PSet #3 - Modelos de Boosting para Invertir en Bolsa

Curso de Minería de Datos

28 de marzo de 2025

1. Introducción y Contexto

En el ámbito de la inversión en bolsa, la toma de decisiones basada en datos se ha vuelto un factor determinante para obtener rendimientos competitivos. El objetivo de este PSet #3 es que los estudiantes diseñen y evalúen un modelo de **algorithmic trading** que sea **rentable**, aplicando diversas técnicas de ensamble y *boosting*.

Cada estudiante (o equipo) tendrá la libertad de:

- Seleccionar una canasta de activos o títulos (acciones, ETFs, etc.) para conformar su portafolio.
- Definir la estrategia de trading, **incluyendo la frecuencia** (intradía, diaria, semanal) y las variables (indicadores técnicos, fundamentales, sentiment analysis, etc.).
- Escoger y entrenar distintos modelos de aprendizaje automático (árboles de decisión, bosques aleatorios, y los métodos de *boosting* requeridos).

2. Objetivo Principal

Desarrollar y entrenar modelos de **boosting** (*LightGBM*, *XGBoost*, *CatBoost* y *AdaBoost*) para la predicción de la variación en el precio (o el rendimiento) de los activos elegidos. Se busca:

1. **Maximizar la rentabilidad** o la métrica de *trading* que se seleccione (p.ej. *accuracy* en la predicción de subidas/bajadas, *RMSE* en la predicción de precios, *Sharpe ratio*, etc.).
2. Comparar los resultados de estos modelos con otros métodos de ensamble (ej. Random Forest o Gradient Boosting "clásico") y con árboles de decisión simples.
3. Escoger el **mejor modelo** según métricas financieras y de error.

3. Indicaciones Generales

Para la resolución del PSet #3, se sugiere el siguiente flujo de trabajo:

3.1. 1. Búsqueda y Preparación del Dataset

- Seleccionar una fuente de datos fiable para descargar el histórico de precios de las acciones o activos que formarán el *portafolio*.
- Incluir otras variables complementarias como indicadores técnicos (`SMA`, `RSI`, `MACD`, etc.), fundamentales (si aplica) o *sentiment analysis* (opcional).
- Dividir las columnas en **features** (entradas al modelo) y **targets** (precio futuro o etiqueta de subida/bajada).

3.2. 2. Análisis Exploratorio (EDA) y Data Wrangling

- Describir estadísticamente la distribución de precios, rendimientos, volumen, etc.
- Identificar outliers o valores atípicos y decidir cómo manejarlos.
- Tratar valores faltantes, si los hubiere (p.ej. cierre de mercado o días festivos).
- Realizar transformaciones necesarias (escalado, normalización, etc.) para preparar los datos de cara al modelado.

3.3. 3. Feature Engineering

- Calcular indicadores técnicos relevantes para la estrategia de trading (promedios móviles, índices de fuerza relativa, osciladores, etc.).
- Crear variables derivadas (diferencias porcentuales, señales de volúmenes inusuales, etc.).
- Seleccionar las **características** con mayor correlación o relevancia para la predicción según su criterio.

3.4. 4. Definición del Problema de Modelado

- Decidir si se trata de un problema de *regresión* (predicción de precio futuro) o *clasificación* (predecir subida/bajada).
- Establecer la ventana de predicción (e.g. predecir el precio al siguiente día, o la ganancia porcentual en la próxima semana).

3.5. 5. Entrenamiento de Modelos

a) Árbol de decisión simple (*baseline*).

b) Bosques Aleatorios (*Random Forest*).

c) AdaBoost.

d) Gradient Boosting (pueden usar *sklearn* u otras librerías).

e) **LightGBM**.

f) **XGBoost**.

g) **CatBoost**.

Nota: Comparar estos métodos basados en:

- Métricas financieras seleccionadas (p.ej. *accuracy* para clasificación, MSE/RMSE para regresión, ratio de ganancia acumulada, *Sharpe ratio*, etc.).
- Tiempo de entrenamiento y facilidad de ajuste de hiperparámetros.

