CLAIMS FRAUD MODEL

OPERATIONAL REVIEW – FINAL MODEL

OCTOBER 2020

CONTENTS

EXHIBIT INTRO - EXECUTIVE SUMMARY

Regis University practicum course requested a project which applied techniques introduced throughout the degree program. Today's conversation will review the candidate model produced.

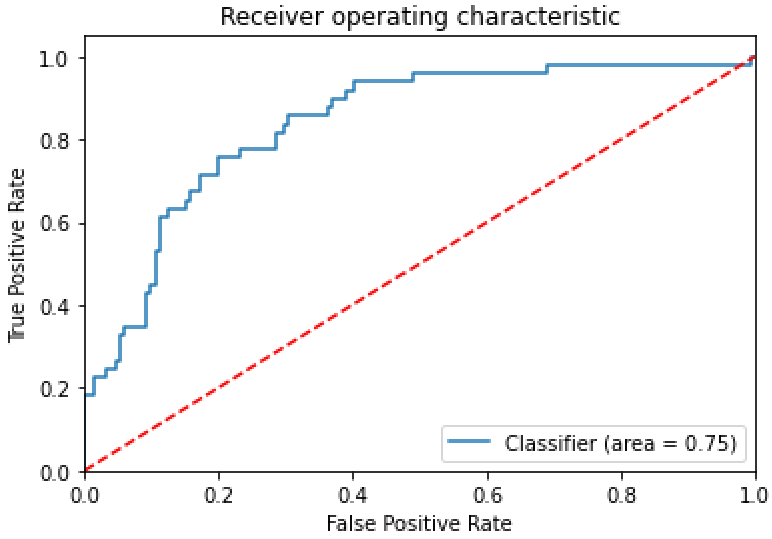
Objectives:

- Provide an overview of the modeling processes and methodologies
- Review the Claims Fraud Model final model performance and prediction explanation
- Discuss unique machine learning techniques used
- Review applicable next steps

Highlights:

- Improved method for database interaction and table creation with Pandas
- The ability to explain the prediction and put it into an actionable insight
- Feature Synthesis, Outlier correction, and Automated Pipeline
- Feature reduction to remove unnecessary data and improve overall performance

EXHIBIT INTRO – Gini (ROC AUC) Chart Building and Interpretation



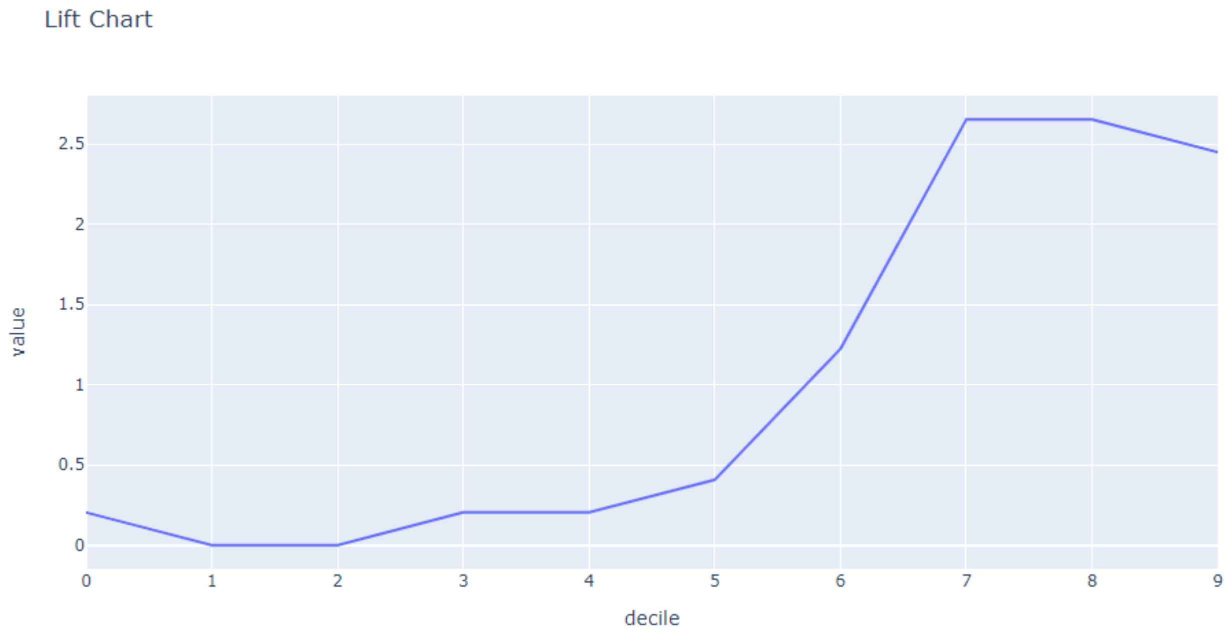
Building

- Line represents the % of true positive (Sensitivity) captured compared with false positive rate (100 - Specificity) at a given cut point

Interpretation

- Measures the classifiers ability to distinguish the two classes
    - Is the area under the curve > .50?
- The straight diagonal line shows a random. The closer the curve is to 45 degrees, the less predictive.
    - Does the curve maximize differential between random?

