Boosting

Analítica de Datos, Universidad de San Andrés

¿Qué es Boosting?

Boosting es una técnica de ensamble que combina múltiples modelos 'débiles' para crear un modelo 'fuerte'. A diferencia de Random Forest, que construye árboles independientes en paralelo, Boosting construye modelos secuencialmente.

Principios Fundamentales del Boosting

Características Clave:

Principios Fundamentales del Boosting

Características Clave:

- Construcción secuencial de modelos

Principios Fundamentales del Boosting

Características Clave:

- Construcción secuencial de modelos
- Ponderación de observaciones

Principios Fundamentales del Boosting

Características Clave:

- Construcción secuencial de modelos
- Ponderación de observaciones
- Combinación ponderada de predicciones

Principios Fundamentales del Boosting

Características Clave:

- Construcción secuencial de modelos
- Ponderación de observaciones
- Combinación ponderada de predicciones
- Enfoque en errores previos

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

- Todas las observaciones inician con el mismo peso

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

- Todas las observaciones inician con el mismo peso
- En cada iteración:

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

- Todas las observaciones inician con el mismo peso
- En cada iteración:
    - Se entrena un modelo débil

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

- Todas las observaciones inician con el mismo peso
- En cada iteración:
  - Se entrena un modelo débil
  - Se identifican errores

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

- Todas las observaciones inician con el mismo peso
- En cada iteración:
  - Se entrena un modelo débil
  - Se identifican errores
  - Se aumenta el peso de errores

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

- Todas las observaciones inician con el mismo peso
- En cada iteración:
  - Se entrena un modelo débil
  - Se identifican errores
  - Se aumenta el peso de errores
  - Se disminuye el peso de aciertos

AdaBoost (1995)

Desarrollado por Yoav Freund y Robert Schapire. Primer algoritmo de Boosting exitoso.

Características:

- Todas las observaciones inician con el mismo peso
- En cada iteración:
  - Se entrena un modelo débil
  - Se identifican errores
  - Se aumenta el peso de errores
  - Se disminuye el peso de aciertos
- Predicción final por promedio ponderado

Gradient Boosting - Características

Características y Ventajas:

Gradient Boosting - Características

Características y Ventajas:

- Utiliza el gradiente del error para identificar la dirección de máximo descenso

Gradient Boosting - Características

Características y Ventajas:

- Utiliza el gradiente del error para identificar la dirección de máximo descenso
- Aplicable a cualquier función de pérdida diferenciable

Gradient Boosting - Características

Características y Ventajas:

- Utiliza el gradiente del error para identificar la dirección de máximo descenso
- Aplicable a cualquier función de pérdida diferenciable
- Más flexible que AdaBoost al no limitarse a clasificación binaria

Gradient Boosting - Características

Características y Ventajas:

- Utiliza el gradiente del error para identificar la dirección de máximo descenso
- Aplicable a cualquier función de pérdida diferenciable
- Más flexible que AdaBoost al no limitarse a clasificación binaria
- Mayor control sobre el proceso de aprendizaje

Gradient Boosting - Parámetros

Parámetros Clave:

Gradient Boosting - Parámetros

Parámetros Clave:

- n_estimators: Número de árboles

Gradient Boosting - Parámetros

Parámetros Clave:

- n_estimators: Número de árboles
- learning_rate: Tasa de aprendizaje

Gradient Boosting - Parámetros

Parámetros Clave:

- n_estimators: Número de árboles
- learning_rate: Tasa de aprendizaje
- max_depth: Profundidad máxima

Gradient Boosting - Parámetros

Parámetros Clave:

- n_estimators: Número de árboles
- learning_rate: Tasa de aprendizaje
- max_depth: Profundidad máxima
- subsample: Fracción de datos por árbol

XGBoost - Optimizaciones

Optimizaciones:

XGBoost - Optimizaciones

Optimizaciones:

- Paralelización y optimización de memoria: Procesamiento más rápido usando múltiples núcleos

XGBoost - Optimizaciones

Optimizaciones:

- Paralelización y optimización de memoria: Procesamiento más rápido usando múltiples núcleos
- Estructuras de datos especializadas: Formatos optimizados para mejor rendimiento

XGBoost - Optimizaciones

Optimizaciones:

- Paralelización y optimización de memoria: Procesamiento más rápido usando múltiples núcleos
- Estructuras de datos especializadas: Formatos optimizados para mejor rendimiento
- Regularización L1 y L2 incorporada: Previene overfitting automáticamente

