Group 7
Dazed & Confusion MATRIX 2.0: FINAL PRESENTATION

Andrea Irina Yzeiri | 260991862

Bogdan Tanasie | 260747949

Marek Krowicki | 260175752

Sam Greene | 260722742

Tiancheng Zhang | 260974250

OUR TEAM

Andrea Yzeiri
Project Manager/Data Scientist
"I can throw something together."

Bogdan Tanasie
Software Developer/Architect
"I think it's not letting me push."

Marek Krowicki
Data Analyst
"I bow to our new machine learning overlords."

Sam Greene
Business Analyst
"As long as I don't have to do the same work as capstone."

Tiancheng Zhang
Data Analyst
"I can relate to Sam."

Content Synopsis
Presentation OVERVIEW

FRAMEWORK

- CONTEXT
- HYPOTHESIS
- THREATS TO VALIDITY
- GITHUB

ANALYSIS

- DATA
- MODELLING
- CONCLUDING RESULTS

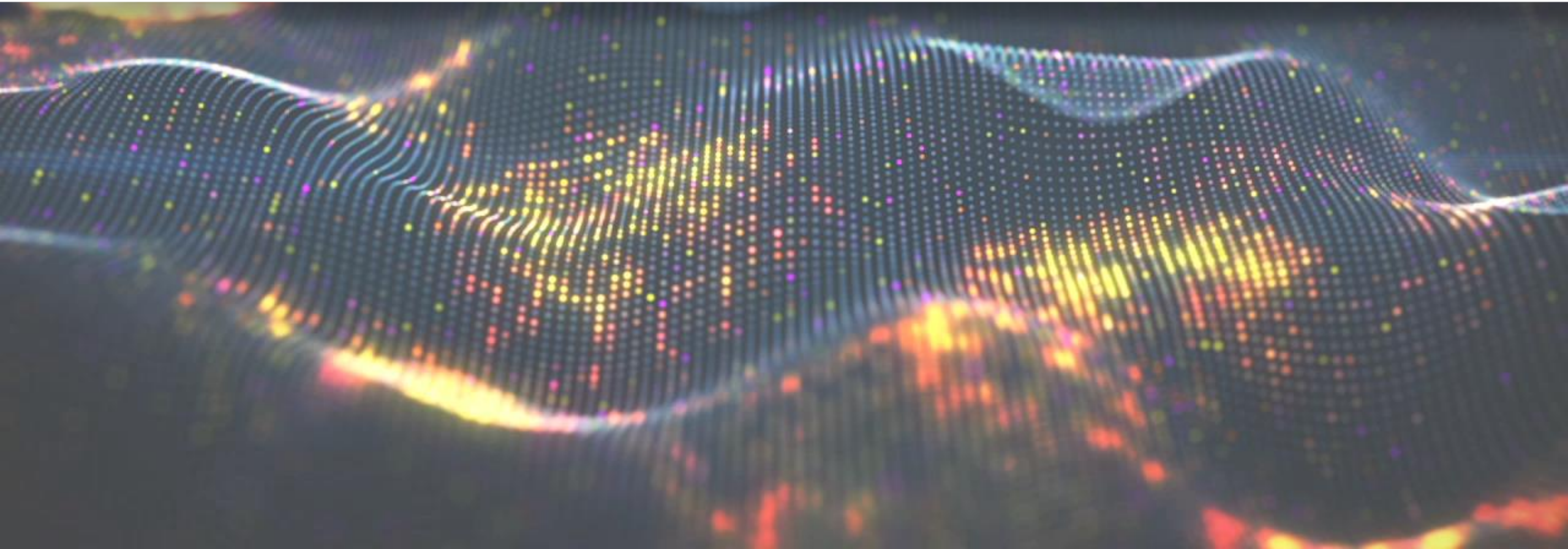
ARCHITECTURE

- LIVE DEMO
- LAUNCHING, MONITORING & MAINTENANCE

ACADEMIC

- LESSONS LEARNED
- FUTURE WORK

FRAMEWORK



CONTEXT

USE CASE – PREDICTIVE POLICING

Predicting the type of crime that will occur based on current characteristics of crime, demographic information and the inherent emergency call.

- “With 58 murders per 100,000 residents in 2019, Baltimore is the deadliest U.S. city…”
- “ The better answer is to give police departments the resources they need to implement meaningful reforms…”

CONTEXT

IDEAL DATA

- Complete data on crimes in Baltimore
- Current, data on crime sprees (arsonery, sexual assault, robberies, etc)
- Criminal Records data
- Last 3 years <
- FBI database
- Police force databases in Baltimore [data augmentation]
- Prisons *Dependent*
- High legal obligations
- High clearance/authorization
- Protect sensitive information on criminals
- Security threats

Data Acquisition

Original

- Open source data on crimes in Baltimore
- Last 2 years
- Limited in variables
- Kaggle
- No legal obligations
- No sensitive information
- Used to test our hypothesis and begin a research cycle!

Augmented

- Open source data
- Last 2 years
- Increased demographic information
- Predictive abilities, 911

HYPOTHESIS

Logical

An *accurate* predictive tool will improve the Baltimore's police force's KPI's.

HYPOTHESIS

Logical

An *accurate* predictive tool will improve the Baltimore's police force's KPI's.

