Capstone: Project Proposal

Zeph Grunschlag

1. Problem Statement

Recently, the IEEE=CIS sponsored a Fraud Detection Kaggle competition. I am planning to use this competition as a basis for my project, but to go beyond the competition in several regards. I plan the following enhancements of the competition:

- a more realistic evaluation methodology than the one defined by Kaggle
- provide a more production realistic data-flow
- analyze different criteria and approaches
- allow for interpretability of the predictions

In particular, false positives/negatives should each have an associated cost and the hyper-parameters tuned to minimize overall cost. As fraud prevention often requires explainability/interpretability for regulatory/compliance reasons, each method should afford a way of interpreting the result so we can give a succinct, human understandable, and correct reason for why a particular observation was predicted as fraud or not.

Additional tasks - time allowing

I would also like to:

- set up a realistic load testing of the system inspired by PaySim approach to transactional data generation
- add a model drift service that monitors the reliability of the model in real time
- investigate an online version of the algorithm that adjusts the model based on recent observations

2. Dataset

I will use the data provided by Kaggle (on the order of 500K rows with 400 features per row and 1.3 GB total). I might also generate some artificial data, especially in the case that I explore the online algorithm approach.

3. Approach to solution

As most Kaggle competitions and as is typical of fraud detection, this is a supervised classification problem. However, it also makes sense to evaluate underlying regression approaches with a numeric threshold that determines whether to consider an observation as fraudulent or not. In this case, observations are financial transactions which the model must predict whether will end up causing fraud. I will evaluate a variety of methods, including deep learning.

4. Deliverables

Deliverables will consist of

- evaluation of the various approaches used for accuracy, cost minimization, interpretability, and possibly resource consumption
- a blog post describing the system and my findings and serving as a tutorial on how to recreate the system
- a public github repo allowing anyone to reproduce my system and findings
- an API with the following endpoints for at least one algorithm investigated, and hopefully for several, probably including XGBoost and DeepLearning:
    - Given an observation, return the prediction (Fraud / Not Fraud / Monitor)
    - Given an observation, return the prediction, together with an interpretation explaining the prediction. EG: "Monitor" because feature #17 > 210

5. Resources

I would like this to represent a relatively realistic system. Thus I could see having the following components (all on AWS):

- Postgres RDS A non-RDMS system such as Redshift, BigQuery, Snowflake, or self-maintained Hive/HDFS cluster containing the original data as well as an audit trail for each prediction that occured along with its interpretation. *For example, if Redshift is chosen a single Redshift instance should suffice, but further investigation is required to the size of that instance.*
- EC2 instance (or K8s cluster?) with Airflow using Spark/Dask jobs to process the data and create the model. *In the case of EC2, a medium instance should suffice.*
- ECS or SageMaker instance to handle the prediction and interpretation API. *A small deployment should suffice, unless I end up simulating a relatively large load.*
- A CI/CD system that with a click of a button (or buttons) does the following:
    - deploys a container to prepare the data and generate a model

- deploys a container running the generated model on *staging*
- deploys a container to run basic tests on the *staging* environment
- deploys a container running the generated model on *production*
- deploys a container to run basic tests on the *production* environment

Notes

- Kaggle competition
  - IEEE-CIS Fraud Detection
  - The evaluation metric (AU ROC)
  - Great starter notebook by Andrew Lukyanenko
- Interpretability - there has been some recent work to provide interpretability even for seeming black box approaches such as XGBoost and Deep Learning
  - Papers with code: A Unified Approach to Interpreting Model Predictions
  - Paper
  - Repo

Versions

This is a work in progress. Here are snapshots of previous versions:

1. 11/24/2019 (*link to be added shortly*)