetivos del Negocio

- Desarrollar un sistema predictivo que genere un retorno de inversión positivo en apuestas deportivas de la NBA.
- Identificar patrones no evidentes para el apostador promedio que puedan proporcionar una ventaja competitiva.
- Automatizar el proceso de recopilación de datos, análisis y predicción para facilitar la toma de decisiones.

Criterios de Éxito

- Precisión predictiva superior al 55-60% (umbral mínimo para obtener rentabilidad en apuestas deportivas).
- Retorno de inversión (ROI) positivo a largo plazo.
- Capacidad para actualizar y ajustar modelos con nuevos datos de temporada.

Evaluación de la Situación

- **Recursos disponibles**: Datos públicos de partidos de la NBA, estadísticas de equipos/jugadores, y líneas de apuestas históricas.
- **Requisitos**: Conocimientos en web scraping, preparación de datos, modelado predictivo y evaluación de resultados.
- **Restricciones**: La naturaleza dinámica y cambiante del deporte profesional, lesiones, cambios de plantilla, etc.
- **Riesgos**: Sobreajuste a datos históricos, cambios en las reglas del juego o en el mercado de apuestas.

2. Comprensión de los Datos

Fuentes de Datos

- Basketball Reference: Estadísticas históricas de partidos, equipos y jugadores de la NBA.
- Vegas Insider: Líneas de apuestas (spreads, totales) históricas.
- ESPN: Estadísticas complementarias y actualizadas.

Descripción de los Datos

El conjunto de datos incluye:

- Resultados históricos de partidos de la NBA (puntuaciones, ganador/perdedor).
- Estadísticas de equipos (ofensivas y defensivas).
- Métricas avanzadas (eficiencia, ritmo de juego, etc.).
- Líneas de apuestas (spreads, over/under).
- Registros de temporadas anteriores.

Exploración de los Datos

- Correlaciones entre estadísticas de equipo y resultados contra el spread.
- Patrones temporales (desempeño en back-to-backs, después de largas giras, etc.).
- Tendencias históricas de resultados sobre/bajo el spread.

Verificación de Calidad de Datos

- Datos faltantes en estadísticas históricas.
- Inconsistencias en los formatos de datos entre diferentes fuentes.
- Posibles errores en los datos scrapeados que requieren limpieza.

3. Preparación de los Datos

Selección de Datos

- Selección de características relevantes que incluyen estadísticas ofensivas y defensivas de equipos.
- Enfoque en datos de las últimas temporadas para capturar tendencias actuales.
- Inclusión de métricas avanzadas como eficiencia ofensiva/defensiva, ritmo de juego, etc.

Limpieza de Datos

- Manejo de valores faltantes (especialmente en estadísticas de jugadores lesionados).
- Estandarización de formatos entre diferentes fuentes de datos.
- Eliminación de duplicados y corrección de inconsistencias.

Construcción de Características

- Creación de características derivadas como:
    - Diferencial de eficiencia entre equipos
    - Métricas de descanso (días desde el último partido)
    - Tendencias recientes (últimos N partidos)
    - Desempeño histórico contra el spread

Integración de Datos

- Combinación de datos de múltiples fuentes para crear un conjunto de datos coherente.
- Sincronización temporal correcta entre estadísticas de equipos y líneas de apuestas.

Formateo de Datos

- Normalización de variables numéricas.
- Codificación de variables categóricas (equipos, ubicación, etc.).
- División en conjuntos de entrenamiento y prueba, asegurando la separación temporal correcta.

4. Modelado

Selección de Técnicas de Modelado

El proyecto utiliza varios algoritmos de machine learning para la clasificación binaria:

- Regresión Logística (línea base)
- Random Forest
- Gradient Boosting
- SVM (Support Vector Machines)
- XGBoost

Diseño de Pruebas

- Validación cruzada con separación temporal para evitar data leakage.
- Evaluación de modelos en temporadas completas no vistas.
- Comparación de rendimiento entre diferentes algoritmos.

Construcción del Modelo

- Entrenamiento de modelos con hiperparámetros optimizados.
- Ensamblaje de modelos para mejorar la estabilidad de las predicciones.
- Implementación de pipeline de entrenamiento automatizado.

Evaluación del Modelo (técnica)

- Métricas de precisión, recall, F1-score.
- Curvas ROC y AUC para evaluar la capacidad discriminatoria.
- Análisis de la matriz de confusión para entender los patrones de error.