Empirical

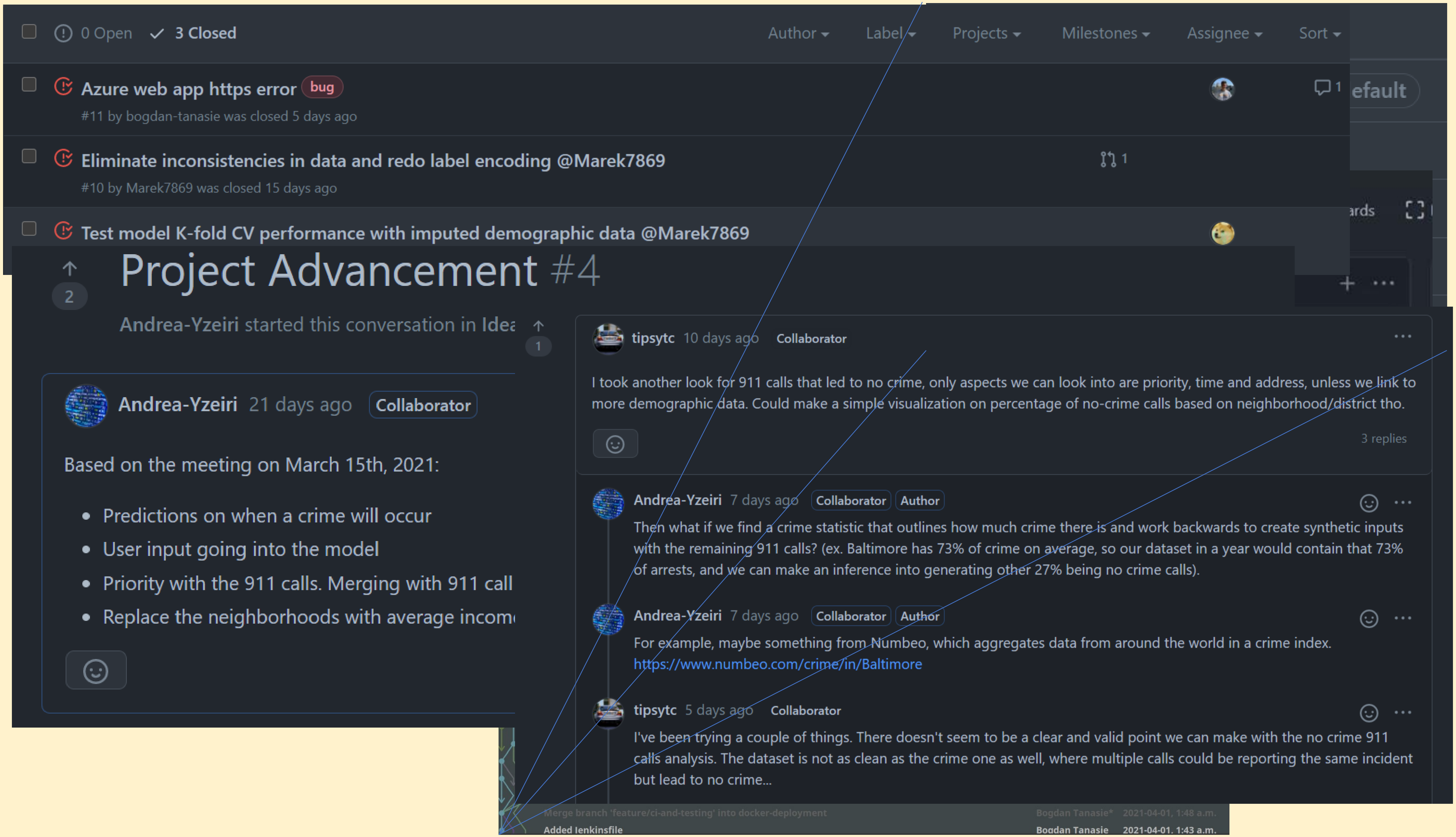
LightGBM provides the best balance of precision and recall for predicting the type of crime to occur next in Baltimore.

THREATS TO VALIDITY

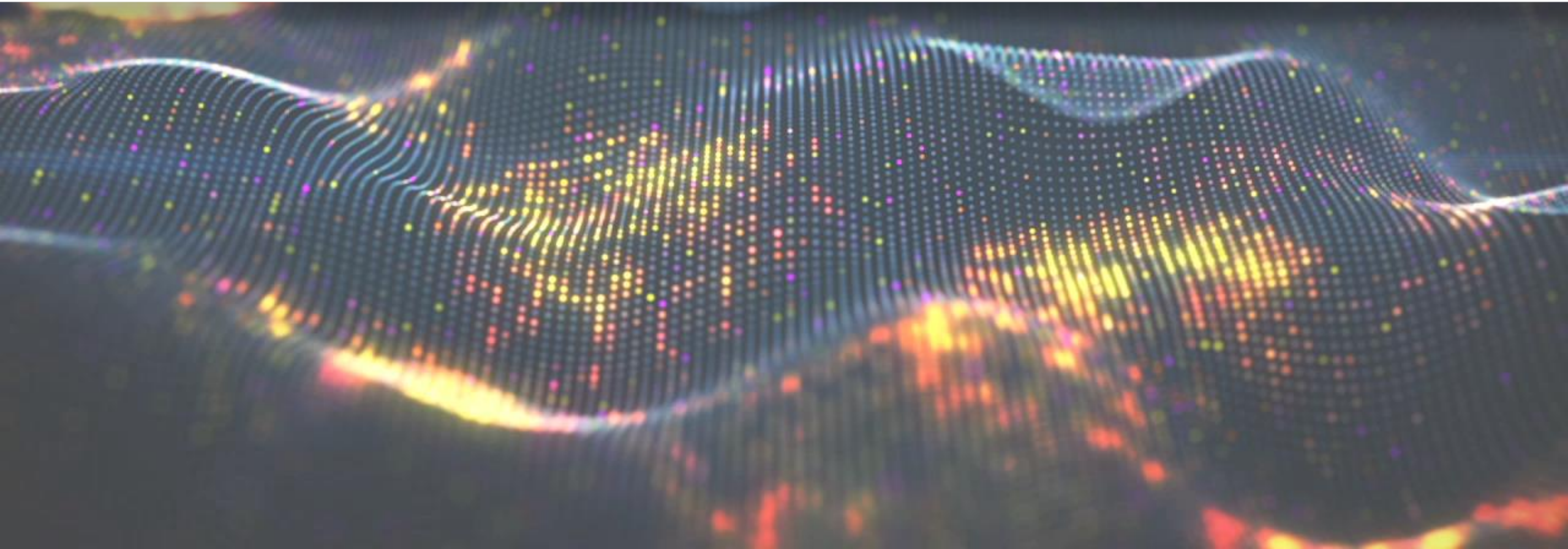
Sampling Bias | There is no instance of “NO crime” in the dataset.

Economic crashes or social changes outside of the norm. | History

Maturation | Crime rates can improve over time.