EXHIBIT INTRO – Lift Chart Building and Interpretation



Building

- Sort claims from low prediction to high (probability)
- Define groups by splitting into 10 equal bins
- For each bin measure the average prediction

Interpretation

- Predictions (line)
  - Is there an obvious upward trend?
  - Does there appear to be good separation?
- Decile lift (value at each line)
  - Total incremental difference in predictions from each decile to prior decile

CONTENTS

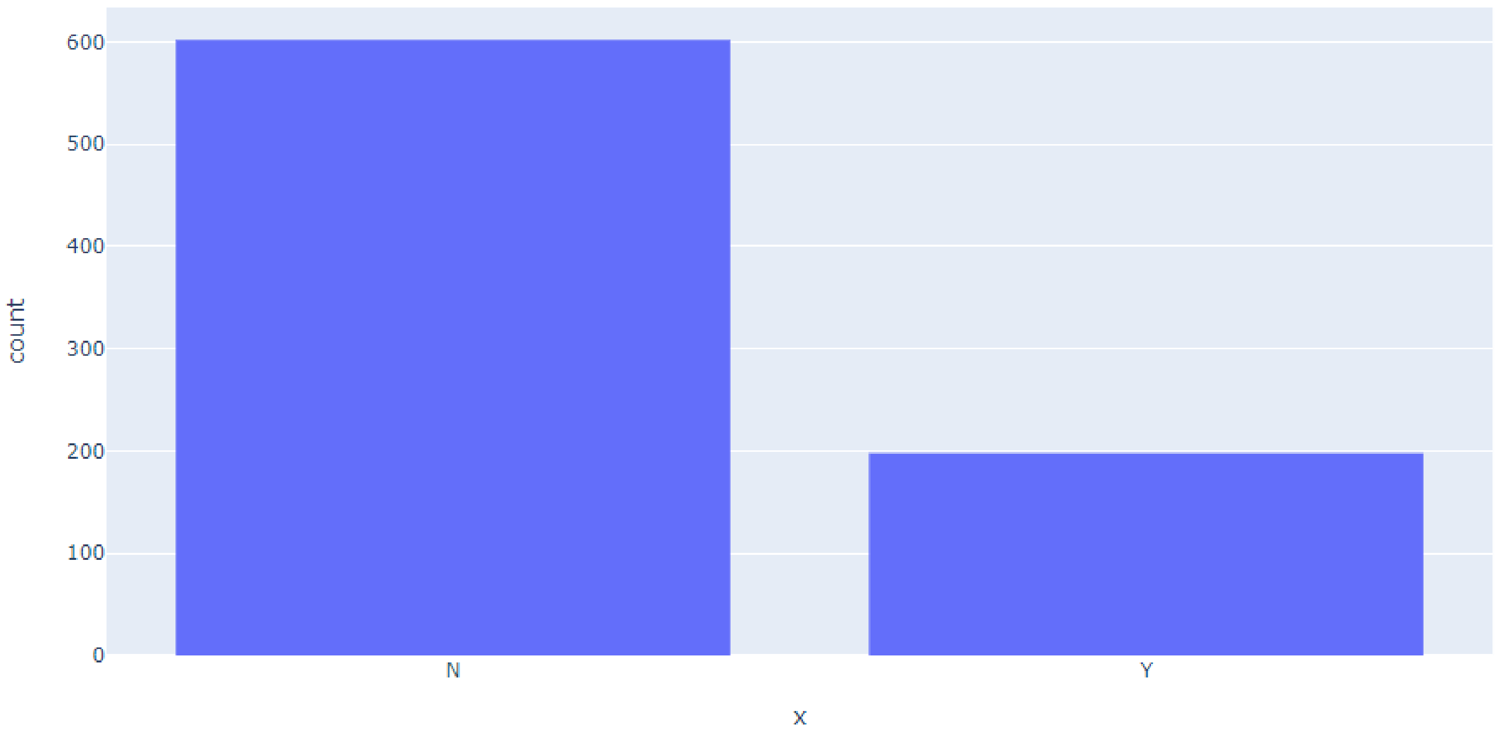
MODELING APPROACH - Overview

- 1000 rows of data
- Data split – 80% for Training, 20% for Testing
- Outlier removal with Gaussian Approximation
- Target Encoding for categorical features
- Deep Feature Synthesis using multiplicative and additive primitives
- Synthetic bootstrapping used to augment data samples
- Automated Pipeline for analysis assistance
- Model performance evaluated using lift and auc curve

MODELING APPROACH – Data (Kaggle Dataset)

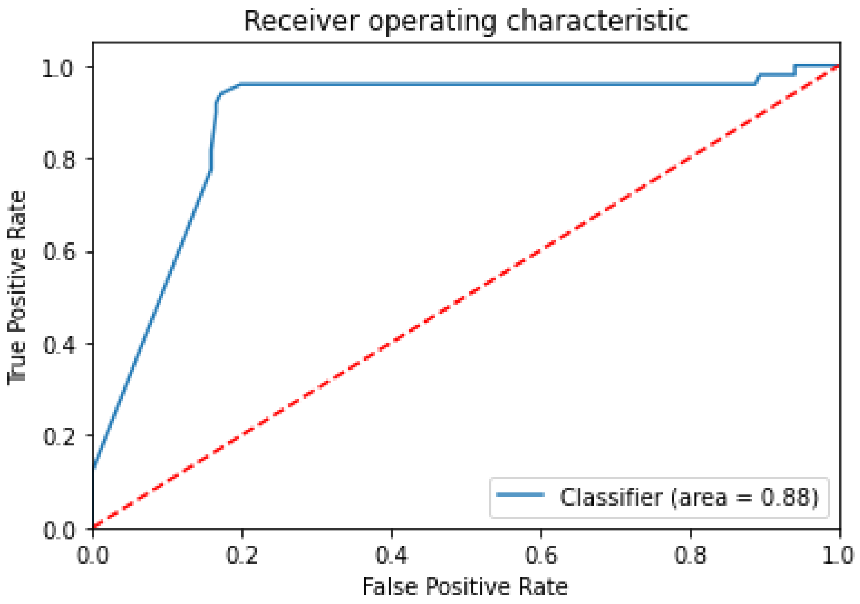
- **Independent Features**
  - 38 total (Mix of categorical/numeric)
- **Target**
  - Binary
  - Y – Fraud Reported
  - N – Fraud Not Reported

TARGET DISTRIBUTION (Training Dataset)