5. Evaluación

Evaluación de Resultados

- Evaluación del rendimiento predictivo en partidos reales.
- Análisis de rentabilidad (retorno de inversión) bajo diferentes estrategias de apuestas.
- Comparación con benchmarks (líneas de Vegas, expertos, etc.).

Proceso de Revisión

- Revisión de características más influyentes en las predicciones.
- Identificación de patrones donde el modelo tiene mejor/peor desempeño.
- Análisis de errores sistemáticos.

Determinación de Próximos Pasos

- Refinamiento del modelo basado en nueva información.
- Expansión a otros mercados de apuestas (totales, moneyline, etc.).
- Implementación de técnicas avanzadas (deep learning, refuerzo, etc.).

6. Despliegue

Plan de Implementación

- Script automatizado para ejecutar predicciones diarias.
- Sistema de actualización de datos y reentrenamiento periódico.
- Interfaz simple para visualizar recomendaciones.

Plan de Monitoreo y Mantenimiento

- Seguimiento continuo del rendimiento del modelo.
- Actualización de los datos con nuevos partidos.
- Reentrenamiento programado con datos más recientes.

Informe Final

- Documentación del proceso completo.
- Resumen de hallazgos y recomendaciones.
- Potencial para futuras mejoras.

Fortalezas, Debilidades y Mejoras por Fase CRISP-DM

1. Comprensión del Negocio

Fortalezas

- Objetivo de negocio claro y medible (predicción de apuestas deportivas con ROI positivo).
- Problema bien definido y acotado (predicción binaria: encima/debajo del spread).
- El dominio de aplicación (NBA) cuenta con abundantes datos públicos y estadísticas.

Debilidades

- Falta documentación explícita sobre la estrategia de apuestas (tamaño, frecuencia, gestión de bankroll).
- No se discuten las implicaciones legales y regulatorias de las apuestas deportivas.
- Ausencia de un análisis profundo del mercado de apuestas y sus ineficiencias.

Mejoras Propuestas

- Documentar una estrategia clara de apuestas incluyendo gestión de riesgo y bankroll.
- Incorporar un análisis de las casas de apuestas y sus diferencias (ventajas para el apostador).
- Definir criterios de confianza para las predicciones y umbrales para decidir cuándo apostar.
- Incluir un análisis FODA (fortalezas, oportunidades, debilidades y amenazas) del proyecto.

2. Comprensión de los Datos

Fortalezas

- Utilización de múltiples fuentes de datos complementarias.
- Inclusión de estadísticas avanzadas de la NBA.
- Capacidad de scraping automatizado para mantener datos actualizados.

Debilidades

- Falta un análisis exploratorio de datos (EDA) detallado y visual.
- No se documenta claramente la completitud y calidad de los datos recogidos.
- Ausencia de análisis de correlaciones entre variables y el resultado objetivo.

Mejoras Propuestas

- Implementar un EDA completo con visualizaciones de distribuciones y correlaciones.
- Crear un dashboard para el análisis visual de tendencias históricas.
- Documentar la estructura de datos, las fuentes y la frecuencia de actualización.
- Agregar análisis específicos sobre factores contextuales (lesiones, cambios de entrenador, etc.).

3. Preparación de los Datos

Fortalezas

- Automatización del proceso de recopilación y preparación de datos.
- Creación de características derivadas relevantes.
- Enfoque en datos recientes para capturar tendencias actuales.

Debilidades

- Falta de transparencia en el manejo de valores faltantes.
- No se documenta detalladamente la ingeniería de características.
- Ausencia de análisis de importancia de características.

Mejoras Propuestas

- Implementar técnicas avanzadas de imputación para valores faltantes.
- Documentar cada paso de transformación y limpieza de datos.
- Agregar características basadas en la dinámica de equipos y jugadores clave.
- Incorporar análisis de componentes principales (PCA) o selección automática de características.
- Considerar información contextual como rivalidades, partidos televisados, etc.

4. Modelado

Fortalezas

- Uso de múltiples algoritmos de clasificación para comparar rendimiento.
- Implementación de validación cruzada temporal para evaluar correctamente.
- Capacidad para reentrenar modelos con nuevos datos.

Debilidades

- Falta de detalles sobre la optimización de hiperparámetros.
- No se exploran modelos más avanzados como redes neuronales o modelos de series temporales.
- Ausencia de ensamblaje de modelos para mejorar la estabilidad.

