Detecting Fraud in a Mobile Money Platform

Supervised Machine Learning Capstone

Colette Gabriel

Mobile Payment Growth

- ☐ **In-store mobile payments expected to reach $503 billion by 2020** (BusinessInsider)
- ☐ **2.1 billion mobile wallet users in 2019** (BusinessWire)
- ☐ **Fraud is highest concern for the financial industry, followed by processing speed** (TD Bank)
- ☐ **Estimated losses from online payment fraud will top $22 billion in 2019** (Juniper Research)

Project Details

- [ ] Six million+ transactions in dataset
- [ ] Data modeled after mobile banking transactions but randomized for anonymity
- [ ] Objective: Build a machine learning model that is both effective and efficient at predicting fraudulent transactions while minimizing false positives

- [ ] Data includes:
  - [ ] Transaction type - payment, transfer, withdrawal
  - [ ] Transaction amount
  - [ ] Customer ID - originator of the transaction
  - [ ] Customer initial balance
  - [ ] Customer balance after transaction
  - [ ] Recipient ID - relevant for payments and transfers
  - [ ] Recipient initial balance
  - [ ] Recipient balance after transaction

- Two columns for fraud detection:
  - 1. Whether or not a transaction was determined to be fraud
  - 2. Whether the transaction was flagged as illegal, defined by an attempt to transfer more than a preset amount (approximately $200 USD*)

*NOTE: Currency displayed in dollars here, but the original data simulation was based on an Somalian currency

Data Preparation

- [ ] Dropped columns capturing unique account names for originating and destination transfers
- [ ] These 2 fields contained more than 10 million unique names
- [ ] Assumption: Without immense processing power, the amount of information gained from encoding this data would not be equal to the exponential gain in processing time

- [ ] Used one hot encoding for ‘type’, which captures the five types of financial transactions (cash withdrawal, debit, payment, transfer, cash deposit)

- [ ] Examined correlations and found a strong correlation between two groups of columns:

