PSET #5

Boosting

Giselle Cevallos
Anahí Andrade
Erick Suárez
Jesús Alarcón

BOOSTING

En el ámbito del Data Mining y el Machine Learning, los algoritmos de ensamble (ensemble methods) se han convertido en herramientas fundamentales para mejorar la precisión de los modelos predictivos. Entre ellos, el método Boosting es uno de los más influyentes debido a su capacidad para convertir modelos débiles en modelos fuertes mediante un proceso secuencial de aprendizaje.

Este informe describe los fundamentos del Boosting y desarrolla un análisis detallado del algoritmo AdaBoost, uno de los primeros y más importantes exponentes de esta familia.

QUE ES EL BOOSTING?

Boosting es una técnica de ensamble que combina múltiples clasificadores débiles (weak learners) para crear un clasificador fuerte (strong learner).

Weak Learner:

Un weak learner es un modelo cuya precisión solo es ligeramente superior a la probabilidad de un clasificador aleatorio.

Ejemplo: un árbol de decisión muy pequeño (stump) de un solo nivel.

MECANISMO

Se basa en entrenar modelos de manera secuencial, donde cada modelo corrige los errores del modelo anterior.

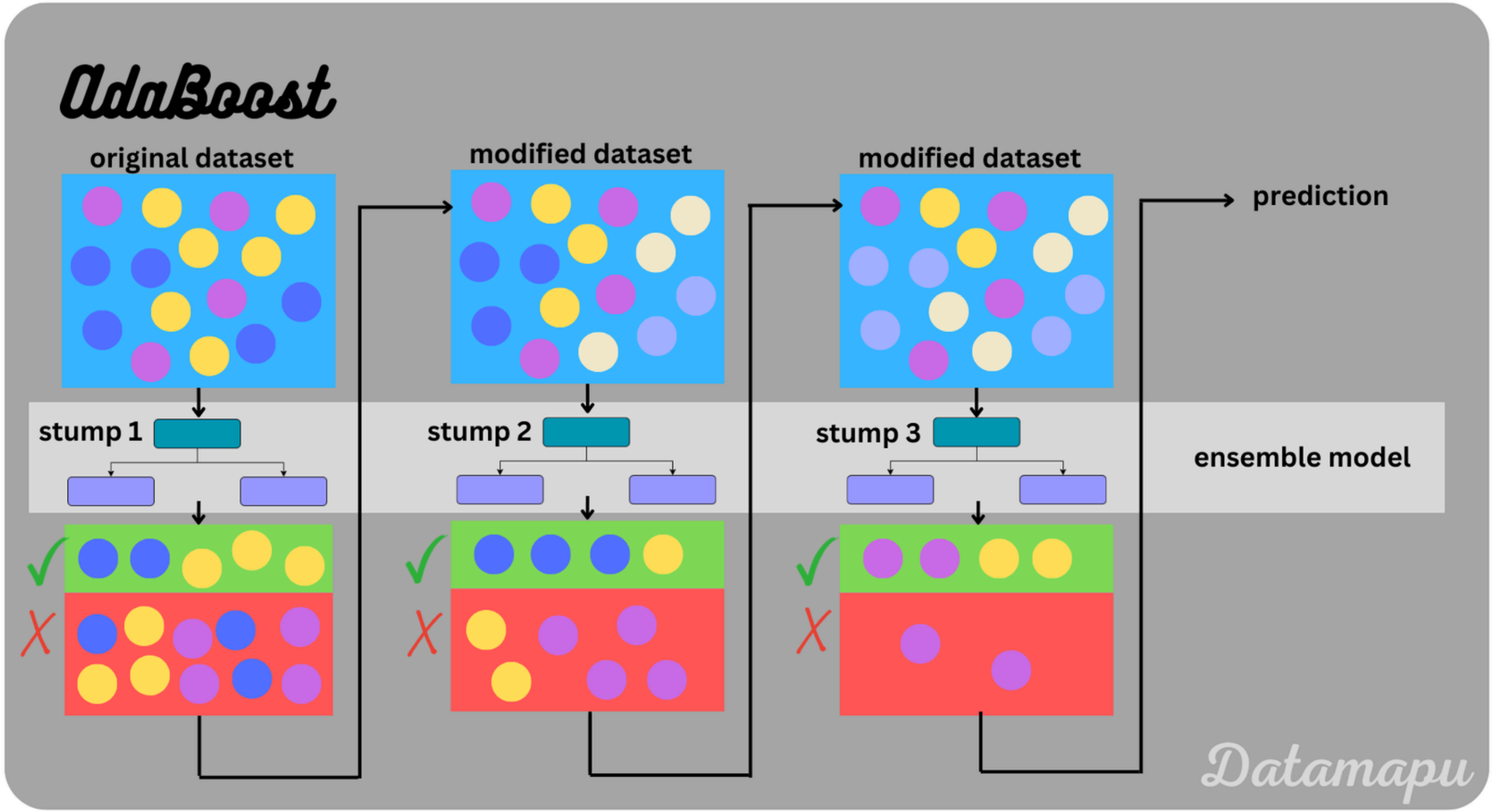
- Se entrena un modelo débil.
- Se evalúan sus errores.
- Se asignan más pesos a los datos mal clasificados.
- Se entrena un nuevo modelo que se enfoque en esas instancias difíciles.
- Se repite el proceso.
- La predicción final se obtiene mediante una combinación ponderada de todos los modelos.