0 Open
3 Closed
Author
Label
Projects
Milestones
Assignee
Sort
Azure web app https error
bug
#11 by bogdan-tanasie was closed 5 days ago
1
Eliminate inconsistencies in data and redo label encoding @Marek7869
#10 by Marek7869 was closed 15 days ago
1
Test model K-fold CV performance with imputed demographic data @Marek7869
Project Advancement #4
2
Andrea-Yzeiri started this conversation in Idea
Andrea-Yzeiri 21 days ago Collaborator
Based on the meeting on March 15th, 2021:
Predictions on when a crime will occur
User input going into the model
Priority with the 911 calls. Merging with 911 call
Replace the neighborhoods with average incom
1
tipsytc 10 days ago Collaborator
I took another look for 911 calls that led to no crime, only aspects we can look into are priority, time and address, unless we link to more demographic data. Could make a simple visualization on percentage of no-crime calls based on neighborhood/district tho.
3 replies
Andrea-Yzeiri 7 days ago Collaborator Author
Then what if we find a crime statistic that outlines how much crime there is and work backwards to create synthetic inputs with the remaining 911 calls? (ex. Baltimore has 73% of crime on average, so our dataset in a year would contain that 73% of arrests, and we can make an inference into generating other 27% being no crime calls).
Andrea-Yzeiri 7 days ago Collaborator Author
For example, maybe something from Numbeo, which aggregates data from around the world in a crime index.
https://www.numbeo.com/crime/in/Baltimore
tipsytc 5 days ago Collaborator
I've been trying a couple of things. There doesn't seem to be a clear and valid point we can make with the no crime 911 calls analysis. The dataset is not as clean as the crime one as well, where multiple calls could be reporting the same incident but lead to no crime...
Merge branch 'feature/ci-and-testing' into docker-deployment
Bogdan Tanasie* 2021-04-01, 1:48 a.m.
Added Jenkinsfile
Bogdan Tanasie 2021-04-01, 1:43 a.m.

ANALYSIS



DATA AUGMENTATION – 911 Calls

"The decision to call 911 to report a crime may or may not affect the person deciding to do so and might not even affect the outcome of the particular incident at hand. When many people in a neighborhood regularly report crimes to 911, the quality of life in the neighborhood and the safety of all its residents can be substantially affected."

- Mario L. Small. **Understanding when people will report crimes to the police.** *Proceedings of the National Academy of Sciences.* Aug 2018, 115 (32) 8057-8059

	Neighborhood	median_household_income	calls_per_household
64	ABELL	32241.138221	0.083382
15	ALLENDALE	36701.906742	0.468165
187	ARCADIA	66010.877444	0.033961
87	ARLINGTON	33139.182464	0.737275
89	ARMISTEAD GARDENS	33179.511029	0.111464

```
result.calls_per_household.corr(result.median_household_income)
```

```
-0.36076924710314096
```

New Features

1. Priority (5 different categories)
2. Description from the call (142 different categories)

Matching Process

1. Exact match on the crime scene street address
2. 911 Calls made 24 hrs before or after the crime
3. Drop duplicates

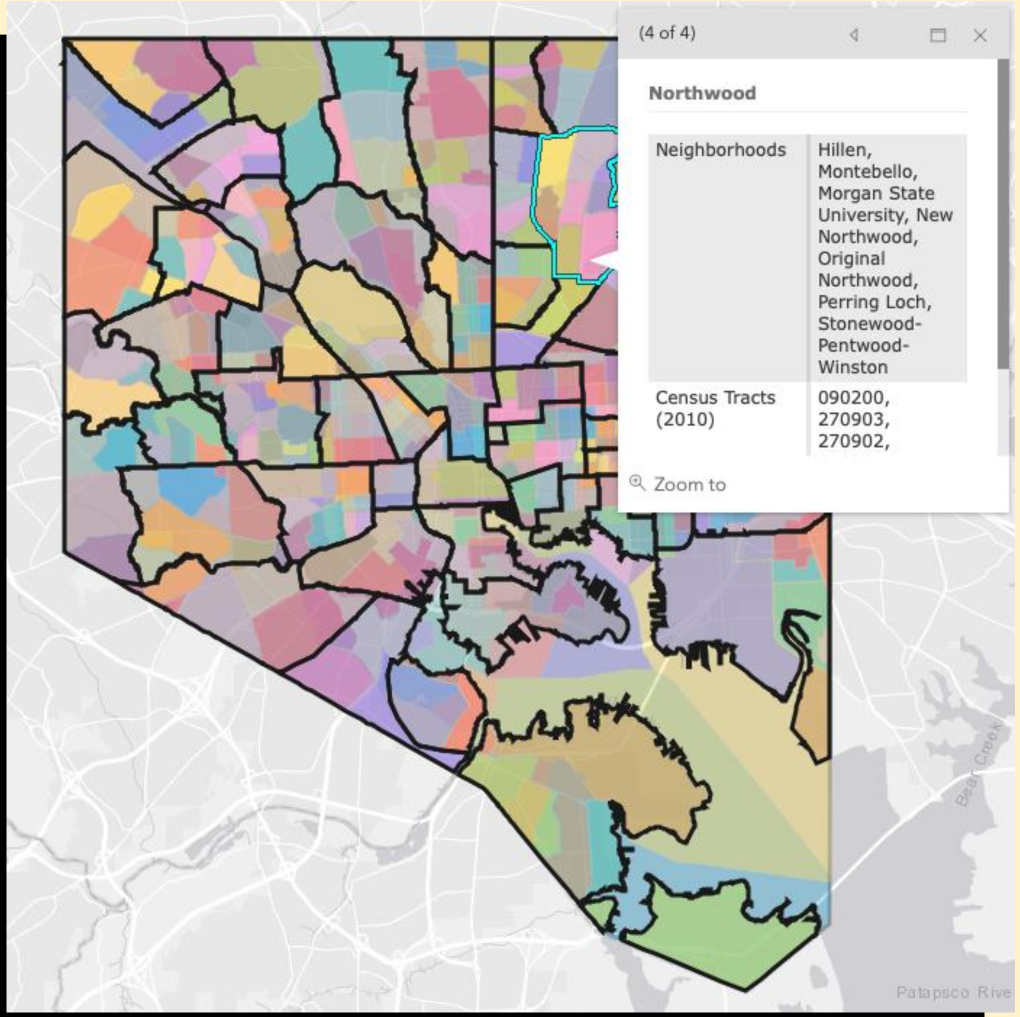
9275 unique matches were found in the year of 2015.

