Data and Feature Engineering Challenges in Machine Learning

Kendall E. Nygard, Mostofa Ahsan, Aakanksha Rastogi, Rashmi Satyal

Proceedings of The Sixteenth International Conference on Advanced Engineering Computing and Applications in Sciences

Valencia, Spain

Presenter, Dr. Kendall E. Nygard, Director, Dakota Digital Academy, Emeritus Professor, North Dakota State University, kendall.Nygard@ndus.edu

DAKOTA DIGITAL ACADEMY
NORTH DAKOTA UNIVERSITY SYSTEM

NDSU NORTH DAKOTA STATE UNIVERSITY

IARIA

The Authors

Dr. Kendall E. Nygard is Director of the Dakota Digital Academy of the North Dakota University System and Distinguished Emeritus Professor of Computer Science at North Dakota State University (NDSU). He is an IARIA Fellow. His Ph. D. is from Virginia Tech University.

Dr. Mostofa Ahsan is a Senior Data Scientist at Highmark Health. He holds an NDSU Ph. D. degree in Computer Science and an MBA.

Dr. Aakanksha Rastogi is a Senior Software Engineer at Medtronic. She holds an NDSU Ph. D. degree in Software Engineering.

Rashmi Satyal is a Technical Consultant at Logic Information Systems. She is a Master of Science graduate of NDSU in Software Engineering.

NDSU NORTH DAKOTA STATE UNIVERSITY

Workgroup Research Interests

- Machine Learning and Data Science
- Cybersecurity
- Software Design and Development
- Software Security
- Ecommerce and Business Analytics
- Applications to Smart Grid, Embedded Devices, Health and Medical Systems and Products

Outline

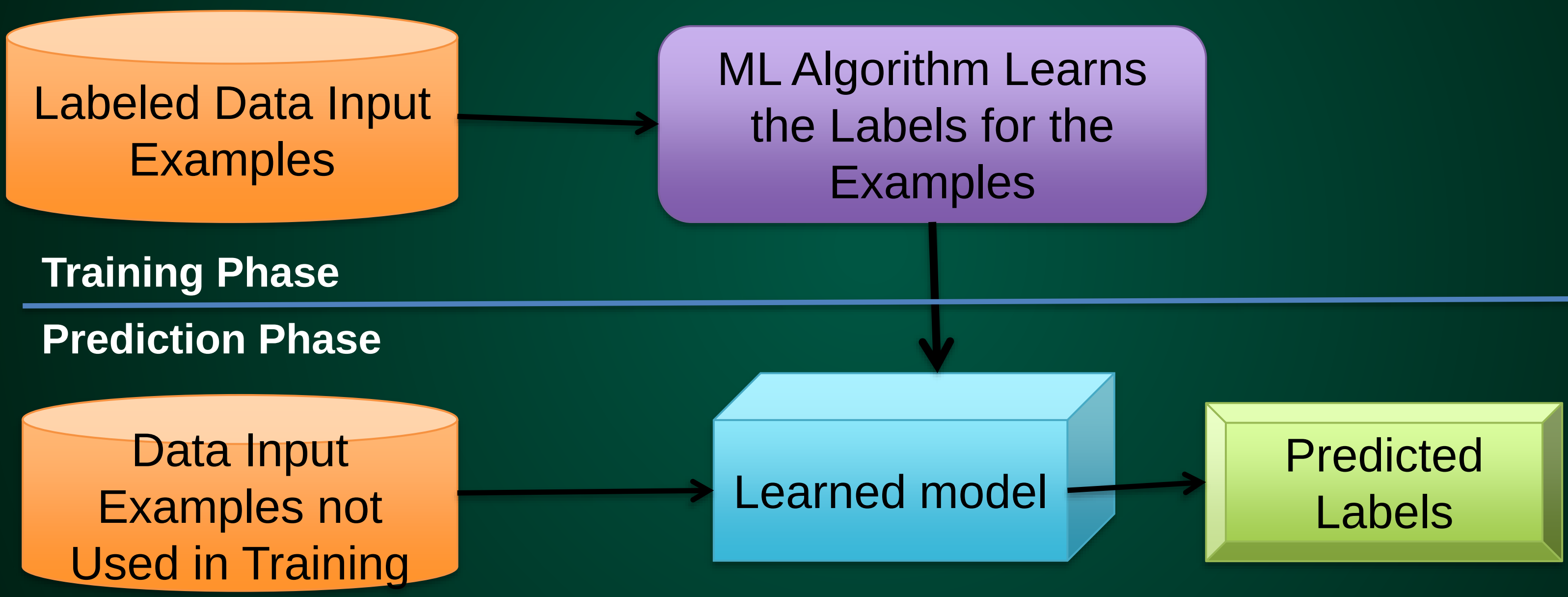
- Supervised Machine Learning (ML)
  - Performance Measures
- Self-Driving Cars Application
  - Attributes
  - ML Methodologies Utilized
- Intrusion Detection Applications
  - Attributes
  - ML Methodologies Utilized
- Data Processing and Feature Engineering
  - Methodologies
  - Illustrations
- Conclusions

