Auctioned Car Predictor

Problem Statement

Auto dealerships face a critical issue while purchasing used cars at auctions: the risk of buying vehicles with serious defects. These problematic purchases, referred to as **"kicks"**, include cars with odometer tampering, mechanical issues, title problems, or other hidden damages. Kicked cars can cause **financial losses** due to transport costs, unplanned repairs, and reselling difficulties.

Goal:

Build a Machine Learning model that predicts whether a car purchase will be a *bad buy* (kick) or not, helping dealerships make smarter purchasing decisions and reduce losses.

Dataset Description

- **Source:** Kaggle
- **Size:** ~150,000 rows × 32 features + 1 target
- **Target Variable:** IsBadBuy (1 = Bad Buy, 0 = Good Buy)

Key Features:

Feature	Description
Auction, Make, Model	Categorical vehicle info
VehOdo, VehYear, VehicleAge	Car age and mileage
Transmission, Color, WheelType	Car configuration
MMRAcquisition*, MMRCurrent*	Market price data
IsOnlineSale, WarrantyCost	Deal-level features

EDA Insights

- **Imbalanced Classes:** Bad buys are less frequent (~13%), requiring careful handling.
- **Missing Values:** Detected in WheelType, PRIMEUNIT, AUCGUART, handled via imputation or category "Unknown".

- **Correlations:**
    - VehicleAge ↔ IsBadBuy: Older cars have higher kick risk.
    - MMRCurrentRetailCleanPrice showed strong influence on buy decision.
- **Top Bad Buy Makes:** Certain manufacturers show higher kick rates — flagged as risky.

Plots included:

- Class imbalance bar chart
- Price vs kick rate scatter
- Heatmap of feature correlation

Feature Engineering

- Converted PurchDate, KickDate to datetime & extracted year/month
- Imputed missing values with:
    - Mode for categorical
    - Median for numerical
- Label encoding for ordinal/categorical features
- Created new feature: PriceGap = MMRCurrentRetailCleanPrice - VehBCost
- Standardized numeric columns for ML compatibility

Models Tried & Hyperparameter Tuning

Baseline Model:

- DummyClassifier (Stratified) → Accuracy: ~86% (misleading due to imbalance)

ML Models:

Model	Accuracy	ROC-AUC	F1 Score
Logistic Regression	✅ Decent	⚠️ Weak	⚠️ Poor
Random Forest	✅ Good	✅ 0.83	✅ 0.67

Model	Accuracy	ROC-AUC	F1 Score
XGBoost	✅ Better	✅ 0.86	✅ 0.71
LightGBM	✅ Fast & Good	✅ 0.85	✅ 0.70

Best Model: XGBoost

Tuned using RandomizedSearchCV over:

- n_estimators, max_depth, learning_rate, subsample, colsample_bytree

---

Final Model Performance

Metric	Value
Accuracy	88.5%
ROC-AUC	0.86
F1 Score	0.71
Precision	0.73
Recall	0.69

✅ **Confusion Matrix**
✅ **ROC Curve**
✅ **Feature Importances Plot**

Challenges Faced & Solutions

Challenge	Solution
Imbalanced dataset	Used stratified train-test split & F1 as main metric
High cardinality features	Label encoding + aggregation

Challenge	Solution
Overfitting	Cross-validation + hyperparameter tuning
Interpretability	Used feature importance + SHAP for explanation

---

Future Scope

- Deploy model as a web app for real-time dealership use (Streamlit / Azure)
- Add more recent auction data for better generalization
- Use SHAP / LIME for more explainable predictions
- Explore hybrid ensemble (RF + XGBoost + LightGBM)
- Create a dashboard to monitor model predictions over time