ADAPTIVE BOOSTING

FUNCIONAMIENTO

- Inicializa los pesos.
- Cada observación inicia con el mismo peso.
- Entrena un weak learner.
- Usualmente un árbol de decisión tipo “decision stump” (profundidad = 1).
- Evalúa los errores.
- Se identifica qué instancias fueron mal clasificadas.
- Aumenta el peso de los errores.
- AdaBoost da mayor importancia a las observaciones mal clasificadas.
- Calcula la contribución del modelo.
- Modelos más precisos obtienen mayor peso; los que fallan mucho, menor peso.
- Combina los modelos.
- La predicción final es un voto ponderado entre todos los weak learners.

FUNCIONAMIENTO



EXPONENTIAL LOSS

La función de pérdida que AdaBoost minimiza es la exponential loss:

$$L = \sum_{i=1}^{m} \exp\left(-y_i F(x_i)\right)$$

- Penaliza de forma exponencial los errores.
- Hace que el algoritmo se enfoque en los ejemplos difíciles.
- Permite la interpretación de AdaBoost como un método de optimización aditiva.

FÓRMULAS ESENCIALES

Inicializacion con concujunto de entrenamiento

conjunto de entrenamiento (X, Y) m observaciones denotadas $(x_1, y_1), \ldots, (x_m, y_m)$

Los valores de y deben ser -1 o 1 para aplicar el método.

Inicie con la distribución discreta

$D_1(i) = 1/m$ observación i en la iteración 1

Para T $t = 1, \ldots, T.$

Construya un clasificador ht

$h_t : X \to \{-1, 1\}.$

Calcule el error asociado ϵt al clasificador (weak learner)

$\epsilon_t = \sum_{i=1}^{m} D_t(i) \times \delta_i$, donde $\delta_i = 0$ si $h_t(x_i) = y_i$,

Si el error es mayor a 0.5, AdaBoost lo descarta automáticamente.

FÓRMULAS ESENCIALES

es decir, si fue correcta la clasificación; caso contrario es δi=1.

Calcule la nueva distribución

$$D_{t+1}(i) = D_t(i) \times F_i / Z_t,$$

Repetimos el proceso T veces

Analisis de resultados

- $F_i = \exp(-\alpha_t)$ si la clasificación fue correcta, es decir si $h_t(x_i) = y_i$.
- $F_i = \exp(\alpha_t)$ si la clasificación fue incorrecta, es decir si $h_t(x_i) \neq y_i$.
- $\alpha_t = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
- Z_t es una constante de normalización de tal manera que $\sum_{i=1}^{m} D_t(i) = 1$. Usualmente es $\sum D_t(i) \times F_i$.

Los puntos mal clasificados se vuelven más importantes para la siguiente iteración.

→ Si el error es bajo, alpha es grande.
→ Si el error es alto, alpha es pequeño.

FÓRMULAS ESENCIALES

Construye el clasificador final

promedio ponderado de los t clasificadores ht

$$H_{final} = sign(\sum_t \alpha_t h_t(x)).$$

1. Los clasificadores buenos (con mayor alpha) influyen más en la decisión.

FÓRMULAS ESENCIALES

Construya el clasificador final

promedio ponderado de los t clasificadores ht

$$H_{final} = sign(\sum_t \alpha_t h_t(x)).$$

1. Los clasificadores buenos (con mayor alpha) influyen más en la decisión.

COMPARACIÓN ADABOOST VS. OTROS BOOSTERS MODERNOS

Algoritmo	Caracteristicas	Ventajas	Desventajas
AdaBoost	Boosting clásico, usa weights	Fácil de impleme ntar, interpretable	Sensible a outliers
Gradient Boosting	Minimiza función de pérdida por gradiente	Flexible, potente	Lento, requiere ajuste
XGBoost	Optimización extrema de GBM	Muy rápido y exacto	Más complejo
LightGBM	Usa árboles con histogramas	Ultra rapido, escala bien	Requiere tuning
CatBoost	Especializado en variables categóricas	Evita one-hot encoding	Más pesado en entrenamiento

SAMME VS SAMME.R

- **SAMME**
  - Usa predicciones categóricas.
  - Más simple, más antiguo.
  - Puede ser menos estable.

- **SAMME.R**
  - Usa predicciones probabilísticas.
  - Mejor desempeño en multiclase.
  - Es la versión usada por defecto en scikit-learn.