Supervised Machine Learning

- Machine learning (ML) is the application of artificial intelligence to make systems capable of learning from problem-specific training data for analytical model building and solving related tasks
- Supervised ML is used for applications where a training data set with known labels or properties can establish a model of ground-truth that is then used on new data as a classifier to make predictions

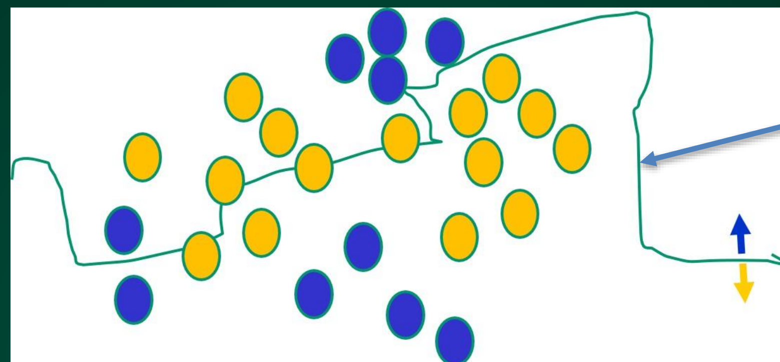
Machine Learning

Supervised Machine Learning Flow



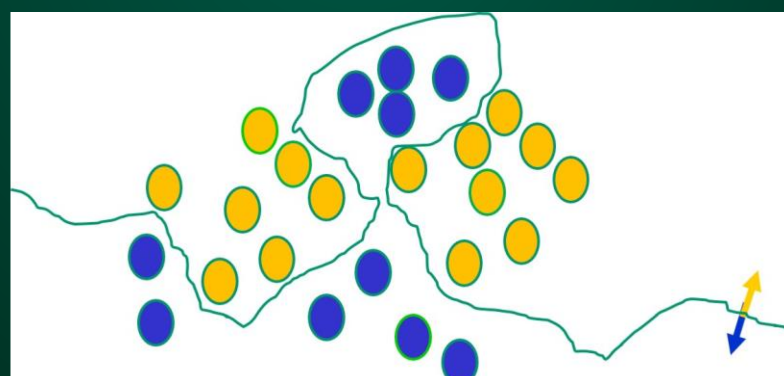
Training a Neural Network ML Model

Random Initial Parameters



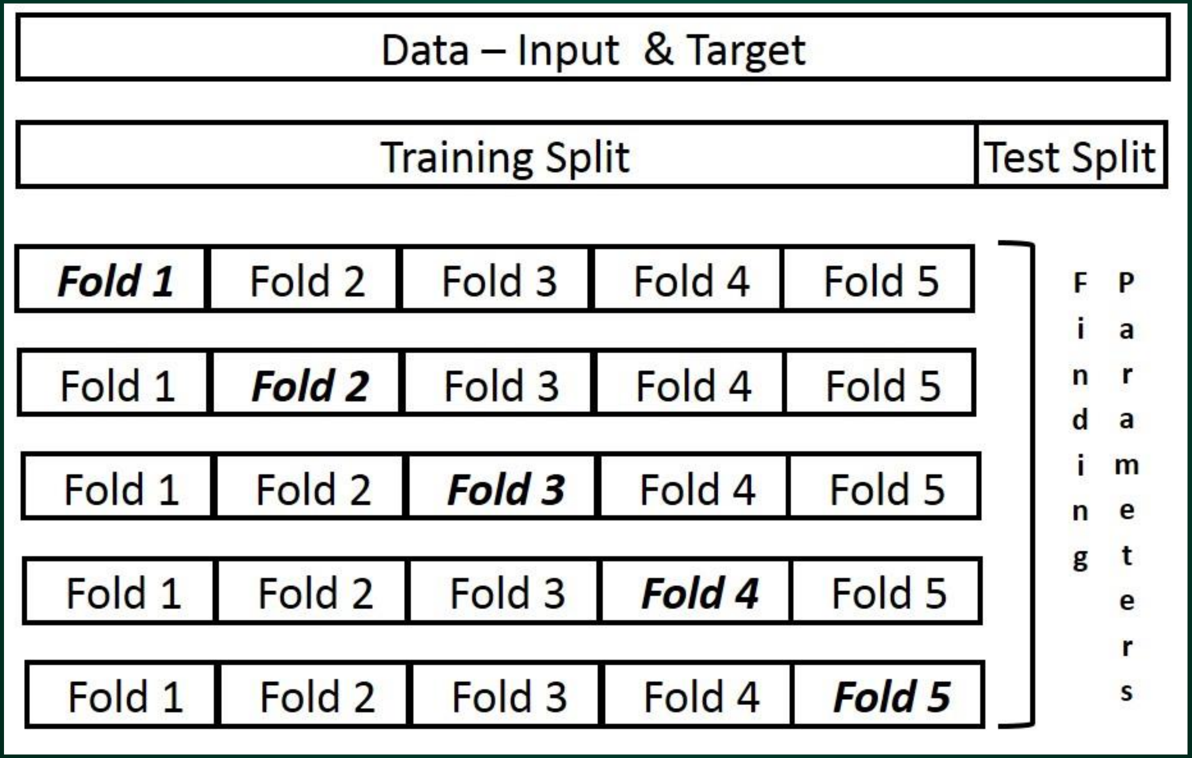
Prediction
Boundary

After Training, Adjusted Parameters



Cross Validation in Supervised Learning

The Idea: Split the data with known labels into a training split and a test split, apply the ML algorithm to the training split until it has learned, then apply it to the test split



Intrusion Detection - The Classification of an Input Vector as an Attack or Not an Attack

Experiments with the famous NSL-KDD data vectors with labels, 125,973 records, 43 features, 4 attack classes

ML predictions of input vectors

Normal = not an attack
Denial of service = attack or not, and attack type
Probe = attack or not, and attack type
Remote to local (R2L) = attack or not, and attack type
User to root (U2R) = attack or not, and attack type