XGBoost - Optimizaciones

Optimizaciones:

- Paralelización y optimización de memoria: Procesamiento más rápido usando múltiples núcleos
- Estructuras de datos especializadas: Formatos optimizados para mejor rendimiento
- Regularización L1 y L2 incorporada: Previene overfitting automáticamente
- Manejo automático de valores faltantes: Procesa datos incompletos sin preprocesamiento

XGBoost - Características Avanzadas

Características Avanzadas:

XGBoost - Características Avanzadas

Características Avanzadas:

- Early stopping inteligente: Monitorea el rendimiento y detiene el entrenamiento cuando no hay mejora

XGBoost - Características Avanzadas

Características Avanzadas:

- Early stopping inteligente: Monitorea el rendimiento y detiene el entrenamiento cuando no hay mejora
- Feature Importance: Proporciona métricas precisas sobre la contribución de cada variable

XGBoost - Características Avanzadas

Características Avanzadas:

- Early stopping inteligente: Monitorea el rendimiento y detiene el entrenamiento cuando no hay mejora
- Feature Importance: Proporciona métricas precisas sobre la contribución de cada variable
- Procesamiento optimizado de datos dispersos: Utiliza estructuras especializadas para matrices sparse

XGBoost - Características Avanzadas

Características Avanzadas:

- Early stopping inteligente: Monitorea el rendimiento y detiene el entrenamiento cuando no hay mejora
- Feature Importance: Proporciona métricas precisas sobre la contribución de cada variable
- Procesamiento optimizado de datos dispersos: Utiliza estructuras especializadas para matrices sparse
- Entrenamiento distribuido: Permite escalar el entrenamiento a múltiples núcleos de procesamiento

Comparación: Construcción

Característica	Random Forest	Boosting
Construcción	Paralela	Secuencial

- Random Forest: Árboles independientes construidos simultáneamente
- Boosting: Modelos construidos secuencialmente, cada uno corrigiendo errores del anterior

Comparación: Velocidad

Característica	Random Forest	Boosting
Velocidad	Más rápido	Más lento

- Random Forest: Paralelización natural, más rápido pero menos preciso
- Boosting: Secuencial, más lento pero potencialmente más preciso

Comparación: Overfitting

Característica	Random Forest	Boosting
Overfitting	Menos propenso	Más propenso

- Random Forest: Menor riesgo de overfitting por promediado
- Boosting: Mayor riesgo de overfitting por ajuste secuencial

Comparación: Ajuste

Característica	Random Forest	Boosting
Ajuste	Menos parámetros	Más parámetros

- Random Forest: Parámetros principales: n_trees, max_depth
- Boosting: Más parámetros: learning_rate, n_estimators, subsample, etc.

Comparación: Interpretabilidad

Característica	Random Forest	Boosting
Interpretabilidad	Mayor	Menor

- Random Forest: Importancia de variables más directa
- Boosting: Importancia de variables más compleja de interpretar

Implementación AdaBoost

```
from sklearn.ensemble import AdaBoostClassifier

ada_model = AdaBoostClassifier(
    n_estimators=100,
    learning_rate=1.0
)
```

Implementación Gradient Boosting

```
from sklearn.ensemble import GradientBoostingClassifier

gb_model = GradientBoostingClassifier(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=3
)
```

Implementación XGBoost

```python
import xgboost as xgb

xgb_model = xgb.XGBClassifier(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=3,
    reg_lambda=1,  # L2
    reg_alpha=0    # L1
)
```

Entrenamiento y Predicción: como siempre

```
# Entrenamiento del modelo
model.fit(X_train, y_train)

# Prediccion
y_pred = model.predict(X_test)
```

Buenas Prácticas

- **Preprocesamiento:**
  - Escalar variables numéricas
  - Codificar variables categóricas

Buenas Prácticas

- **Preprocesamiento:**
  - Escalar variables numéricas
  - Codificar variables categóricas
- **Validación:**
  - Validación cruzada
  - Ajuste de hiperparámetros

Buenas Prácticas

- **Preprocesamiento:**
  - Escalar variables numéricas
  - Codificar variables categóricas
- **Validación:**
  - Validación cruzada
  - Ajuste de hiperparámetros
- **Optimización:**
  - Early stopping
  - Learning rate bajo (0.01-0.1)
  - Monitoreo de errores