HIPERPARÁMETROS CLAVE

Hiperparámetro	Efecto
n_estimators (50–800)	Capacidad del modelo; más → riesgo de overfitting.
learning_rate (0.01–1)	Controla intensidad del update.
base_estimator.max_depth (1–3)	Profundidad del stump.
algorithm	SAMME/SAMME.R.
random_state	Reproducibilidad.

VENTAJAS Y LIMITACIONES

- **Ventajas**
  - Interpretable.
  - Funciona bien con modelos simples.
  - Enfoca el aprendizaje en ejemplos difíciles.

- **Limitaciones**
  - Extremadamente sensible a outliers.
  - Puede sobreajustar rápido.
  - No escala tan bien como XGB/LGBM.

VENTAJAS Y LIMITACIONES

- **Buenas Prácticas**
  - Mantener learning_rate bajo (0.01–0.1).
  - Usar muchos árboles pequeños (stumps).
  - Controlar outliers antes del entrenamiento.
  - Revisar curvas train/validación.
  - Mantener profundidad baja del weak learner.

RESULTADOS

Dataset: 300,000 registros de taxis NYC

- Balanceado: 100,000 muestras por año
- Variable objetivo: total_amount
- Semilla aleatoria fija: 42

12 modelos evaluados:

- Baselines: Media, Regresión Lineal
- Boosting: AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost
- Ensemble: Voting, Bagging, Pasting Métrica principal: RMSE

RESULTADOS

MODELO	RMSE VAL	MAE VAL	R² VAL	TIEMPO (S)	OBSERVACIONES
1 GradientBoosting	$2.57	$0.90	0.9680	345.07	*Mejor balance general*
2 VotingRegressor	$2.58	$0.96	0.9677	17.64	*Muy eficiente*
3 Pasting	$2.59	$0.94	0.9676	31.68	-
4 Bagging	$2.59	$0.94	0.9676	26.72	*Similar a Pasting*
5 CatBoost	$2.60	$0.98	0.9673	24.10	*Rápido y preciso*
6 Baseline-LinearReg	$2.72	$1.12	0.9642	0.00	*Referencia base*
7 LightGBM	$2.81	$0.93	0.9618	59.05	-
8 XGBoost	$3.39	$0.95	0.9444	12.76	*Rendimiento inferior al esperado*
9 AdaBoost	$3.91	$2.19	0.9258	226.35	*Alto error en outliers*
10 Baseline-Mean	$14.44	$8.25	-0.01	0.00	-

ANÁLISIS DE DESEMPEÑO ADABOOST

Posición: 9 de 12 modelos
RMSE Validación: $3.91 | MAE: $2.19
¿Por qué el bajo rendimiento?

ALTA SENSIBILIDAD A OUTLIERS

- DATASET CONTIENE TARIFAS INUSUALMENTE ALTAS
- PESOS EXPONENCIALES MAGNIFICAN EL ERROR EN OUTLIERS
- EL MODELO SE SESGA INTENTANDO CORREGIR CASOS PERDIDOS

LIMITACIÓN DE LOS STUMPS

- PROFUNDIDAD = 1 LIMITA LA COMPLEJIDAD
- INCAPAZ DE CAPTURAR INTERACCIONES NO LINEALES
- ZONAS DE NYC REQUIEREN MAYOR PROFUNDIDAD DE ÁRBOL

HALLAZGOS

GANADOR: GRADIENT BOOSTING

- RMSE TEST: $2.71 | R^2: 0.9748
- MEJORA DEL 5.52% VS BASELINE LINEAL
- EXCELENTE EN CORRECCIÓN DE ERRORES DE MEDIA DISTANCIA

VOTINGREGRESSOR: ANÁLISIS DESTACABLE

- RMSE: $2.58 (PRÁCTICAMENTE IGUAL AL MEJOR)
- TIEMPO: 17S VS 345S DE GRADIENT BOOSTING
- IDEAL PARA IMPLEMENTACIÓN EN TIEMPO REAL

CATBOOST: MÁS ROBUSTO

- BUEN RENDIMIENTO “OUT-OF-THE-BOX”
-SE NECESITA UN AJUSTE BAJO DE HIPERPARÁMETROS

PERSPECTIVA DE NEGOCIO

PREDICTOR DOMINANTE: BASE_FARE_COMPONENTS

PATRONES DESCUBIERTOS:

- RUSH_HOUR: TARIFAS DINÁMICAS SEGÚN HORA
- BOROUGH: CONTEXTO GEOGRÁFICO IMPORTA
- EL MODELO ENTIENDE CONTEXTO TEMPORAL Y ESPACIAL

PRECISIÓN ALCANZADA: MAE $0.99

→ PREDICCION DE +- 1 EN LA MAYORIA DE CASOS
→ SUFICIENTE PARA ESTIMACIONES DE PRECIO CERRADAS
→ MEJORA LA EXPERIENCIA DEL USUARIO ANTES DE ABORDAR

GRACIAS

Boosting