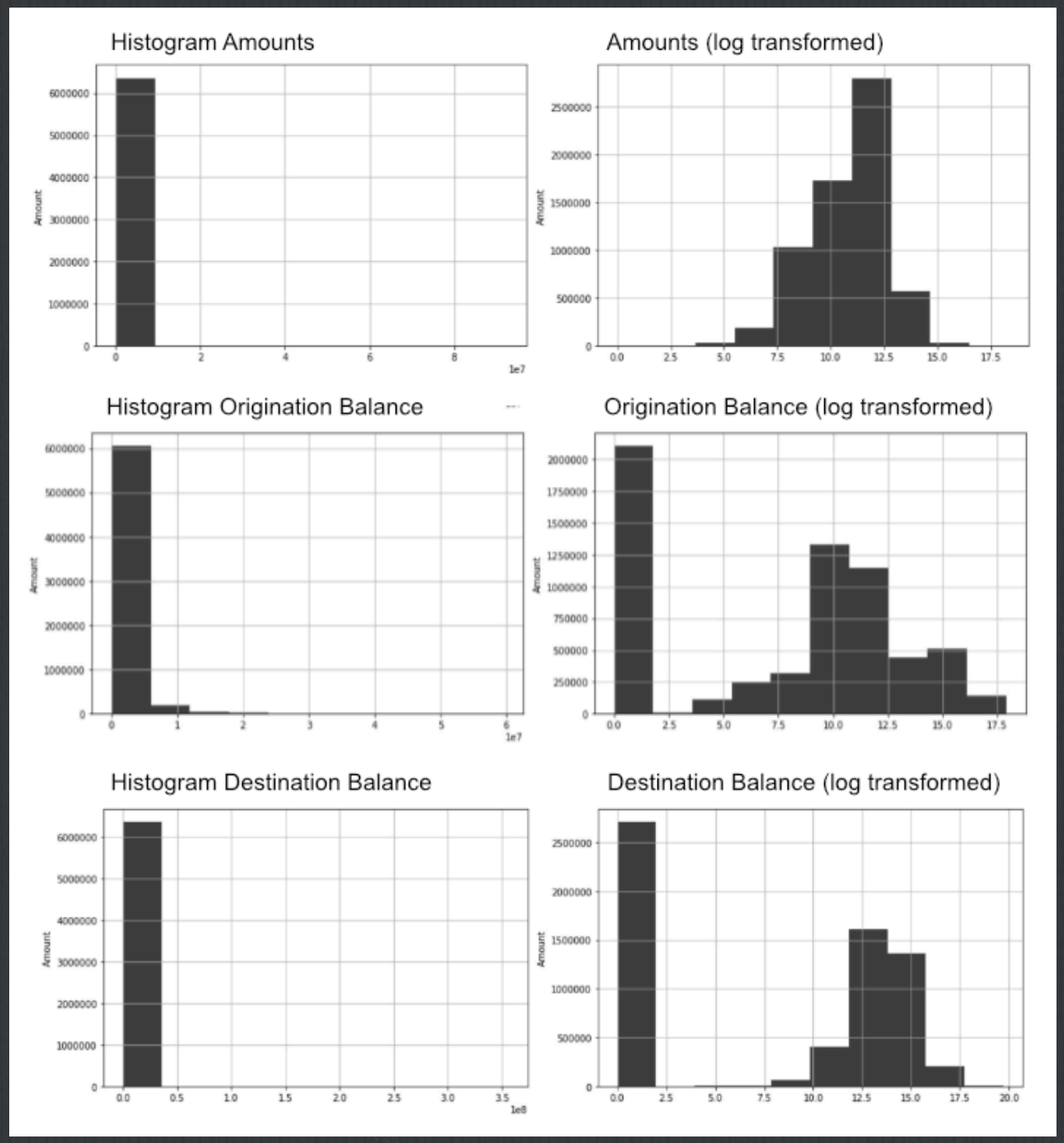
    1. old and new origination balance

    2. old and new destination balance

- [ ] Removed the ‘new’ balance for both origination and destination and kept only the original (pre-transaction) balances

- [ ] Assumption: There was a high correlation between these two sets of data. Keeping both old and new data points would prejudice the model and skew results