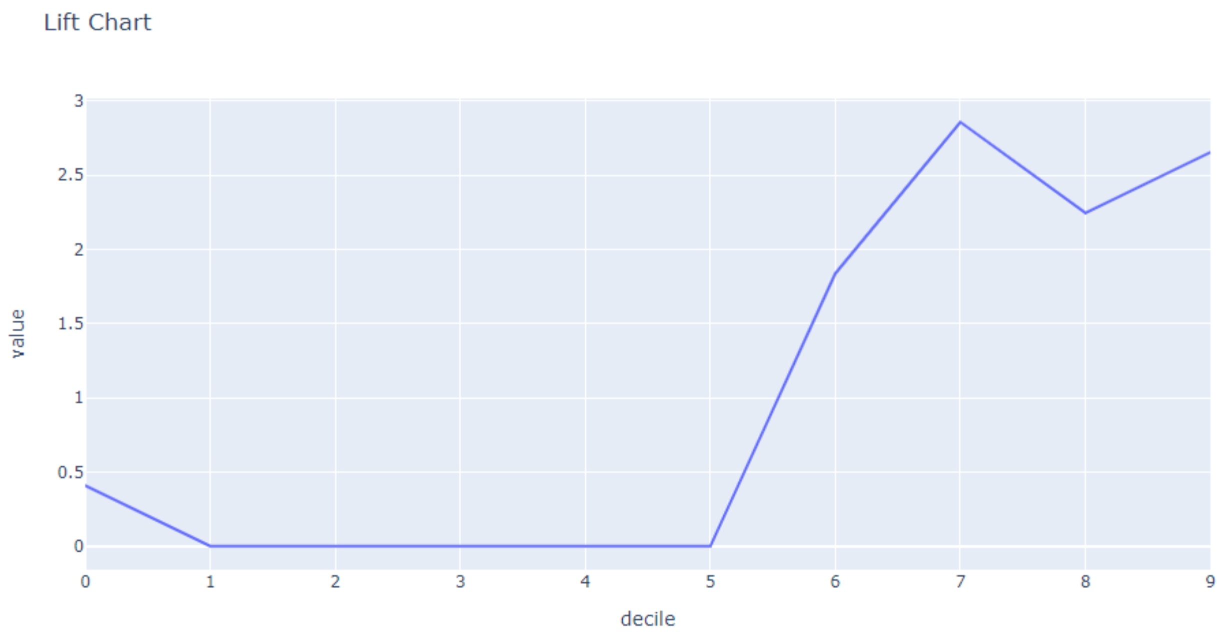
CONTENTS

FRAUD MODEL EVALUATIONS – ROC AUC Curve



- **Impressions**
  - Excellent predictive power
  - Classifier has the ability to explain ~90% of the correct positive in the first 20%
  - Stair-stepping expected, but very smooth

FRAUD MODEL EVALUATIONS – Lift Chart

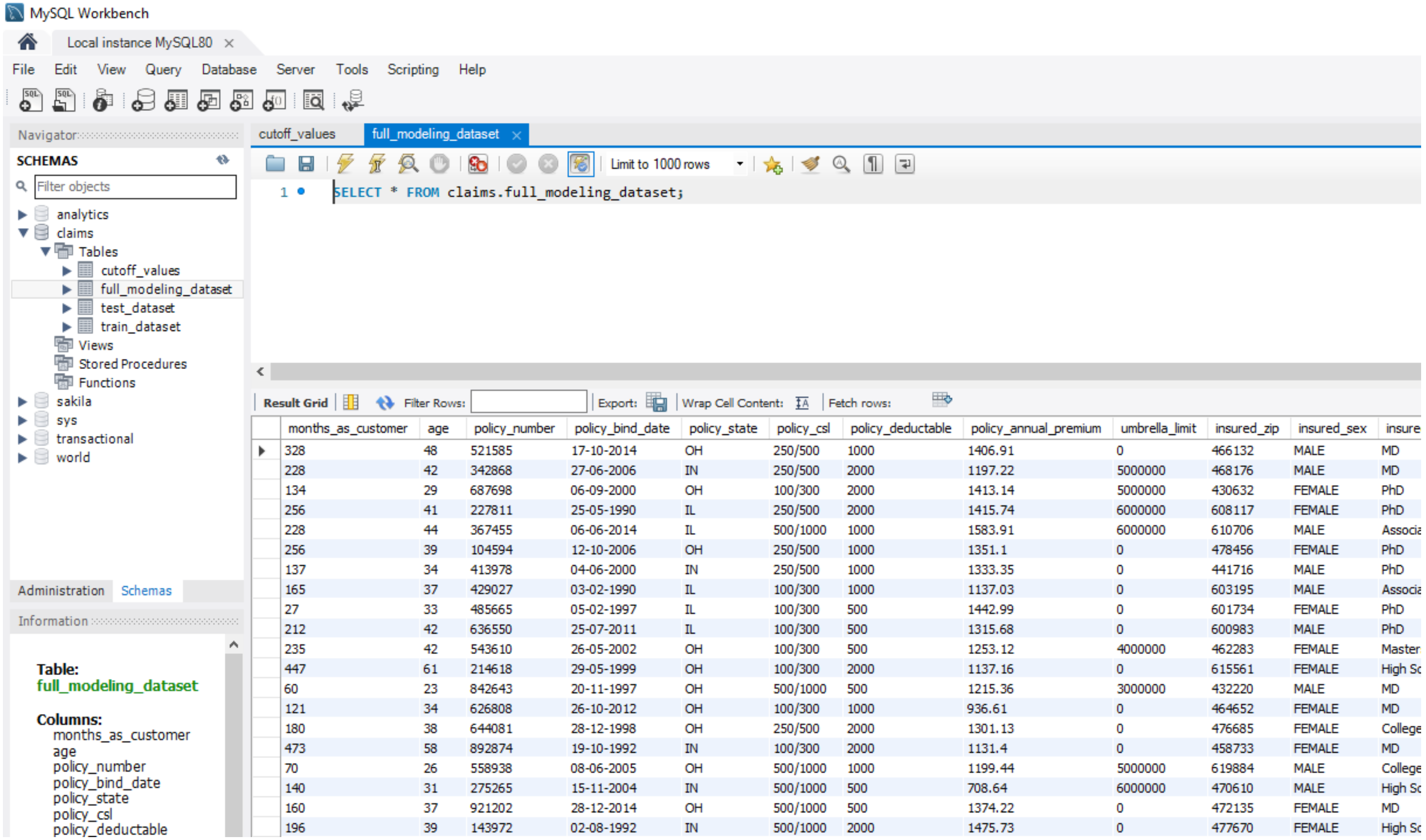


- **Impressions**
  - Great separation between the two classes
  - Cut off was not set to reduce false positives due to reduced performance
  - Volatility expected in the lower probabilities, but very little

