California Traffic and Pedestrian Stops Project

Boliang Liu, Shiqi Tao, Yingtong Lin, Wenyao Zhang
3.11.2021

Project Introduction

- Traffic stops:a **temporary detention** of a driver of a vehicle by police to investigate **a possible crime** or **minor violation of law**

Analytics Goals

- Final goal: Predict whether the violated driver or pedestrian will be arrested?

- 4 machine learning algorithms:
  - Random Forest
  - Gradient Boosting
  - DeepLearning
  - XGBoost

Data Description

- Data Source:

  https://www.kaggle.com/stanford-open-policing/stanford-open-policing-project-california

- Data Size: 2.32 GB
- 22 features+1 label
- 14,536,338 observations
- Covers all the stop data of California in 2013

Data Pipeline

Store data in AWS S3 Bucket

Reading from S3, preprocess data using Pyspark on AWS EMR

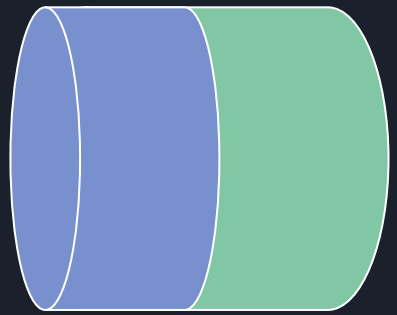
Machine Learning on AWS EMR

Write processed data parquet ready for model fitting back to S3

Data Preprocessing Algorithms

Categorical Columns

location, gender, race, violation. search type, contraband_found, is arrested



StringIndexer
encodes a string column of labels to a column of label indices

OneHotEncoder
maps a categorical feature to a binary vector

Dataframe Ready for Model Fitting

ML Results

Random Forest	not arrested (prediction)	arrested (prediction)
not arrested (actual)	2830389	2298
arrested (actual)	18187	55275
auc score	0.941	

XGBoost	not arrested (prediction)	arrested (prediction)
not arrested (actual)	2830225	2462
arrested (actual)	17885	55577
auc score	0.95	

Deep Learning	not arrested (prediction)	arrested (prediction)
not arrested (actual)	2827831	4856
arrested (actual)	19812	53650
auc score	0.919	

Gradient Boosting	not arrested (prediction)	arrested (prediction)
not arrested (actual)	2830140	2547
arrested (actual)	17921	55541
auc score	0.945	

Lesson Learned

- Spark SQL
  - More convenient for writing queries
- Loading/Writing data using Spark SQL
  - From/To local machine or AWS S3
- Spark ML
  - Apply ML algorithms: Decision Tree, Random Forest, etc.
  - Feature engineering: Impute missing value, OHE, VectorAssembler, etc.
- H2O and Sparkling Water
  - Create H2OFrame
  - Apply algorithms: Deep Learning, AutoML, Stacked Ensemble, etc.
- Run code and Sparkling Water on EMR

Lesson Learned

- Collaboration
  - Get access to shared data through AWS S3 and EMR
- ML thinking style
  - Process data
  - ML algorithms
  - Metrics
- Helpful for Practicum!!!
  - UCSF database
  - Use Pyspark to query data and store data
  - Use Spark ML / deep learning later

Thank you so much!!!

Look forward to seeing you in May for Product Analytics 🎉