Dealing with Outliers

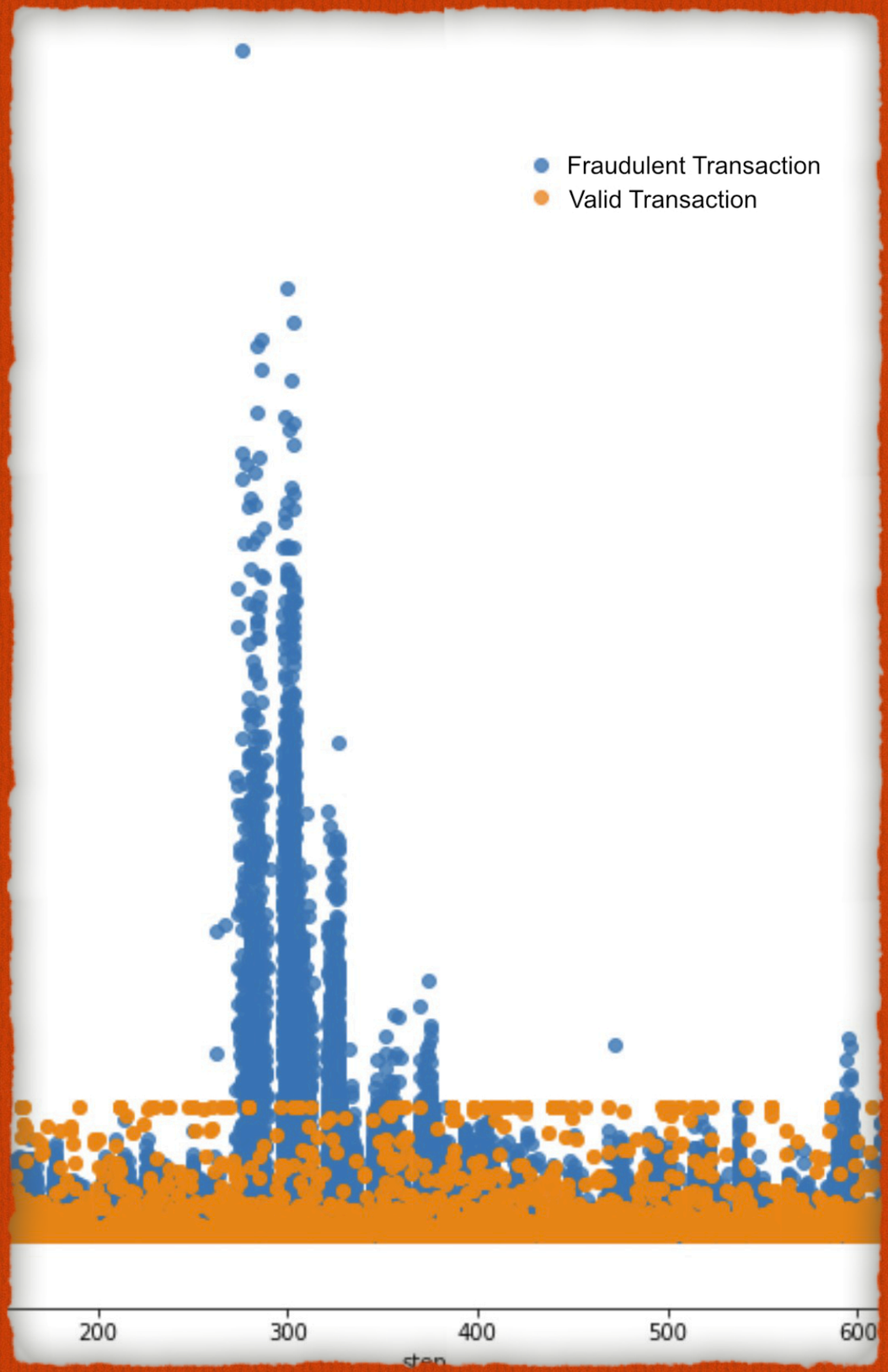


Currency Variables transformed by