CONTENTS

DATABASE CREATION AND EXTRACTION – Server/Database

- **MySQL Server Local**
  - Local database server created using MySQL Workbench
- **Claims Database**
  - Store full, train, and test datasets as tables
  - Store training dataset distribution information (later applied to test)
- **Rationale**
  - Emulate a production analytics environment
  - Improve query techniques for analytics ready data
- **Future Motivation**
  - Implement SQL Replication



months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	insured_sex	insure
328	48	521585	17-10-2014	OH	250/500	1000	1406.91	0	466132	MALE	MD
228	42	342868	27-06-2006	IN	250/500	2000	1197.22	5000000	468176	MALE	MD
134	29	687698	06-09-2000	OH	100/300	2000	1413.14	5000000	430632	FEMALE	PhD
256	41	227811	25-05-1990	IL	250/500	2000	1415.74	6000000	608117	FEMALE	PhD
228	44	367455	06-06-2014	IL	500/1000	1000	1583.91	6000000	610706	MALE	Associa
256	39	104594	12-10-2006	OH	250/500	1000	1351.1	0	478456	FEMALE	PhD
137	34	413978	04-06-2000	IN	250/500	1000	1333.35	0	441716	MALE	PhD
165	37	429027	03-02-1990	IL	100/300	1000	1137.03	0	603195	MALE	Associa
27	33	485665	05-02-1997	IL	100/300	500	1442.99	0	601734	FEMALE	PhD
212	42	636550	25-07-2011	IL	100/300	500	1315.68	0	600983	MALE	PhD
235	42	543610	26-05-2002	OH	100/300	500	1253.12	4000000	462283	FEMALE	Master
447	61	214618	29-05-1999	OH	100/300	2000	1137.16	0	615561	FEMALE	High Sc
60	23	842643	20-11-1997	OH	500/1000	500	1215.36	3000000	432220	MALE	MD
121	34	626808	26-10-2012	OH	100/300	1000	936.61	0	464652	FEMALE	MD
180	38	644081	28-12-1998	OH	250/500	2000	1301.13	0	476685	FEMALE	College
473	58	892874	19-10-1992	IN	100/300	2000	1131.4	0	458733	FEMALE	MD
70	26	558938	08-06-2005	OH	500/1000	1000	1199.44	5000000	619884	MALE	College
140	31	275265	15-11-2004	IN	500/1000	500	708.64	6000000	470610	MALE	High Sc
160	37	921202	28-12-2014	OH	500/1000	500	1374.22	0	472135	FEMALE	MD
196	39	143972	02-08-1992	IN	500/1000	2000	1475.73	0	477670	FEMALE	High Sc

DATABASE CREATION AND EXTRACTION – SQLAlchemy

Create (and fill) a table with 3 lines of code? Straight from a Pandas?

Reduce table create/write time

```
# Connecting to mysql database using sqlalchemy. This allows us to insert and retrieve dataframes with ease
from sqlalchemy import create_engine

# Creating sqlalchemy engine
engine = create_engine(f'mysql+mysqlconnector://{user}:{password}@127.0.0.1:3306/claims', echo=False)

# Saving the datasets
df.to_sql(name='full_modeling_dataset', con=engine, if_exists = 'append', index=False)
```

Read it back to Pandas just as easily

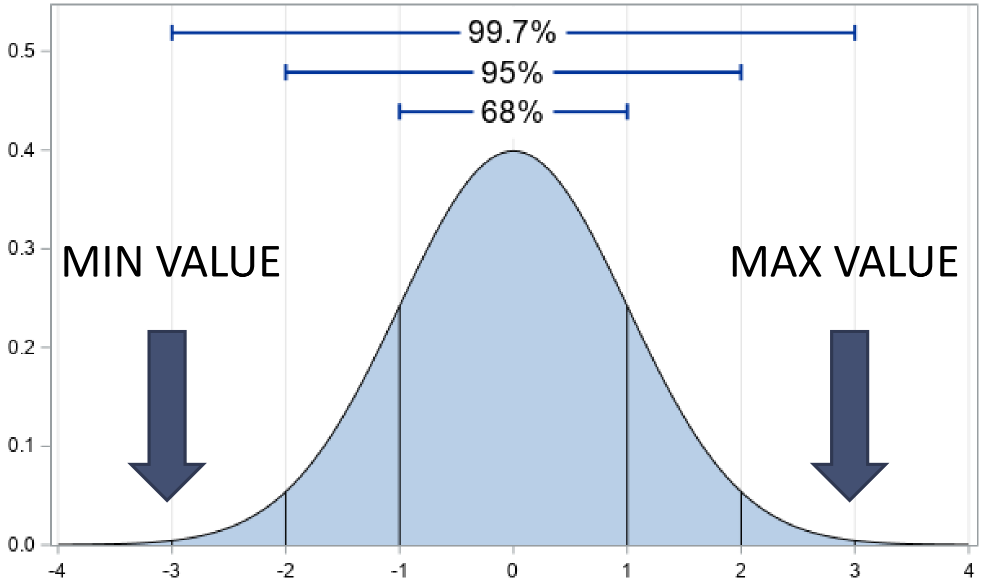
Reduce scientist extraction/query time

```
# Retrieve modeling dataset from the database

from sqlalchemy import create_engine

# Create Engine
engine = create_engine(f'mysql+mysqlconnector://{user}:{password}@127.0.0.1:3306/claims', echo=False)

# Connection
dbConnection = engine.connect()

# Reading the table into a dataframe
df = pd.read_sql("select * from claims.train_dataset", dbConnection);

# Closing the connection
dbConnection.close()
```