Attributes in the NSL-KDD Data Vectors

Feature Type	Feature Names
Basic	Duration, Protocol_type, Service, Flag, Src_bytes, Dst_bytes, Land, Wrong_fragment, Urgent
Content related	Hot, Num_failed_logins, Logged_in, Num_compromised, Root_shell, Su_attempted, Num_root, Num_file_creations, Num_shells, Num_access_files, Num_outbound_cmds, Is_hot_login, Is_guest_login
Time related	Count, Srv_count, Serror_rate, Srv_serror_rate, Rerror_rate, Srv_rerror_rate, Same_srv_rate, Diff_srv_rate, Srv_diff_host_rate
Host based traffic	Dst_host_count, Dst_host_srv_count, Dst_host_same_srv_rate, Dst_host_diff_srv_rate, Dst_host_same_src_port_rate, Dst_host_srv_diff_host_rate, Dst_host_serror_rate, Dst_host_srv_serror_rate, Dst_host_rerror_rate, Dst_host_srv_rerror_rate

NDS

ML Performance Measures with Attack Examples

TP = True Positive = Correct prediction of an attack

TN = True Negative = Correct prediction of not an attack

FN = False Negative = Incorrect prediction of not an attack

FP = False Positive = Incorrect prediction of an attack

Accuracy = (TP+TN)/(TP+TN+FP+FN) = % of reports that are correct

Precision = TP/(TP+FP) = % of vectors reported as an attack that actually are an attack

Recall = TP/(TP+FN) = % of vectors that are attacks that do get reported as an attack

Alert!! False negatives are deadly in intrusion detection! That is because a true attack slips through! So a high Recall measure is vital!