np.log(df.amount+1)

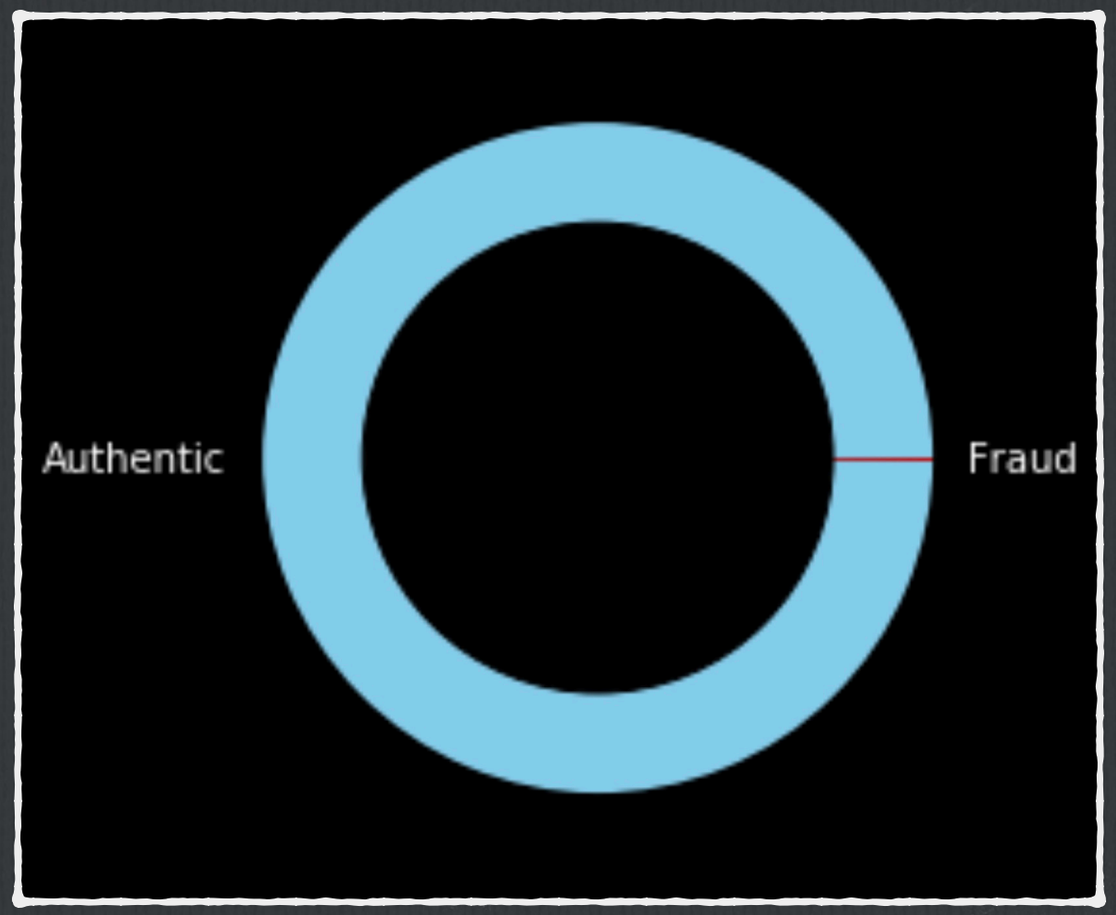
Amount
Origination Balance
Destination Balance