DATA AUGMENTATION- Demographics

- Reasoning – understand how predictions affected areas with certain demographic characteristics to then be able to assess bias
- Source – Open Data Baltimore
- Variables gathered:
  - Median household income
  - Median price of homes sold
  - Percent of households living below poverty line
  - Percent of population 18-24, 25-64, 65+
  - Percent of Asian, Hispanic, African-American, White residents
  - Racial Diversity index
  - Total number of households

DATA AUGMENTATION- Demographics

- Data was collected by Community Statistical Area (CSA)
- Each CSA is a group of neighborhoods
- Manually matched neighborhood in our data to corresponding CSA
- Imputed data for missing years with median for same CSA of other years



DATA AUGMENTATION – Demographics

- Demographic data not available for all neighborhoods
- Imputed missing data with MICEforest
  - Multiple Imputation by Chained Equations (MICE) imputes missing data through an iterative series of predictive models, in this case random forest
  - Uses predictive mean matching, selecting an original datapoint close to the predicted value of the missing sample
  - Closest N values are chosen as candidates, and a value is chosen at random
- Imputed values were averaged by missing neighborhood, to preserve structure of data augmentation

DATA AUGMENTATION – Ethical AI

Artificial intellig

Predi

If the initial bias in predictive policing is for factors other than crime, the result may be the deepening of injustice

by J

AI researchers condemn

How to Fight Bias with Predictive Policing

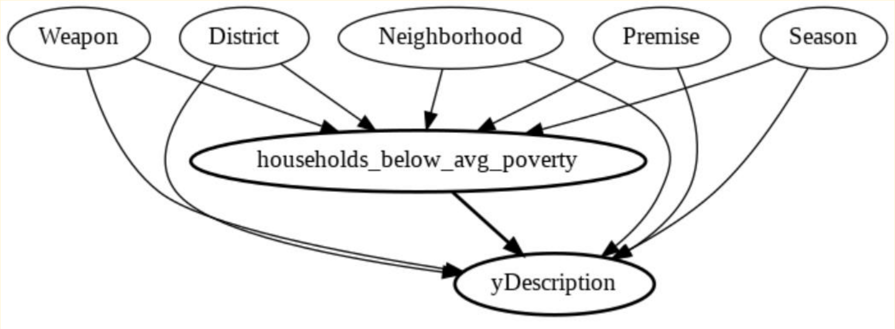
The data-driven technique can perpetuate inequality, but if done right, it also presents an unprecedented opportunity to advance social justice

DATA AUGMENTATION – Ethical AI

- Socioeconomic correlations
  - Poverty & Crime
    - "… poverty is the dominant explanatory factor with regard to aggravated assault & burglary". (Poverty, Urbanization, And Crime)
    - "The findings imply that problems in neighborhoods begin to manifest themselves at much lower levels of poverty." (GHETTOS, THRESHOLDS, AND CRIME: DOES CONCENTRATED POVERTY REALLY HAVE AN ACCELERATING INCREASING EFFECT ON CRIME?)
  - Urbanization & Crime
    - "…urbanisation encourages crime as the rate of crime is higher in large cities and in urbanised areas." (Urbanisation and crime: a case study of Pakistan)
- Baltimore Facts
  - Highest youth crime rate
    - "Nearly half of suspects recently arrested by BPD for crimes are juveniles."
  - High youth homicide rate
    - "However, for youth (under 25 years old), the homicide mortality rate was 31.3 per 100,000 youth."

CAUSAL INFERENCE – Demographics

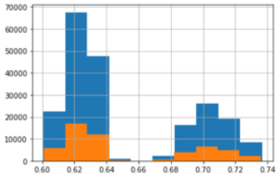
- Example of setup where we use description as target and a demographic variable as treatment



% 18-24

	Weapon_FIREARM	Weapon_HANDS	Weapon_KNIFE	Weapon_NONE	Weapon_OTHER	Neighborhood	Premise	Month	Hour	Outside
coefficient	0.440629	0.310348	0.372136	-0.001046	0.435249	-0.000336	0.000286	0.000459	-0.000133	-0.059341

```
[2021-04-08 18:46:43,066|kedro.runner.sequential_runner|INFO] Completed 1 out of 1 tasks
[2021-04-08 18:46:43,068|kedro.runner.sequential_runner|INFO] Pipeline execution completed successfully.
[2021-04-08 18:46:43,076|causallift.context.flexible_context|INFO] Run pipeline (tags: ['221_estimate_propensity'], SequentialRunner, only_missing: False)
[2021-04-08 18:46:43,091|kedro.pipeline.node|INFO] Running node: estimate_propensity([args,df_00,propensity_model]) -> [df_01]
[2021-04-08 18:46:43,140|causallift.nodes.estimate_propensity|INFO]
### Histogram of propensity score for train and test data:
```




```
[2021-04-08 18:46:43,442|causallift.nodes.estimate_propensity|INFO]
### Score Table for logistic regression to calculate propensity score:
```

	# samples	# classes	accuracy	precision	recall	f1	roc_auc	observed CVR	predicted CVR
train	210496	2	0.652772	0.652772	1.0	0.789912	0.5	0.652772	1.0
test	52625	2	0.652770	0.652770	1.0	0.789910	0.5	0.652770	1.0

```
[2021-04-08 18:46:43,840|causallift.nodes.estimate_propensity|INFO]
### Confusion Matrix for Train:
```

	Pred_0	Pred_1
True_0	0	73090
True_1	0	137406

```
[2021-04-08 18:46:43,997|causallift.nodes.estimate_propensity|INFO]
### Confusion Matrix for Test:
```

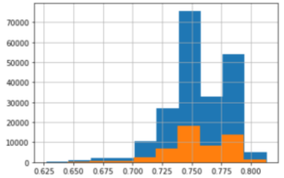
	Pred_0	Pred_1
True_0	0	18273
True_1	0	34352

- DoWhy not suitable for this type of categorical analysis
- Instead use CausalLift API
  - Use binary and multiclass setup to run these models
  - Interesting to see effects of demographic variables on model
  - Still more to explore

Average Median Income

	Weapon_FIREARM	Weapon_HANDS	Weapon_KNIFE	Weapon_NONE	Weapon_OTHER	Neighborhood	Premise	Month	Hour	Outside
coefficient	0.13235	0.067172	0.0	0.123579	0.058252	0.000261	0.005184	-0.001545	0.002284	-0.258417

```
[2021-04-08 18:47:06,846|kedro.runner.sequential_runner|INFO] Completed 1 out of 1 tasks
[2021-04-08 18:47:06,848|kedro.runner.sequential_runner|INFO] Pipeline execution completed successfully.
[2021-04-08 18:47:06,856|causallift.context.flexible_context|INFO] Run pipeline (tags: ['221_estimate_propensity'], SequentialRunner, only_missing: False)
[2021-04-08 18:47:06,873|kedro.pipeline.node|INFO] Running node: estimate_propensity([args,df_00,propensity_model]) -> [df_01]
[2021-04-08 18:47:06,923|causallift.nodes.estimate_propensity|INFO]
### Histogram of propensity score for train and test data:
```




```
[2021-04-08 18:47:07,121|causallift.nodes.estimate_propensity|INFO]
### Score Table for logistic regression to calculate propensity score:
```