Self-driving Car Collisions

Date	Place	Time	Vehicle Hit or Ran Into	Flashing Lights	Bright Objects
01/22/2018	Culver City, CA	11AM	Parked Fire truck	Y	-
05/20/2018	Laguna Beach, CA	11AM	Police SUV	Y	-
12/07/2019	Norwalk, CT	4AM	Parked Cruiser + Disabled car (Chain Collision)	Y	Flares
12/29/2019	Cloverdale, IN	8AM	Parked Fire truck (earlier crash scene)	Y	-
01/22/2020	West Bridgewater, MA	10PM	State Trooper SUV + Car (Chain Collision)	Y	Illuminated Arrow Board
07/30/2020	Cochise County, AZ	4AM	Highway Patrol + Ambulance (Chain Collision)	Y	-
08/26/2020	Spring Hope, NC	12AM	Deputy's Cruiser + State Trooper Vehicle (Chain Collision)	Y	-

https://www.skynettoday.com/briefs/tesla-investigations

https://abcnews.go.com/US/tesla-autopilot-crashes-parked-police-car/story?id=55525536

ML Study: Under What Conditions do Self-Driving Cars Collisions Occur?

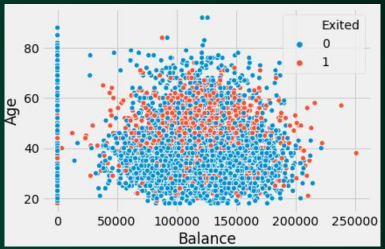
- The Data: Collision reports with 140 attributes concerning weather, movements of the vehicles, type of collision, time of day, other vehicle type, injury type, vehicle damage, etc.
- A linear sequential supervised learning Artificial Neural Network ML model called NoTrust was devised, validated, and tested to classify the target data in terms of trust or lack of trust, risk, and safety
- Python Keras libraries with TensorFlow backend were utilized
- Feature Engineering – Many combinations of the attributes were systematically run and evaluated to determine the highly important ones to retain and eliminate those that are redundant
- A relatively small feature set performed well in making accurate trust and do not trust predictions

Under What Conditions do Self-Driving Cars Take Inappropriate Actions that Can Cause a Collision?

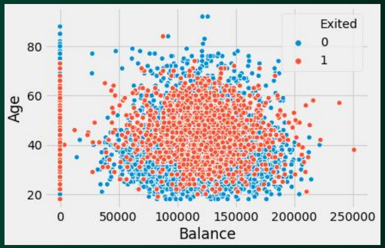
- Anti-autonomy = Actions of the self driving car that decrease trust, increase risk, and reduce safety
- An expanded model that includes anti-autonomous traits of the self-driving car and measures of severity of damages resulted in a need to add more attributes, such as unusual weather and types of obstacles
- Adding more attributes and predictors induced data imbalance and overfitting, decreased quality of predictions and added more noise and redundancies
- Both the linear sequential ANN and Recurrent Neural Networks (RNN) with Long Short Term Memory (LSTM) were applied
- A relatively small feature set performed well in making accurate trust and do not trust predictions

The Curse of Imbalanced Data

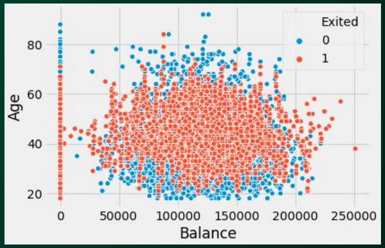
- The blue outnumber the orange in the original data



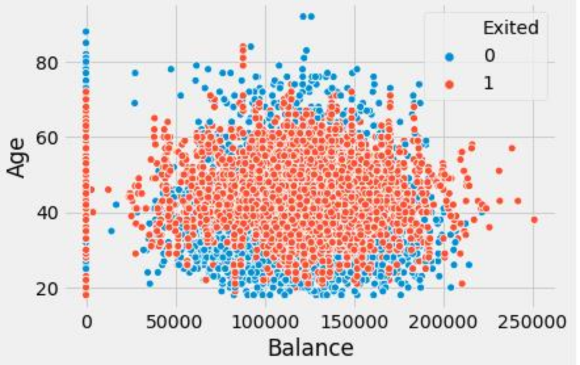
- Oversample the orange class. Overfitting easily occurs