Dealing with Imbalanced Data



Imbalanced Data Sets

Class imbalance problems are found in multiple areas such as telecommunication, fraud detection, and medical diagnosis



- For millions of credit card transactions processed daily only a small fraction are fraudulent
- In this dataset, the fraud rate was 0.129%, or just 8,219 out of the 6.3 million transactions represented in the dataset

Methods for Dealing with Imbalanced Data

- Oversampling - artificially add minority transactions so that the number of samples in training set balances
    - pros - higher sample rate
    - cons - can add noise to the dataset
- Undersampling - use all minority samples and randomly select an equal number of majority class samples to create a balanced dataset
    - pros - uses only authentic fraud transactions
    - cons - smaller sample sizes

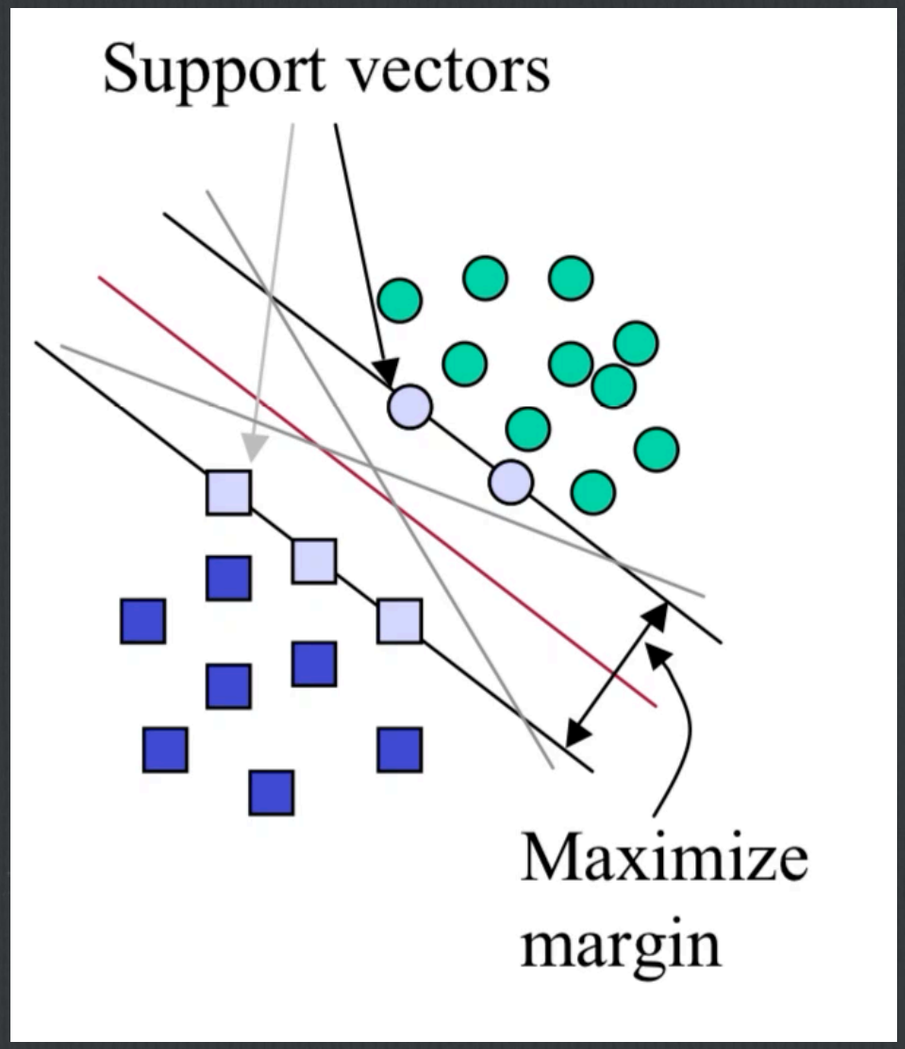
Methodology

Under Sampling Methodology

- Created 25 small balanced datasets of 13,500 each
- Each dataset consists of:
  - Consistent sample of 80% of all fraud transactions (n=6,570)
  - Random sample of an equal number of valid transactions

SVM - Support Vector Machine

- Very flexible, efficient at classification
- Chooses decision boundary that maximizes distance from nearest data points of all classes
- Support vectors are the data points closest to the hyperplane and are most difficult to classify
- Works by finding the optimum hyperplane in an N-dimensional space



Graphic by R. Berwick, MIT

SVM - Cost Function

- Cost Function (C)- two goals to balance

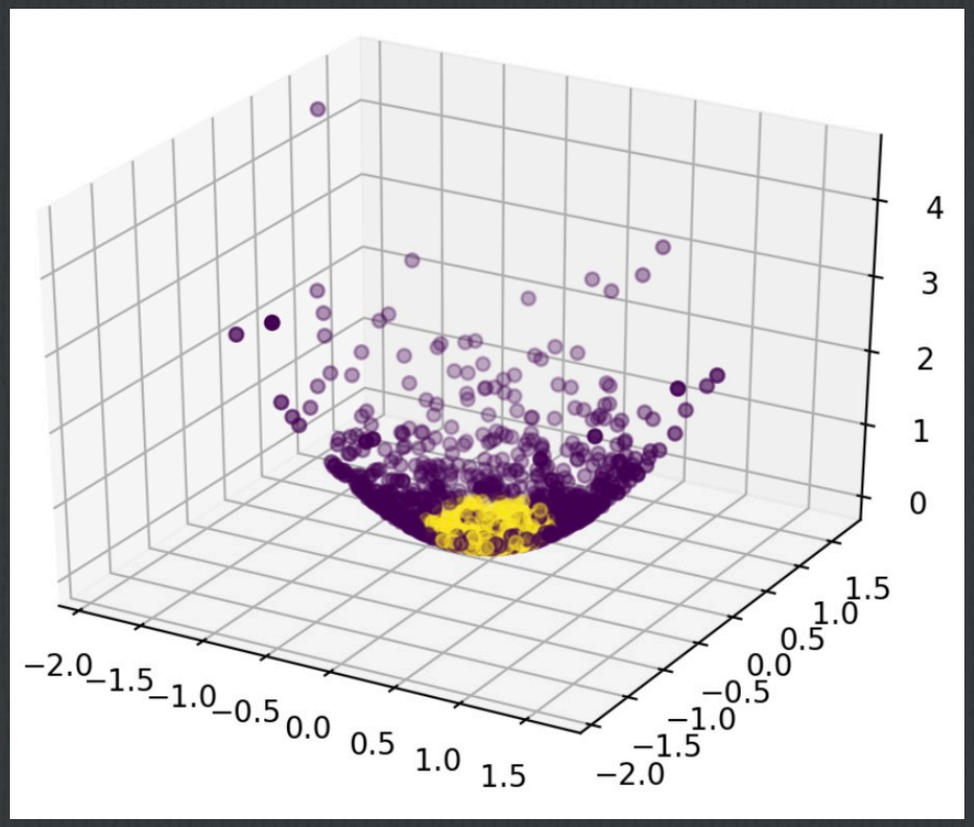
1. maximize margin between nearest points of each class to the boundary