3.6. 6. Validación, Evaluación y Selección

- Separar los datos en entrenamiento y prueba (pueden usar validación cruzada si lo consideran).
- Evaluar las métricas en el conjunto de *test*.
- Ajustar hiperparámetros (*grid search*, *random search* o *bayesian search*) de **LightGBM**, **XGBoost**, **CatBoost** y **AdaBoost**.
- Seleccionar el modelo de mejor desempeño según la métrica objetiva (p.ej. *maximizar retorno*, *accuracy*, *minimizar error*, etc.).

3.7. 7. Conclusiones y Recomendaciones

- Discutir sobre la utilidad real del modelo para *algorithmic trading*.
- Identificar limitaciones (datos insuficientes, *overfitting*, sesgos temporales, etc.).
- Proponer mejoras para futuros trabajos (más datos, enfoque multi-asset, modelos híbridos, etc.).

4. Estructura Recomendada de Carpeta en el Repositorio

A fin de seguir buenas prácticas de la industria, se sugiere usar la siguiente estructura de carpetas:

```
.
 data
    raw
    processed
 notebooks
    1_EDA_DataWrangling.ipynb
    2_FeatureEngineering.ipynb
    3_Modeling.ipynb
    4_Evaluation_Selection.ipynb
    5_FinalAnalysis.ipynb
 src
    data_utils.py
    features_utils.py
    modeling_utils.py
    metrics_utils.py
 models
    (archivos .pkl, .joblib o checkpoints)
 docs
    (reportes, presentaciones, etc.)
 README.md
 requirements.txt
```

- **data**:
  - `raw`: Conjunto de datos originales descargados.
  - `processed`: Datos limpios, transformados o divididos en `train/test`.
- **notebooks**:
  - Cada `.ipynb` representa una fase del proyecto (EDA, feature engineering, modelado, etc.).
- **src**: Scripts con funciones reutilizables (carga de datos, transformaciones, training loops, etc.).
- **models**: Modelos entrenados, ya sea en formato `.pkl` u otro.
- **docs**: Lugar para guardar documentación, PDF final, presentaciones.
- **README.md**: Descripción del proyecto, cómo ejecutarlo, dependencias, etc.
- **requirements.txt**: Lista de librerías o entornos necesarios (`pandas`, `numpy`, `matplotlib`, `sklearn`, `xgboost`, `lightgbm`, `catboost`, etc.).

5. Entrega y Evaluación

5.1. Informe Final

- Incluir un **informe en formato PDF** dentro de la carpeta `docs/` (o adjunto a la plataforma que se use).
- El informe debe detallar cada uno de los pasos seguidos (EDA, data wrangling, creación de features, metodología de modelado y evaluación, etc.).
- Presentar tablas de resultados, métricas y gráficas que comparen los distintos modelos de boosting (`LightGBM`, `XGBoost`, `CatBoost` y `AdaBoost`) y, opcionalmente, otros métodos de ensamble.

5.2. Entrega en Repositorio

- Subir todo el proyecto a **GitHub** (o sistema de control de versiones equivalente).
- Incluir la URL del repositorio en el informe final.

5.3. Criterios de Evaluación

1. **Calidad del EDA y Data Wrangling** (manejo de outliers, valores faltantes, transformaciones).
2. **Feature Engineering e Ingeniería de Indicadores Técnicos** (relevancia y justificación).
3. **Implementación Correcta de los Modelos de Ensamble y Boosting** (`LightGBM`, `XGBoost`, `CatBoost`, `AdaBoost`, etc.).
4. **Hiperparámetros y Comparación de Resultados** (métricas elegidas y coherencia con la tarea).
5. **Conclusiones de Negocio** (interpretación de los resultados y viabilidad de la estrategia de trading).
6. **Organización y Documentación del Proyecto** (estructura de carpetas, claridad del `README`, reporte final).

¡Éxitos con el PSet #3 y a descubrir la mejor estrategia de inversión basada en Boosting!