Mejoras Propuestas

- Implementar búsqueda sistemática de hiperparámetros (Grid Search, Random Search, Bayesiano).
- Explorar arquitecturas de deep learning para capturar patrones más complejos.
- Desarrollar modelos específicos por equipos o divisiones.
- Implementar técnicas de ensamblaje (stacking, blending) para mejorar la precisión.
- Considerar modelos de series temporales para capturar tendencias y estacionalidad.

5. Evaluación

Fortalezas

- Evaluación basada en métricas relevantes para el dominio (ROI, precisión predictiva).
- Capacidad para evaluar el rendimiento en datos históricos reales.

Debilidades

- Falta de backtesting exhaustivo con diferentes estrategias de apuestas.
- No se analizan los errores del modelo en profundidad.
- Ausencia de comparativas con benchmarks (expertos, líneas de consenso, etc.).

Mejoras Propuestas

- Implementar backtesting completo con simulación de apuestas reales.
- Analizar los errores del modelo por tipo de partido, equipos, condiciones, etc.
- Comparar el rendimiento con predictores humanos y otros modelos públicos.
- Desarrollar métricas de confianza para cada predicción.
- Crear visualizaciones de rendimiento a lo largo del tiempo.

6. Despliegue

Fortalezas

- Script automatizado para realizar predicciones.
- Capacidad para actualizar datos y reentrenar periódicamente.

Debilidades

- Interfaz de usuario limitada.
- Falta de integración con sistemas de seguimiento de apuestas.
- No se menciona la monitorización continua del rendimiento.

Mejoras Propuestas

- Desarrollar una interfaz web/móvil para visualizar predicciones y resultados.
- Implementar notificaciones automáticas para oportunidades de apuestas.
- Crear un sistema de seguimiento y registro de resultados.
- Desarrollar un dashboard para monitorizar el rendimiento del modelo en tiempo real.
- Implementar alertas para detectar cambios significativos en el rendimiento del modelo.

Extensiones y Recomendaciones para el Proyecto NBA-ML-Betting

Propuestas de Extensión

1. Expansión del Alcance Predictivo

Mercados de Apuestas Adicionales

- **Predicción de totales (over/under)**: Desarrollar modelos específicos para predecir si el total de puntos superará o quedará por debajo de la línea establecida.
- **Apuestas de jugadores (prop bets)**: Crear modelos para predecir estadísticas individuales de jugadores (puntos, rebotes, asistencias).
- **Apuestas de primer cuarto/primera mitad**: Desarrollar modelos específicos para segmentos del partido.

Ligas y Deportes Adicionales

- Expandir la metodología a otras ligas de baloncesto (Euroliga, NCAA, WNBA).
- Adaptar el enfoque a otros deportes de equipo (NFL, MLB, NHL).

2. Mejoras Técnicas

Modelos Avanzados

- Implementar arquitecturas de deep learning (LSTM, GRU) para capturar dependencias temporales.
- Explorar modelos de atención para identificar factores clave en cada predicción.
- Desarrollar modelos específicos por equipo o por tipo de partido.

Ingeniería de Características Avanzada

- Incorporar datos de rastreo (tracking data) si están disponibles.
- Desarrollar características basadas en análisis de redes para modelar interacciones entre jugadores.
- Incluir métricas de momentum y tendencias recientes con ponderación temporal.

3. Integración y Automatización

Plataforma Integral

- Desarrollar una aplicación web/móvil completa con:
  - Dashboard de predicciones y resultados
  - Seguimiento de apuestas y rendimiento
  - Alertas y notificaciones para oportunidades de apuestas
  - Visualización de tendencias y patrones

Automatización Avanzada

- Integración con APIs de casas de apuestas para obtener líneas en tiempo real.
- Sistema automatizado de colocación de apuestas basado en las predicciones y niveles de confianza.
- Reentrenamiento automático cuando el rendimiento cae por debajo de umbrales predefinidos.

Recomendaciones Técnicas Detalladas

1. Mejoras en la Recopilación de Datos

```
# Ejemplo de mejora para el scraping de datos
def enhanced_data_scraper(season, include_advanced=True,
include_injuries=True):
    """
    Recopila datos mejorados con estadísticas avanzadas e información de
lesiones

    Args:
        season (int): Año de la temporada
        include_advanced (bool): Incluir métricas avanzadas
        include_injuries (bool): Incluir datos de lesiones

    Returns:
        pd.DataFrame: Datos completos de la temporada
    """
    # Código para scraping básico
    base_data = scrape_basic_stats(season)

    # Añadir métricas avanzadas si se solicitan
    if include_advanced:
        advanced_data = scrape_advanced_stats(season)
        base_data = pd.merge(base_data, advanced_data, on=['team_id',
'game_id'])

    # Añadir información de lesiones si se solicita
    if include_injuries:
        injury_data = scrape_injury_reports(season)
        base_data = enrich_with_injuries(base_data, injury_data)

    return base_data
```

