Team 1 project documentation

Team Leader: Abdelrahman Ragab

Team: Dina Elsokary – Ahmed Nasser – Ahmed Talaat

Trainer: Jezz martin

Project idea:

- **Fraud detection** - Identify fraudulent activities using Amazon SageMaker.

Problem description:

- You work for a multinational bank.
- Over the last few months, there has been a significant uptick in the number of customers experiencing credit card fraud.
- You need to use ML to identify fraudulent credit card transactions before they have a larger impact on your company.

What is the Problem?

Predicting Credit Card Fraud

Introduction to business scenario

You work for a multinational bank. There has been a significant increase in the number of customers experiencing credit card fraud over the last few months. A major news outlet even recently published a story about the credit card fraud you and other banks are experiencing.

As a response to this situation, you have been tasked to solve part of this problem by leveraging machine learning to identify fraudulent credit card transactions before they have a larger impact on your company. You have been given access to a dataset of past credit card transactions, which you can use to train a machine learning model to predict if transactions are fraudulent or not.

About this dataset

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred over the course of two days and includes examples of both fraudulent and legitimate transactions.

Features

The dataset contains over 30 numerical features, most of which have undergone principal component analysis (PCA) transformations because of personal privacy issues with the data. The only features that have not been transformed with PCA are 'Time' and 'Amount'. The feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction amount. 'Class' is the response or target variable, and it takes a value of '1' in cases of fraud and '0' otherwise.

Features:

`V1, V2, ... V28`: Principal components obtained with PCA

Non-PCA features:

- \`Time\`: Seconds elapsed between each transaction and the first transaction in the dataset, \$T_x - t_0\$

- \`Amount\`: Transaction amount; this feature can be used for example-dependent cost-sensitive learning

- \`Class\`: Target variable where \`Fraud = 1\` and \`Not Fraud = 0\`

Read through a business scenario and:

1. Determine if and why ML is an appropriate solution to deploy.

- ML is appropriate solution because of the scale , variety and speed required

2. Formulate the business problem, success metrics, and desired ML output.

- The problem is fraud transactions which affects the customer and the bank.
- the success metric is automating the detection of potentially fraudulent activity and flagging that activity for review.

3. Identify the type of ML problem you are dealing with.

Supervised Machine learning (xgboost algorithm)

4. Analyze the appropriateness of the data you're working with.

Yes, because it contains the pervious fraud transaction information

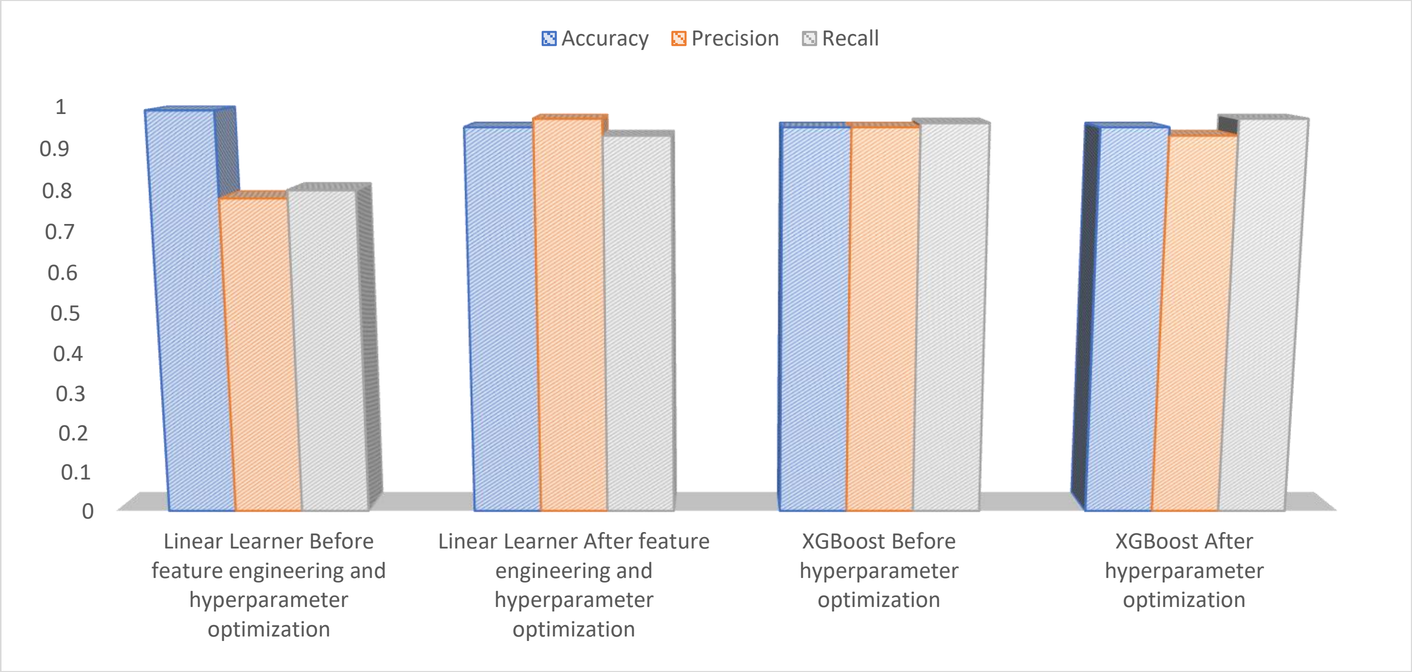
Steps to make the solution:

- Step 1: Problem formulation and data collection
  - ✓ Formulate the business problem, success metrics, and desired ML output
  - ✓ Identify the type of ML problem
  - ✓ Analyze the appropriateness of the data
- Step 2: Data preprocessing and visualization
  - ✓ First, import the necessary libraries and read the data into a Pandas dataframe. After that, explore the data. Look for the shape of the dataset and explore the columns and the types of columns (numerical, categorical). Consider performing basic statistics on the features to get a sense of feature means and ranges. Take a close look at the target column and determine its distribution.
  - ✓ Look specifically at the distribution of features like Amount and Time and calculate the linear correlations between the features in the dataset.
- Step 3: Model training and evaluation
  - ✓ Split the data into train_data, validation_data, and test_data (80%,10%,10%)
  - ✓ Convert the dataset to an appropriate file format that the Amazon SageMaker training job can use.
  - ✓ Upload the data to your Amazon S3 bucket.
  - ✓ For the first try we will use linear estimator with predictor type: binary classifier. Then trying XGBoost algorithm.
  - ✓ Evaluation metrics (Accuracy, Precision, Recall)
- Step 4: Feature engineering
  - ✓ Choosing a method to deal with this imbalanced dataset
  - ✓ Feature reduction

- Hyperparameter optimization
  - ✓ Launching an Amazon SageMaker hyperparameter tuning job and viewing the evaluation metrics
- Hosting (Deployment)

Results:

		Accuracy	*Precision*	*Recall*
Linear learner	Before feature engineering and hyperparameter optimization	0.99	0.78	0.80
	After feature engineering and hyperparameter optimization	0.95	0.97	0.93
XGBoost	Before hyperparameter optimization	0.95	0.95	0.96
	After hyperparameter optimization	0.95	0.93	0.97


▨ Accuracy
▨ Precision
▨ Recall
Linear Learner Before feature engineering and hyperparameter optimization
Linear Learner After feature engineering and hyperparameter optimization
XGBoost Before hyperparameter optimization
XGBoost After hyperparameter optimization