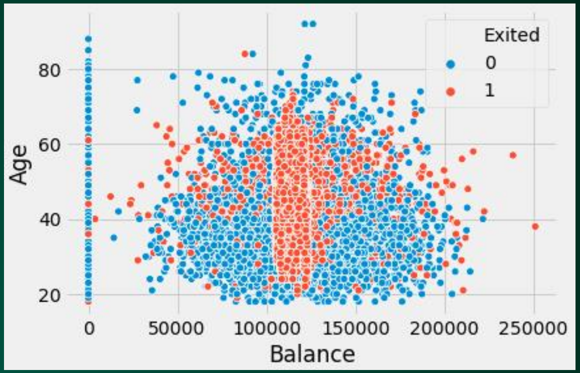
- Synthetic Minority Oversampling Technique (SMOTE). Oversample the orange class by statistically combining members to create hybrids. K-nearest neighbors can be used



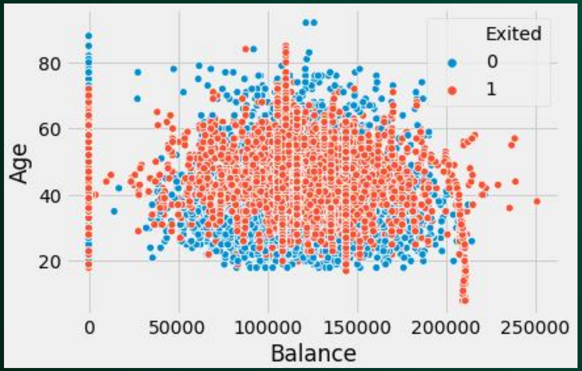
The Curse of Imbalanced Data (Continued)



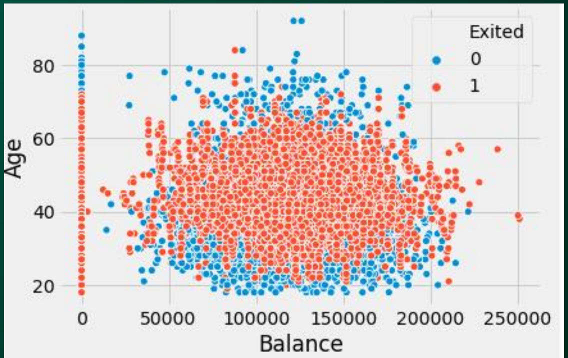
Borderline-SMOTE



K-Means SMOTE.



SVM SMOTE



ADASYN

Feature Engineering -

- Feature engineering is the process of working with input data to select or create features that are computationally efficient and have excellent predictive performance
- Categorical data for self-driving car collisions include presence or absence of a specific weather condition, type of car, specific driving action, or device failure, etc.
- Categorical data for intrusion detection include presence of file access attempts, range of error rate, or protocol type, etc.
- Numerous methods to map categorial data into values that can be used in an ML model
  - 1-hot encoding. Use a 0-1 binary variable for presence or absence of an element of a category
  - Hash encoding. Use a hashing function to map categories into a predetermined range of integer values
  - Base-N encoding. Convert the categories into arrays of their Base N representation

Feature Engineering Methods

- Filters. Pair each feature with a categorical output, such as whether a vector is an specific attack or not. Then apply a statistical test such ANOVA to measure the importance of the feature in the prediction, providing a way to filter out features of low importance
- Wrappers. Evaluate candidate subsets of features by running the ML model on just the subset, to provide a means of eliminating low-ranking combinations. Combinatorial explosion can happen (n!/(n-r)*r! easily gets huge). To limit the number of subsets evaluated, a heuristic such as simulated annealing or tabu search is applied.
- Embedding. Mathematical calculate information within the search space as the procedure evolves to produce scores for features that can be used to rank the relative importance of features and provide a means of eliminating some of them.

Conclusions

- Machine learning methods were used in two applications
  - Predicting risk and trust concerning collisions incurred by self-driving cars
  - Predicting whether input vectors from the internet are attacks of known types
- Performance metrics were presented
- Methods for processing imbalanced input data were described
- Methods of feature engineering were described