	# samples	# classes	accuracy	precision	recall	f1	roc_auc	observed CVR	predicted CVR
train	210496	2	0.756366	0.756366	1.0	0.861285	0.5	0.756366	1.0
test	52625	2	0.756371	0.756371	1.0	0.861288	0.5	0.756371	1.0

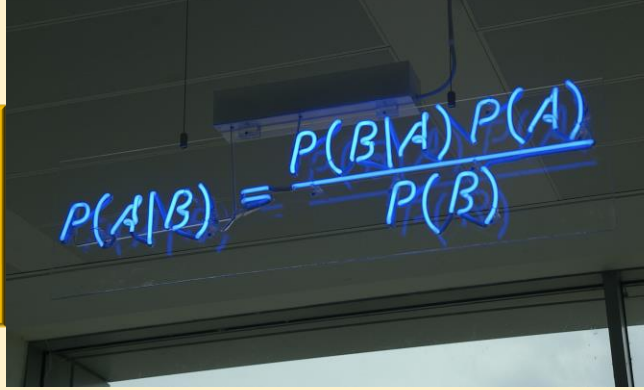
```
[2021-04-08 18:47:07,579|causallift.nodes.estimate_propensity|INFO]
### Confusion Matrix for Train:
```

	Pred_0	Pred_1
True_0	0	51284
True_1	0	159212

```
[2021-04-08 18:47:07,756|causallift.nodes.estimate_propensity|INFO]
### Confusion Matrix for Test:
```

	Pred_0	Pred_1
True_0	0	12821
True_1	0	39804

EXPLORATORY MODEL – Bayesian Network



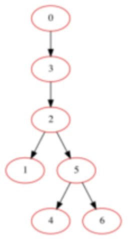
- A **Bayesian network** is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG).
  - Exact fitting is computational costly
  - Chow-Liu approximation returns the optimal tree-like structure for the Bayesian network
  - Greedy approximation (default) greedily attempts to find the best structure

- Needs SME to justify assumption
- Alternative imputation method
- Alternative causal inference

```
#SUPER FAST with Chow-Liu Approximation
model1 = BayesianNetwork.from_samples(train_array, algorithm='chow-liu')
print(model1.structure)
```

```
((), (2,), (3,), (0,), (5,), (2,), (5,))
```

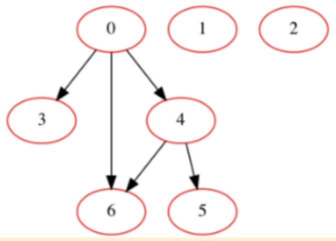
```
model1.plot()
```




```
#Greedy
model2 = BayesianNetwork.from_samples(train_array, algorithm='greedy')
print(model2.structure)
```

```
((), (), (), (0,), (0,), (4,), (0, 4))
```

```
model2.plot()
```



	Inside/Outside	District	Neighborhood	Premise	Priority	CallDescription	Description
9082	Inside	SOUTHEASTERN	HIGHLANDTOWN	ROW/TOWNHO	MEDIUM	DISORDERLY	BURGLARY

```
#predict target variable
model2.predict([['Inside', 'SOUTHEASTERN', 'HIGHLANDTOWN', 'ROW/TOWNHO', 'MEDIUM',
        'DISORDERLY', None]])
```

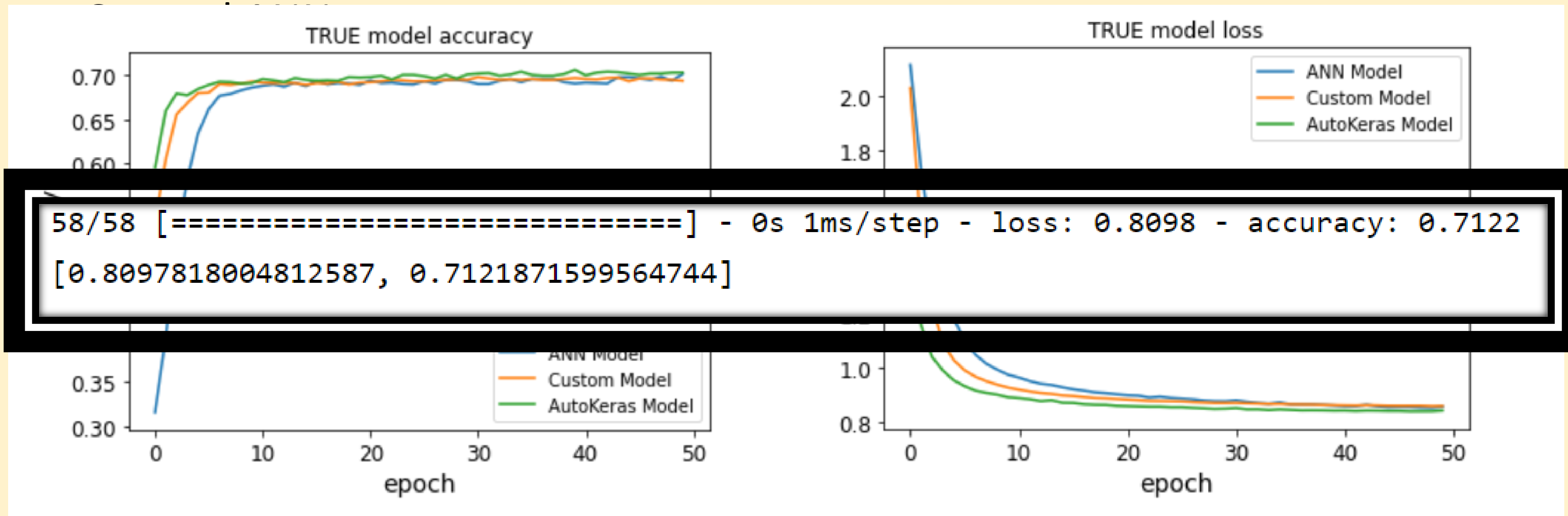
```
[array(['Inside', 'SOUTHEASTERN', 'HIGHLANDTOWN', 'ROW/TOWNHO', 'MEDIUM',
        'DISORDERLY', 'BURGLARY'], dtype=object)]
```

```
#impute missing feature
model2.predict([['Inside', 'SOUTHEASTERN', 'HIGHLANDTOWN', 'ROW/TOWNHO', None,
        'DISORDERLY', None]])
```

```
[array(['Inside', 'SOUTHEASTERN', 'HIGHLANDTOWN', 'ROW/TOWNHO', 'MEDIUM',
        'DISORDERLY', 'BURGLARY'], dtype=object)]
```