CONTENTS

FEATURE ENGINEERING – Outlier Removal & Encoding

- **Censoring (Capping) Methodology for Numeric Columns**
  - Calculate cutoff value from training dataset mean and standard deviation
  - Set low/high values at 3 standard deviations from mean
  - Apply to train and test features
  - Improves predictive power by removing outliers via distribution distortion



(Charfaoui, 2018)

Cutoff Value example

	feature	lower	upper
0	months_as_customer	-1.410366e+02	5.448916e+02
1	age	1.176904e+01	6.598596e+01
2	policy_deductable	-6.952633e+02	2.942763e+03
3	policy_annual_premium	5.307070e+02	1.985314e+03
4	umbrella_limit	-5.806608e+06	8.069108e+06

FEATURE ENGINEERING – Outlier Removal & Encoding

- **Target Encoding Methodology for Categorical Columns**
  - Convert categorical to numeric

BEFORE

policy_state
IL
IL
OH
OH
IN

AFTER

policy_state
0.224806
0.224806
0.274021
0.274021
0.241379

FEATURE ENGINEERING – Feature Tools (Automated Feature Engineering)

- **Deep Feature Synthesis**
  - Automatically create new features from dataset
  - Provides more features to try in model with ease
  - Saves massive amounts of time
- **Created several new features**
- **Able to work with relational database structure**
  - Assists analysts understand what features are important in a large database

EXAMPLE:

age
39.0
35.0
44.0
62.0
30.0

+

incident_hour_of_the_day
1.0
23.0
15.0
13.0
8.0

=

age + incident_hour_of_the_day
40.0
58.0
59.0
75.0
38.0

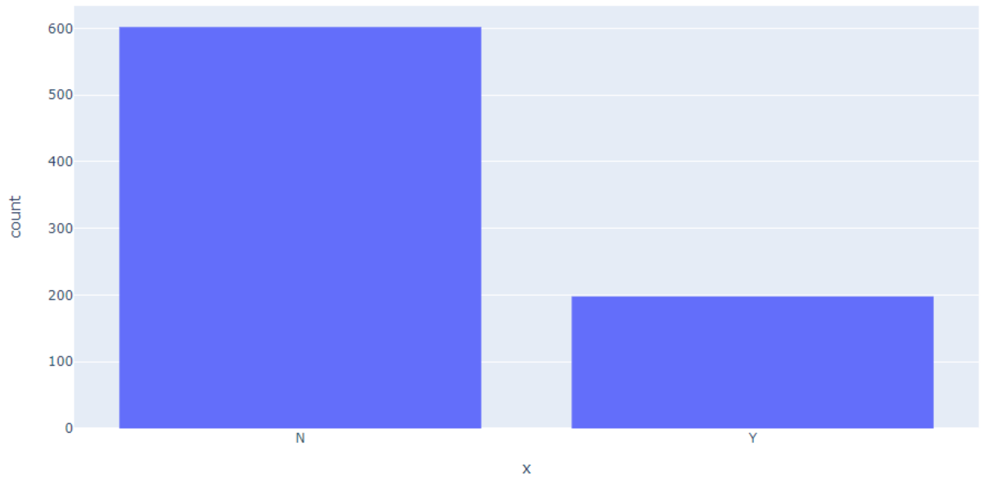
CONTENTS

- Exhibit Intro
- Modeling Approach
- Fraud Model Evaluations
- Database Creation and Extraction Techniques
- Feature Engineering
- **Bootstrapping**
- Modeling Techniques (TPOT, Pipeline, Gridsearch, Etc)
- Implementation / Business Impact
- Future Steps

BOOTSTRAPPING– Synthetic Sampling (SVMSMOTE)

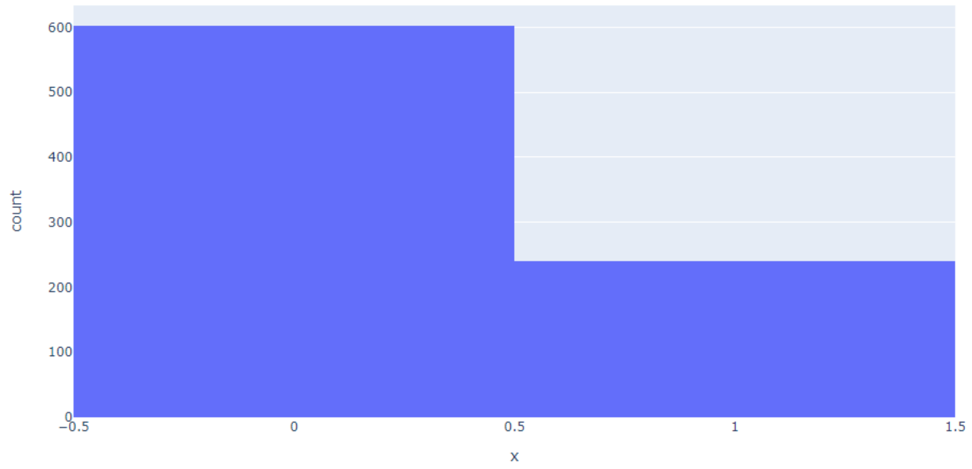
- **Increases data volume**
  - Basically, gives us more data!
- **Results in a more robust model**
  - With such a small dataset, this technique allows us to better generalize the population