2. minimize cumulative distance of data points that are on the wrong side of the margin from the boundary (soft margin)

SVM - Kernel Choice

- Kernel - the set of mathematical functions that transform the data to define the optimal margin

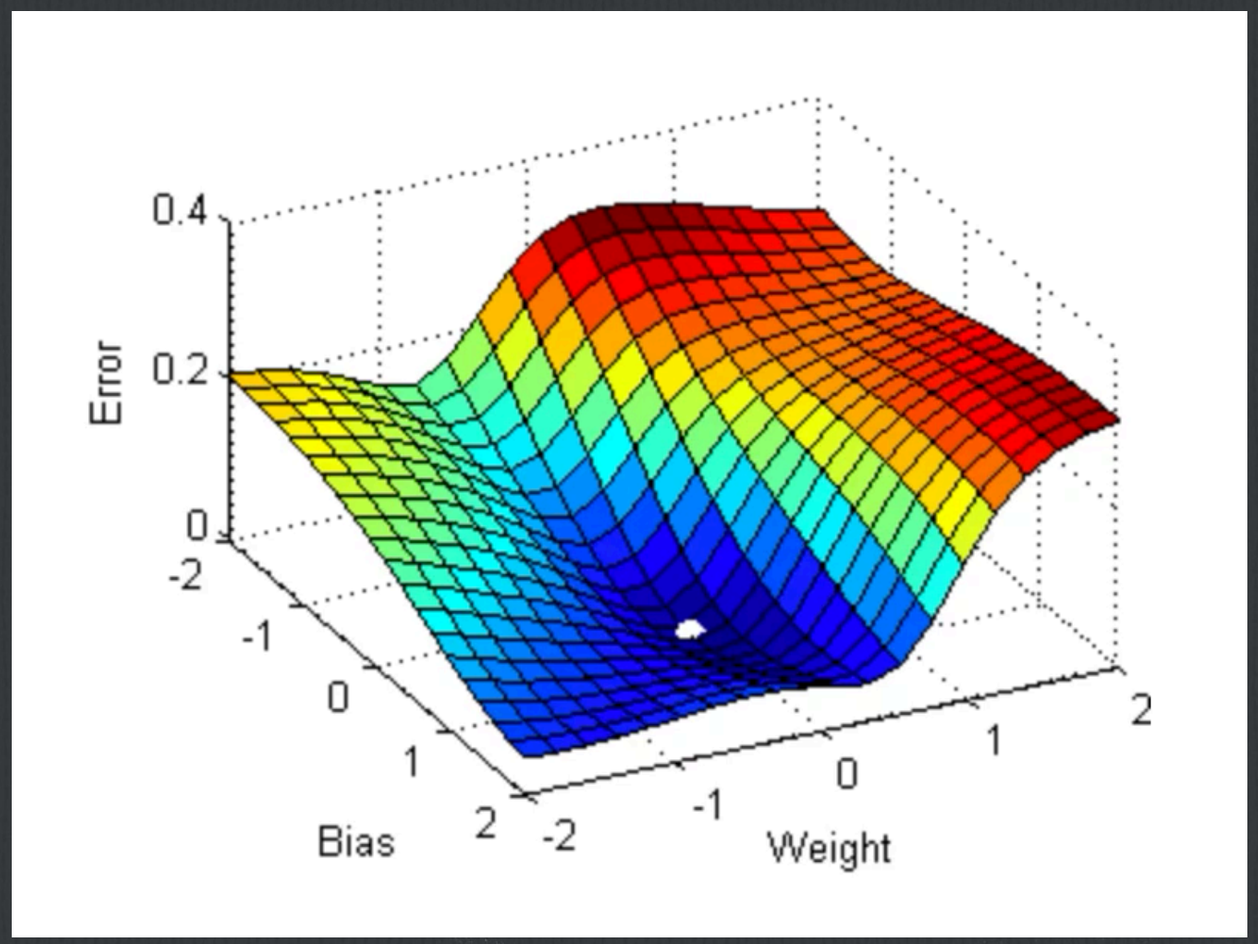
- **Kernels explored:**
  - linear
  - sigmoid
  - radial basis function (rbf)