MODELING – Neural Networks

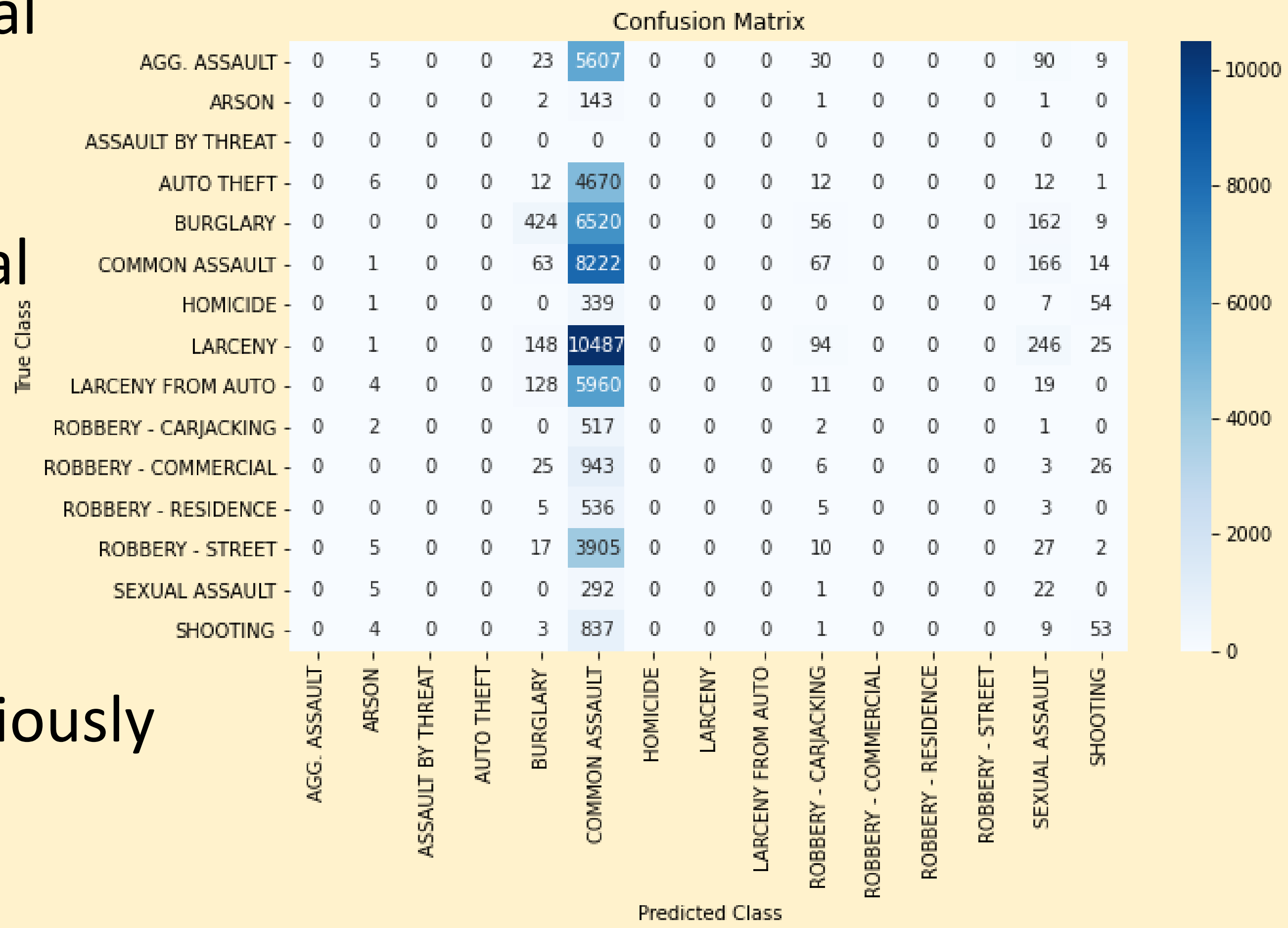
- [Neural networks are popular within the study of predictive policing]()
- Same split dataset trained on 3 neural networks




```
58/58 [==============================] - 0s 1ms/step - loss: 0.8098 - accuracy: 0.7122
[0.8097818004812587, 0.7121871599564744]
```

MODELLING

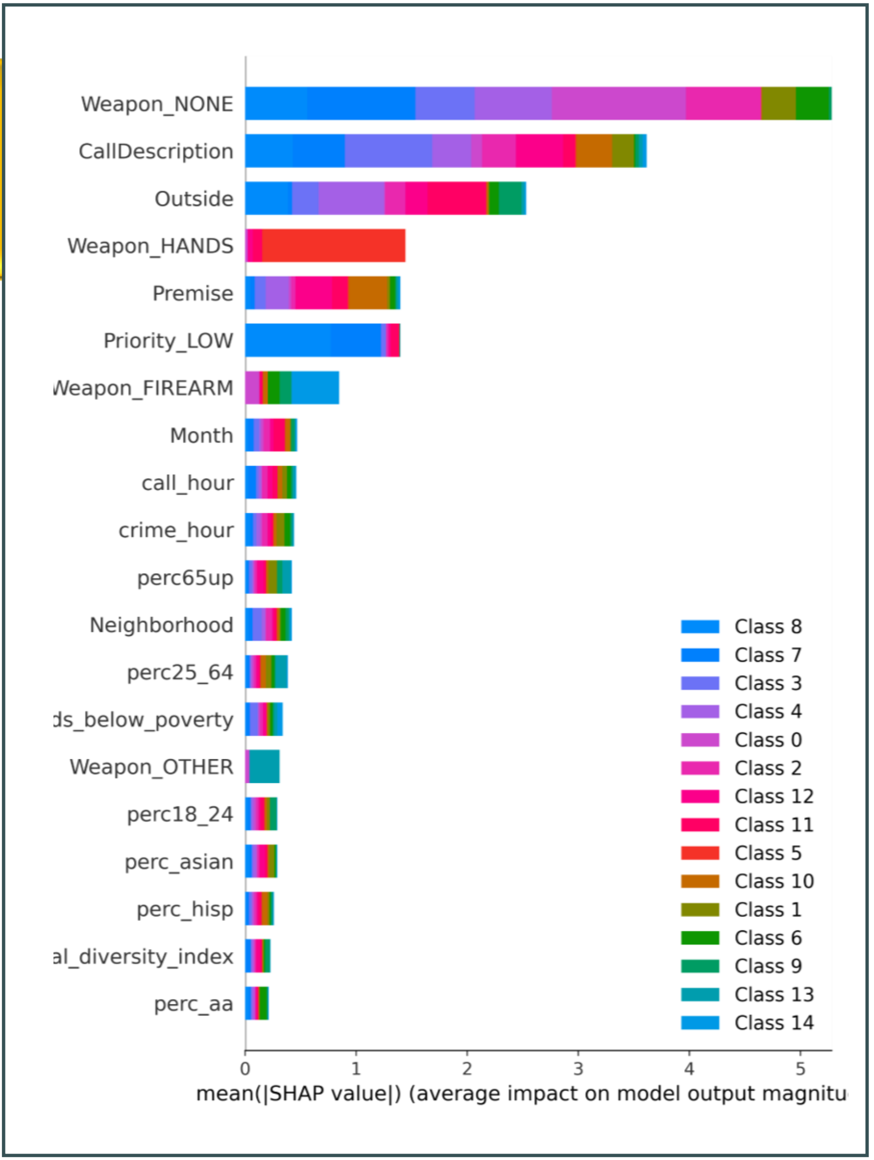
- LightGBM model tuned with original dataset + demographic data
- Use of Hyperopt and Mlflow
- Train results not better than original
- Test results much worse
    - Test data re-cleaned
    - Premises regrouped
    - Format made consistent
- Test results much worse than the previous model, but precision hilariously high