BEFORE



Bootstrap

AFTER



CONTENTS

MODELING TECHNIQUES – A Shameless Automation Plug: TPOT and DASK

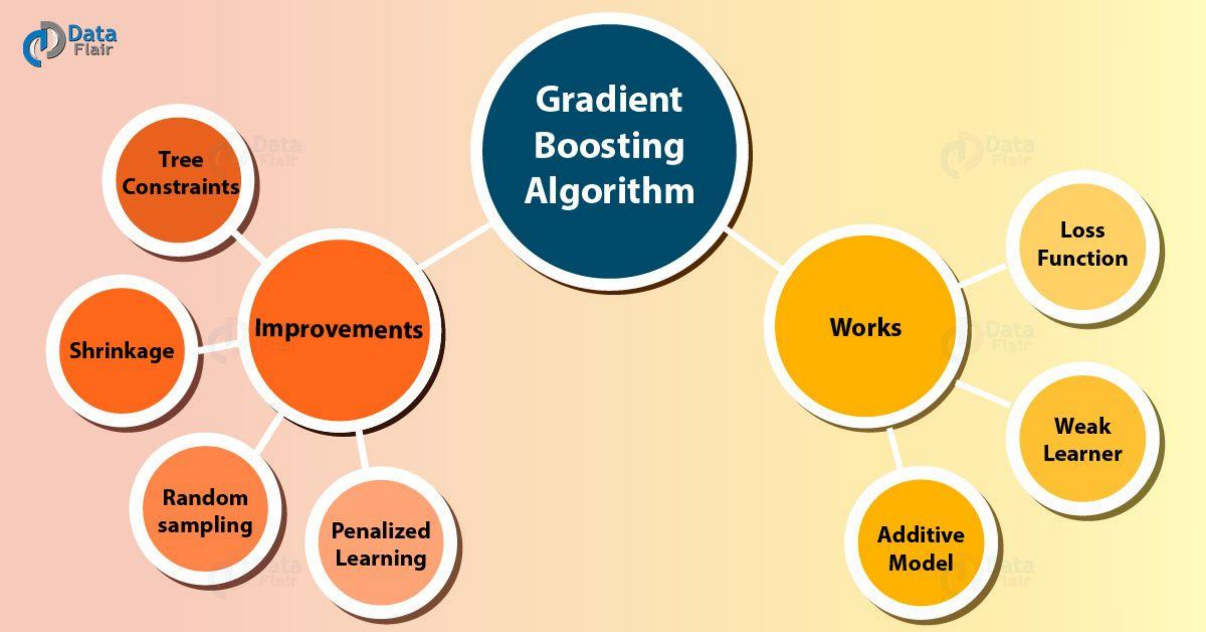
- **TPOT Automated Pipeline**
  - A genetic programming approach to selecting an optimal solution
  - Tries many different pipelines
  - Gives the analyst a good starting point with a difficult problem
  - Helps, but does not replace the scientist
- **DASK**
  - Parallel computing for large data
  - Allows for multiple processes to run at same time for quicker solve time

EXAMPLE

MODELING TECHNIQUES – TPOT and DASK

MODELING TECHNIQUES – Final Model and Feature Selection

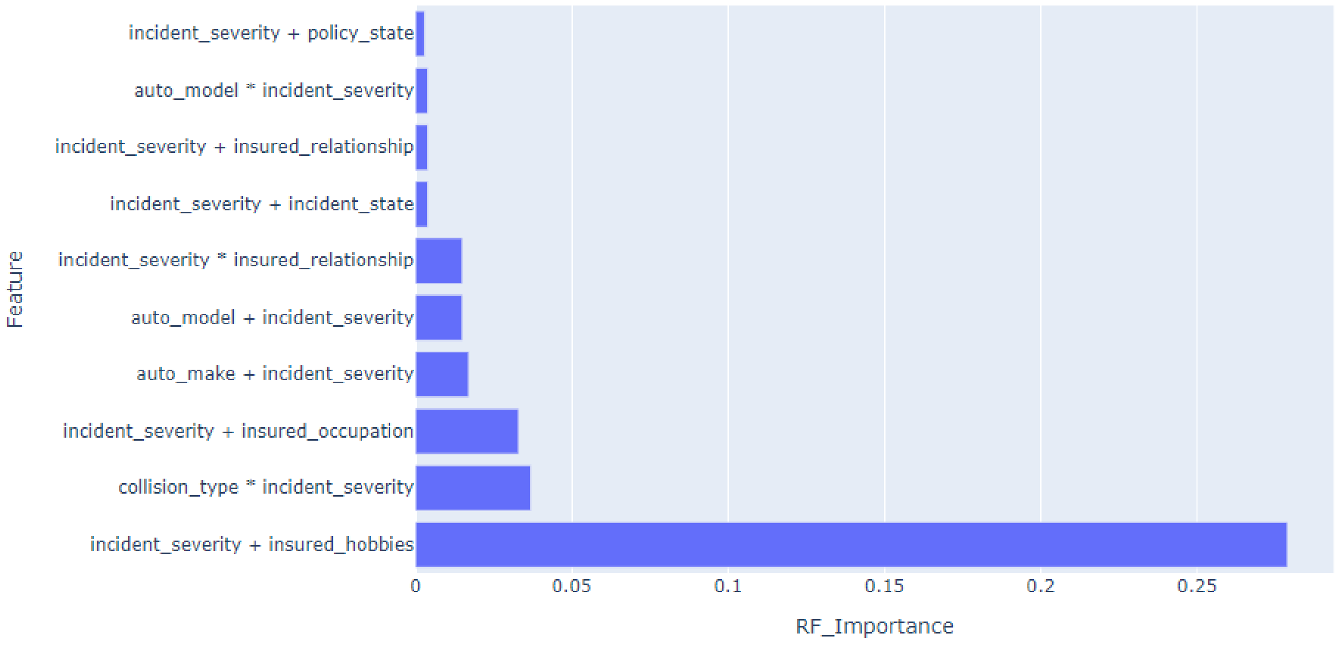
- **Gradient Boosting Classifier**
  - Best performer
  - Allowed for prediction explanation
  - Ensemble for smoother lift
- **KBest with Anova feature selection**
  - Reduce non-predictive features
  - Iteratively try different combinations
  - Faster predictions!