2. Ingeniería de Características Innovadora

```
# Ejemplo de nuevas características para el modelo
def create_enhanced_features(df):
    """
    Crea características avanzadas para mejorar el poder predictivo

    Args:
        df (pd.DataFrame): DataFrame con datos básicos

    Returns:
        pd.DataFrame: DataFrame con características adicionales
    """
    # Momentum basado en últimos N partidos (ponderado por recencia)
    df['momentum_score'] = calculate_weighted_momentum(df, window=10)

    # Fatiga basada en calendario y minutos jugados
    df['fatigue_index'] = calculate_fatigue_index(df)

    # Métricas de desempeño contra el spread
    df['ats_performance_10g'] = calculate_ats_performance(df, window=10)

    # Índice de desajuste (matchup index) basado en estilos de juego
    df['style_mismatch_index'] = calculate_style_mismatch(df)

    # Ventaja de descanso relativa
    df['rest_advantage'] = df['days_rest'] - df['opponent_days_rest']

    # Tendencia de líneas (line movement)
    df['line_movement'] = df['closing_spread'] - df['opening_spread']

    return df
```

3. Evaluación Avanzada de Modelos

```
# Ejemplo de evaluación avanzada con simulación de apuestas
def evaluate_betting_performance(model, X_test, y_test, odds_test,
bankroll=1000, unit_size=0.03,
                                 confidence_threshold=0.6):
    """
    Evalúa el modelo simulando una estrategia de apuestas real

    Args:
        model: Modelo entrenado
        X_test: Características de prueba
        y_test: Valores reales
        odds_test: Momios asociados
        bankroll: Capital inicial
        unit_size: Tamaño de apuesta como fracción del bankroll
        confidence_threshold: Umbral mínimo de confianza para apostar

    Returns:
```

```
        dict: Métricas de rendimiento de apuestas
    """
    # Predicciones y probabilidades
    y_pred = model.predict(X_test)
    y_prob = model.predict_proba(X_test)[:, 1]  # Probabilidad de la clase
positiva

    # Simulación de apuestas
    results = []
    current_bankroll = bankroll
    bets_placed = 0

    for i in range(len(y_test)):
        # Solo apostar si la confianza supera el umbral
        confidence = max(y_prob[i], 1 - y_prob[i])

        if confidence >= confidence_threshold:
            prediction = y_pred[i]
            actual = y_test.iloc[i]
            odd = odds_test.iloc[i]

            # Calcular tamaño de apuesta (Kelly criterion simplificado)
            edge = calculate_edge(confidence, odd)
            bet_size = current_bankroll * unit_size * (edge / odd)
            bet_size = min(bet_size, current_bankroll * 0.05)  # Limitar
riesgo máximo

            # Resultado de la apuesta
            if prediction == actual:
                profit = bet_size * (odd - 1)
                current_bankroll += profit
                result = 'win'
            else:
                current_bankroll -= bet_size
                result = 'loss'

            results.append({
                'bet_number': bets_placed + 1,
                'confidence': confidence,
                'bet_size': bet_size,
                'result': result,
                'bankroll': current_bankroll
            })

            bets_placed += 1

    # Calcular métricas
    roi = (current_bankroll - bankroll) / bankroll
    win_rate = sum(1 for r in results if r['result'] == 'win') /
bets_placed if bets_placed > 0 else 0
    avg_bet_size = sum(r['bet_size'] for r in results) / bets_placed if
bets_placed > 0 else 0

    return {
```

```
        'final_bankroll': current_bankroll,
        'roi': roi,
        'bets_placed': bets_placed,
        'win_rate': win_rate,
        'avg_bet_size': avg_bet_size,
        'bet_history': results
    }
```