MODELLING

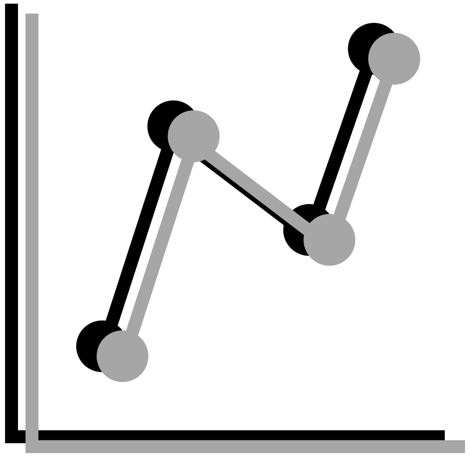
- 911 calls + demographic data used to build new model using 2015 data
- 911 calls very messy – many spelling mistakes and formatting errors
    - Very lengthy cleaning process

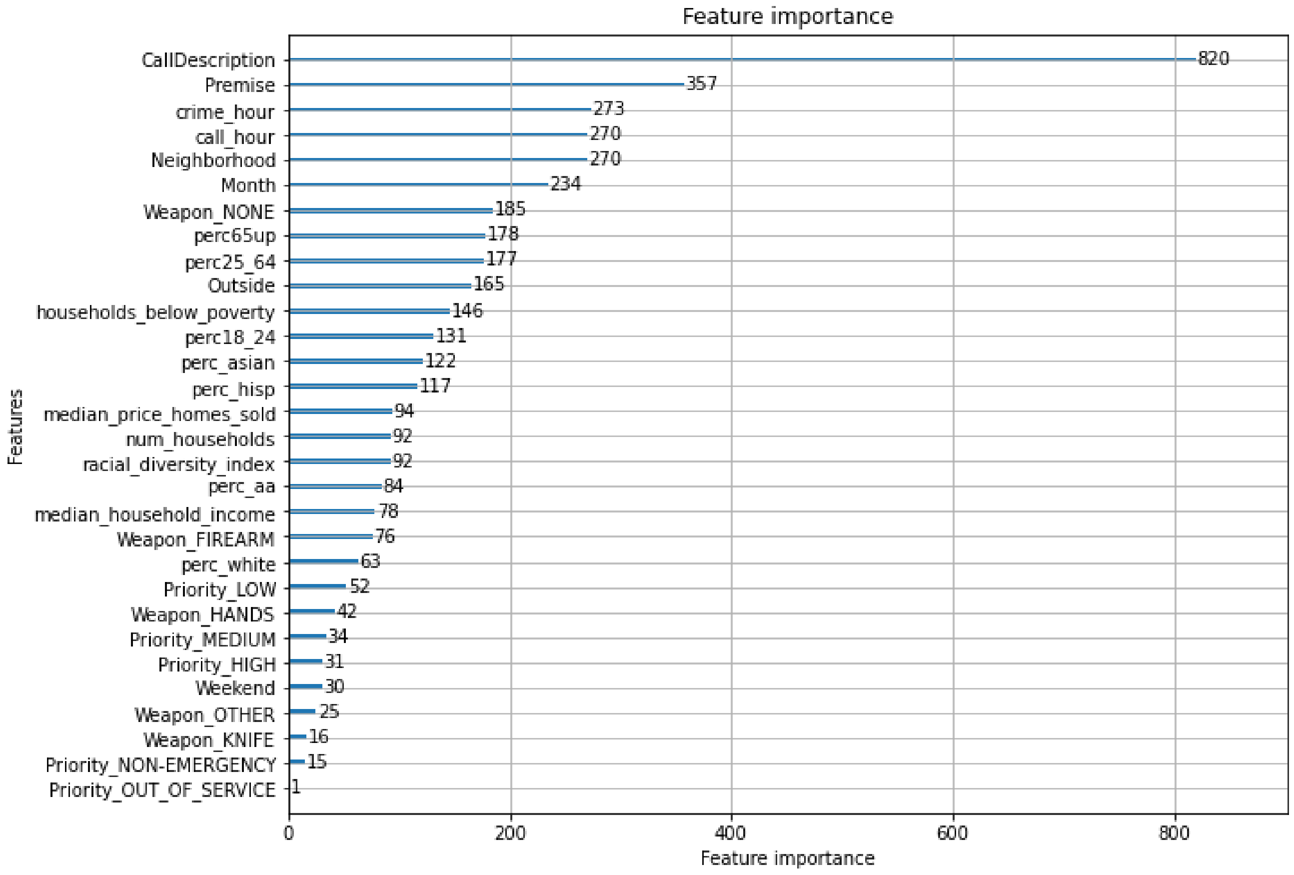
```
df['CallDescription'] = df['CallDescription'].replace('CHECK WELL BEIN', 'CHECK WELLBEING')
df['CallDescription'] = df['CallDescription'].replace('CHECKWELLBEING', 'CHECK WELLBEING')
df['CallDescription'] = df['CallDescription'].replace('CHEK WELL BEING', 'CHECK WELLBEING')
df['CallDescription'] = df['CallDescription'].replace('CHK WELL BEING', 'CHECK WELLBEING')
df['CallDescription'] = df['CallDescription'].replace('CK WELL BEING', 'CHECK WELLBEING')
df['CallDescription'] = df['CallDescription'].replace('CK WELL-BEING', 'CHECK WELLBEING')
df['CallDescription'] = df['CallDescription'].replace('WELL BEING CHECK', 'CHECK WELLBEING')
df['CallDescription'] = df['CallDescription'].replace('WELLBEING CHECK', 'CHECK WELLBEING')
```