(Gradient Boosting Algorithm – Working and Improvements, n.d.)

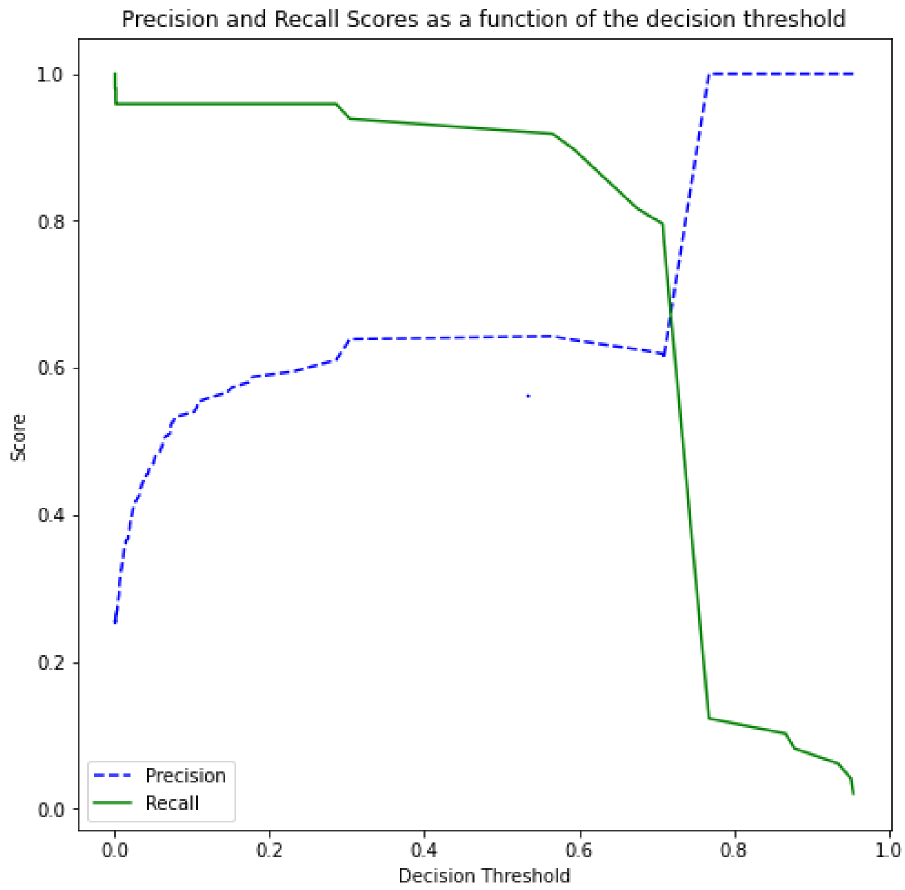
MODELING TECHNIQUES – Important Features

- **Permutation Importance**
  - Gives better understanding of what is really important
  - Incident severity: The real hero
  - Insured hobbies: The real hero?
- **All important features from FeatureTools**
- **High cardinality may be influencing the importance here**



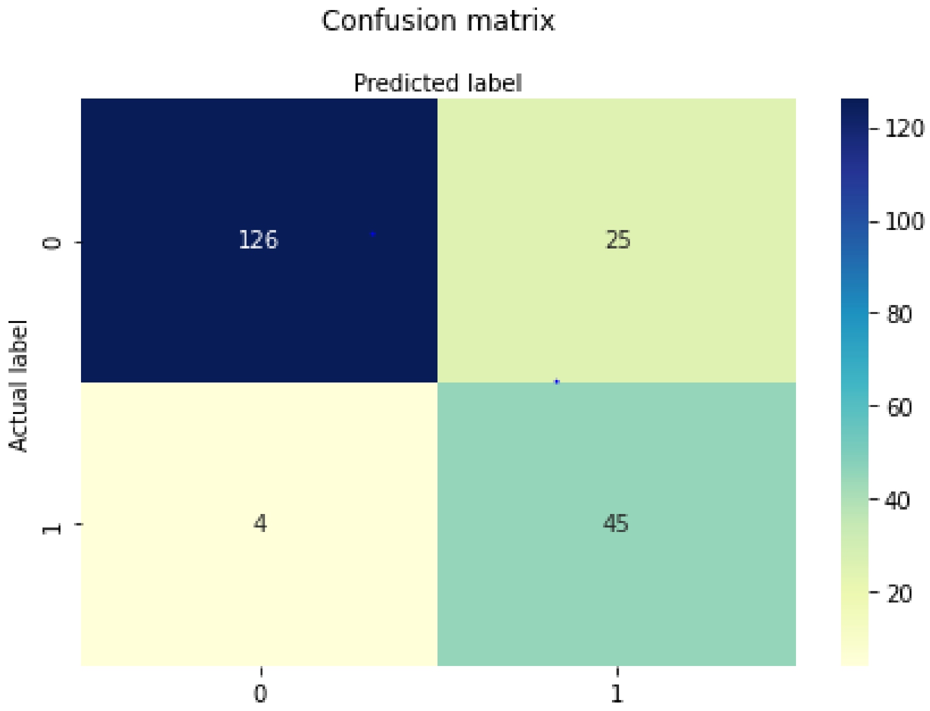
MODELING TECHNIQUES – Precision vs Recall

- The chart to the left shows the difference between precision and recall
- The chart to the right shows the models tradeoff between these two metrics at various decision thresholds
- The threshold was kept at the default of .50. There was little improvement adjusting this value




```
Accuracy: 0.855
Precision: 0.6428571428571429
Recall: 0.9183673469387755
```

MODELING TECHNIQUES – Confusion Matrix



- **Impressions**
  - Excellent false negative rates
  - False positives leave something to be desired
  - Overall accuracy is excellent
- **Future Recommendation**
  - Create two separate models
    - One for explaining predictions
    - One for detection

CONTENTS

IMPLEMENTATION / BUSINESS IMPACT – What now?