4. Visualización de Resultados e Insights

```
# Ejemplo de visualización de rendimiento de apuestas
def visualize_betting_performance(performance_results):
    """
    Crea visualizaciones del rendimiento de las apuestas

    Args:
        performance_results (dict): Resultados de la simulación de
apuestas
    """
    import matplotlib.pyplot as plt
    import seaborn as sns

    # Configuración de estilo
    sns.set_style('whitegrid')
    plt.figure(figsize=(14, 10))

    # Gráfico de evolución del bankroll
    plt.subplot(2, 2, 1)
    bankroll_data = [r['bankroll'] for r in
performance_results['bet_history']]
    bet_numbers = [r['bet_number'] for r in
performance_results['bet_history']]
    plt.plot(bet_numbers, bankroll_data, linewidth=2)
    plt.title('Evolución del Bankroll', fontsize=14)
    plt.xlabel('Número de Apuesta', fontsize=12)
    plt.ylabel('Bankroll ($)', fontsize=12)

    # Gráfico de distribución de confianza
    plt.subplot(2, 2, 2)
    confidence_data = [r['confidence'] for r in
performance_results['bet_history']]
    results = [1 if r['result'] == 'win' else 0 for r in
performance_results['bet_history']]
    plt.scatter(confidence_data, results, alpha=0.6)
    plt.title('Confianza vs. Resultado', fontsize=14)
    plt.xlabel('Confianza del Modelo', fontsize=12)
    plt.ylabel('Resultado (1=Win, 0=Loss)', fontsize=12)

    # Histograma de retornos por apuesta
    plt.subplot(2, 2, 3)
    wins = [r for r in performance_results['bet_history'] if r['result']
```

```
== 'win']
    losses = [r for r in performance_results['bet_history'] if r['result']
== 'loss']

    win_sizes = [w['bet_size'] for w in wins]
    loss_sizes = [-l['bet_size'] for l in losses]

    plt.hist([win_sizes, loss_sizes], bins=20, label=['Ganancias',
'Pérdidas'])
    plt.legend()
    plt.title('Distribución de Ganancias/Pérdidas', fontsize=14)
    plt.xlabel('Monto ($)', fontsize=12)
    plt.ylabel('Frecuencia', fontsize=12)

    # Métricas clave
    plt.subplot(2, 2, 4)
    plt.axis('off')
    plt.text(0.1, 0.9, f"ROI: {performance_results['roi']:.2%}",
fontsize=16)
    plt.text(0.1, 0.8, f"Apuestas Totales:
{performance_results['bets_placed']}", fontsize=16)
    plt.text(0.1, 0.7, f"Tasa de Victoria:
{performance_results['win_rate']:.2%}", fontsize=16)
    plt.text(0.1, 0.6, f"Tamaño Promedio de Apuesta:
${performance_results['avg_bet_size']:.2f}", fontsize=16)
    plt.text(0.1, 0.5, f"Bankroll Final:
${performance_results['final_bankroll']:.2f}", fontsize=16)

    plt.tight_layout()
    plt.savefig('betting_performance.png', dpi=300)
    plt.show()
```

Recomendaciones de Implementación

1. Enfoque de Desarrollo Iterativo

1. **Fase 1**: Mejorar la recopilación y preparación de datos.
    - Implementar scraping más robusto y completo.
    - Desarrollar pipeline de limpieza y transformación avanzada.
2. **Fase 2**: Optimizar los modelos predictivos.
    - Comparar sistemáticamente diferentes algoritmos.
    - Implementar optimización de hiperparámetros automatizada.
    - Desarrollar ensamblaje de modelos.
3. **Fase 3**: Crear sistema de evaluación y simulación.
    - Implementar backtesting con estrategias de apuestas realistas.
    - Desarrollar métricas de evaluación específicas para apuestas.

4. **Fase 4**: Desarrollar interfaz y automatización.
    - Crear dashboard para visualización de predicciones y resultados.
    - Implementar sistema de alertas y notificaciones.
    - Automatizar el flujo completo (datos → predicción → recomendación).

2. Validación y Pruebas

- Realizar pruebas con apuestas simuladas (paper trading) durante al menos una temporada completa.
- Implementar gradualmente con apuestas de montos pequeños para validar el rendimiento real.
- Establecer procesos de validación continua y ajuste de modelos.

3. Consideraciones Éticas y Legales

- Investigar y cumplir con las regulaciones locales sobre apuestas deportivas.
- Implementar prácticas de juego responsable (límites, alertas, etc.).
- Documentar transparentemente las limitaciones del modelo y los riesgos asociados.

Conclusión

El proyecto "NBA-Machine-Learning-Sports-Betting" tiene un sólido fundamento técnico y un objetivo claro, pero puede beneficiarse significativamente de las mejoras propuestas en todas las fases del proceso CRISP-DM. Con la implementación de estas recomendaciones, el sistema podría evolucionar desde un proyecto experimental hacia una herramienta robusta y potencialmente rentable para la toma de decisiones en apuestas deportivas.