Graphic from Thinkful

Gradient Descent

Gradient Descent



- Optimization algorithm to find coefficients of a function that minimize cost function
- Standard (batch) Gradient Descent - calculates gradients for the whole dataset to perform just one update

Graphic by H. Avjyan, KBR, Inc.

Stochastic Gradient Descent

- Stochastic Gradient Descent uses a single data point per iteration (randomly chosen) to approximate the true cost gradient

- pros - vastly increases efficiency

- cons - results can be noisy (can be overcome by tweaking hyperparameters)

SGDClassifier

- SGD is actually an optimization method, not a machine learning algorithm
- Scikit-learn SGDClassifier = linear classifier optimized by SGD
- Alpha = 1/C, regularization parameter to reduce overfitting
- Number of iterations - affects the learning rate, SGDClassifier can run slower and overfit with too many iterations

GridSearchCV

- Scikit-learn tool to automate what would be a manual search through multiple parameters to find the best tuned algorithm

- SVC parameter grid (kernel = rbf):

  {'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001, 0.00001, 10]}

- SGDClassifier parameter grid:

  {"n_iter_no_change": [1, 5, 10], "alpha": [0.0001, 0.001, 0.01, 0.1, 1, 10, 100], "penalty": ["none", "l1", "l2"]}

Results

Measuring Performance

- Accuracy can be a useful measure of how well a model performs, but is NOT effective when a dataset is severely imbalanced
- With a fraud rate of 0.129%, the model could do nothing, predict all transactions as valid, and have an accuracy rate of 99.87%
- Precision and Recall are better predictors when dealing with highly imbalanced datasets

Components to Evaluate

- ☐ True Positive = accurately predicted fraud transaction
- ☐ True Negative = accurately predicted valid transaction
- ☐ False Positive = predicting fraud for a valid transaction
- ☐ False Negative = mislabeling a fraudulent transaction

Precision vs Recall

- Precision = measures exactness of the classifier, the percentage of fraud accurately detected by the model
  - Precision = true positive / (true positive + false positives)
  - Low precision means irrelevant results and inconvenience to customers — increased false positives
- Recall = measures completeness of the classifier, the percentage of total fraud transactions correctly classified
  - Recall = true positive / (true positive + false negative)
  - Low recall means false negatives, so fraud will go undetected

F1 Score

- F1 Score = is an overall accuracy measure for classifiers that describes the balance between precision and recall
  - F1 Score = 2 * (precision * recall) / (precision + recall)
  - Results between 0 (total failure) and 1 (perfect accuracy)

SVM - Results

SVM Parameters	Recall	Precision	F1
svm.SVC(gamma='scale', decision_function_shape='ovo')	99.62	91.96	95.63
SVC(C=100, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)	99.55	95.99	97.74

SGDClassifier - Results

SGD Parameters	Recall	Precision	F1
SGDClassifier(tol=1e-3, shuffle=True)	99.62	91.47	95.37
SGDClassifier(alpha=0.0001, average=False, class_weight=None, early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True, l1_ratio=0.15, learning_rate='optimal', loss='hinge', max_iter=1000, n_iter_no_change=5, n_jobs=None, penalty='none', power_t=0.5, random_state=None, shuffle=True, tol=0.001, validation_fraction=0.1, verbose=0, warm_start=False)	89.28	76.9	82.04

Potential Next Steps

- ☐ High accuracy and recall still leaves a lot to be desired for millions of transactions
- ☐ Around 440 false negatives in the 6 million data points here
- ☐ Need to look at ways to include more data, incorporate the names of transactors into the dataset without slowing down processing