(Weaver, 2019)

- **Implementation Considerations**
  - High false positive rate
    - May impact customer treatment
  - Cut off may need tweaking
- **Beta Testing**
  - Consider a pilot test group
- **Training**
  - Educate adjusters to maximize model effectiveness

IMPLEMENTATION / BUSINESS IMPACT – Explaining Predictions

Example of Fraudulent Claim

y=1 (probability **0.709**, score **0.446**) top features

Contribution?	Feature
+1.292	incident_severity + insured_hobbies
+0.266	collision_type * incident_severity
+0.245	incident_severity + insured_occupation
+0.049	auto_model + incident_severity
+0.041	incident_severity + policy_state
-0.054	auto_model * incident_severity
-0.171	auto_make + incident_severity
-0.182	incident_severity + insured_relationship
-0.278	incident_severity + incident_state
-0.305	incident_severity * insured_relationship
-0.458	<BIAS>

Based on the input, we can see that the top contributing feature in this prediction is the additive interaction between incident_severity and insured_hobbies

- **Model has the ability to explain why a claim may be fraudulent**
- **Based on the input values, an adjuster will receive more information relating to the prediction.**

Example of possible reason message:

Special Investigative Unit Referral

A fraudulent claim has been detected on policy 217938

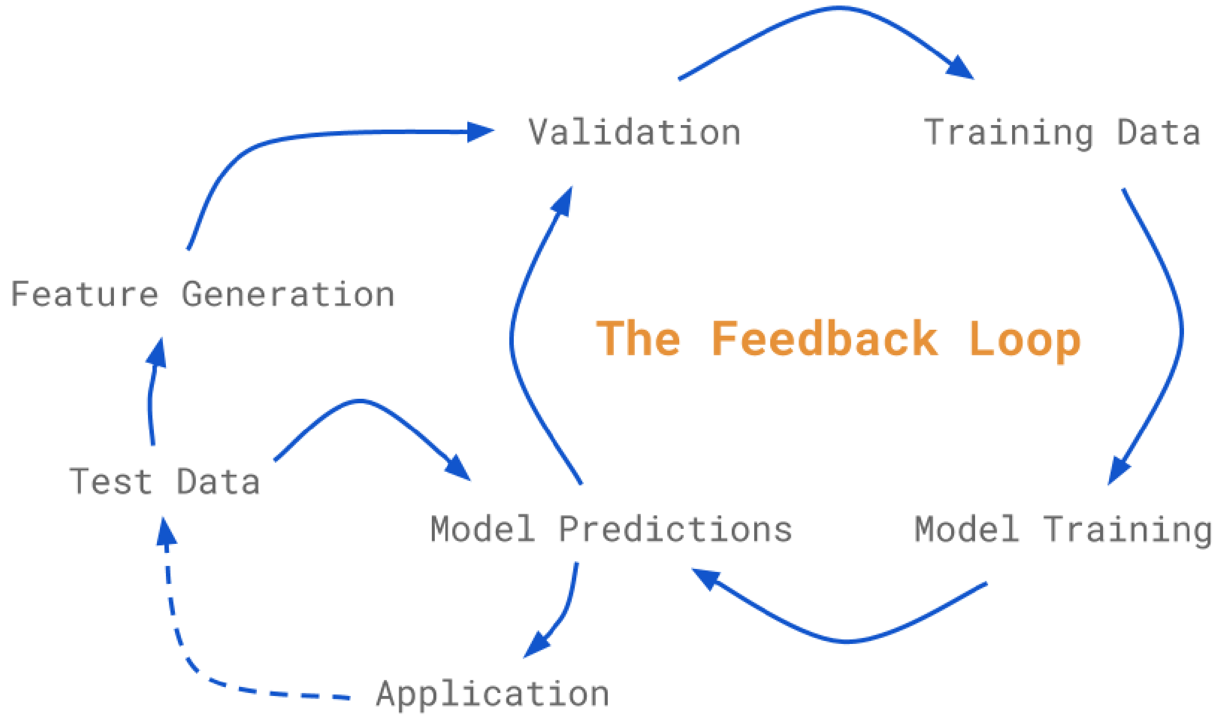
This claim was flagged due to:

Interaction between the incident severity (Major Damage) and the insured's hobby (Sky Diving) is more likely than average to be fraudulant

CONTENTS

FUTURE STEPS – Feedback

- **An iterative process**
  - Future products depend on business insight and suggestion
  - Improved performance overtime
- **Is it working?**
  - Hands-on experience can be the best guidance
- **More data is better**
  - Understanding how this impacts adjusters creates actionable insight data. What action did they take?



(Morrison, 2019)

FUTURE STEPS – Two Models

- **Split the product into two separate models**
    - One model focused on increasing predictive power
    - One model to focus on explaining the prediction
- **The product does both fairly well, but there is always room for improvement**
- **Feedback loop may provide more data for the explanation model**
    - Improve insight into fraudulent claims
    - Provide actionable insight based on previous predictions
- **Explore other models less ideal for score reasons, more ideal for predictability**
    - Black box models may improve results

References

Charfaoui, Y. (2018, June). *Hands-on with Feature Engineering Techniques: Handling Outliers*. Retrieved from HeartBeat: https://heartbeat.fritz.ai/hands-on-with-feature-engineering-techniques-dealing-with-outliers-fcc9f57cb63b

Gradient Boosting Algorithm – Working and Improvements. (n.d.). Retrieved from Data Flair: https://data-flair.training/blogs/gradient-boosting-algorithm/

Morrison, J. (2019, December 9). *Designing Effective Supervised Machine Learning Systems*. Retrieved from Medium: https://medium.com/@joemorrison/designing-effective-supervised-machine-learning-systems-91eb7b466129

Precision and Recall. (2020, October 9). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Precision_and_recall

Weaver, D. (2019, May 15). *Three Tips for Deferring Insurance Fraud*. Retrieved from Inform: https://www.inform-software.com/blog/post/3-tips-for-deterring-insurance-fraud