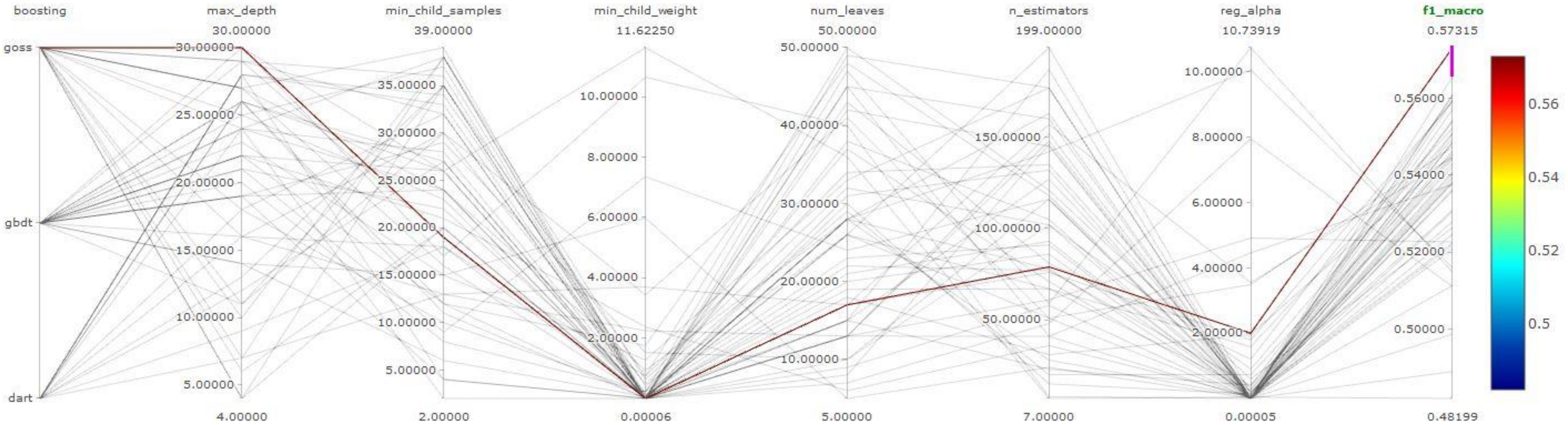


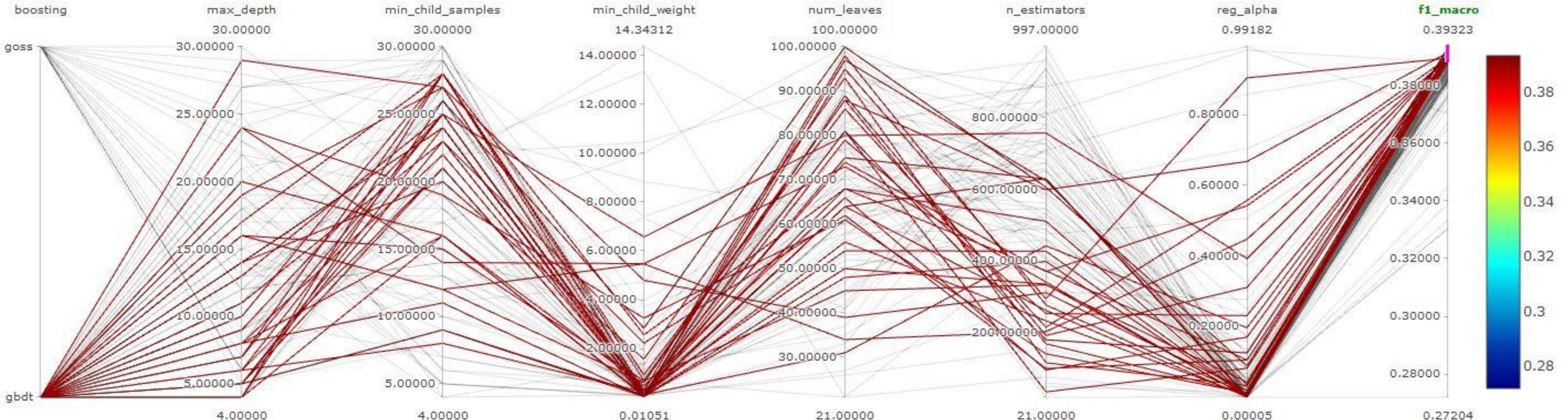
MODELLING

- Call description extremely valuable for predictions
- Holdout set performance very high






boosting
max_depth
min_child_samples
min_child_weight
num_leaves
n_estimators
reg_alpha
f1_macro
30.00000
39.00000
11.62250
50.00000
199.00000
10.73919
0.57315
goss
gbdt
dart
4.00000
2.00000
0.00006
5.00000
7.00000
0.00005
0.48199


boosting
max_depth
min_child_samples
min_child_weight
num_leaves
n_estimators
reg_alpha
f1_macro
30.00000
30.00000
14.34312
100.00000
997.00000
0.99182
0.39323
goss
gbdt
4.00000
4.00000
0.01051
21.00000
21.00000
0.00005
0.27204

RESULTS

- LightGBM model performs much much better on train CV

```
metrics.accuracy_score(predictions,Y_test)
```

```
0.1706410532287408
```

```
metrics.recall_score(predictions,Y_test,average="weighted",zero_division=0)
```

```
0.1706410532287408
```

```
metrics.precision_score(predictions,Y_test,average="weighted",zero_division=0)
```

Score Metrics

	Measure	Initial Model Score	New Model Score
0	F1	43.9500	76.8000
1	Accuracy	40.4000	77.2000
2	Recall	40.4000	77.2000
3	Precision	51.5600	78.5000

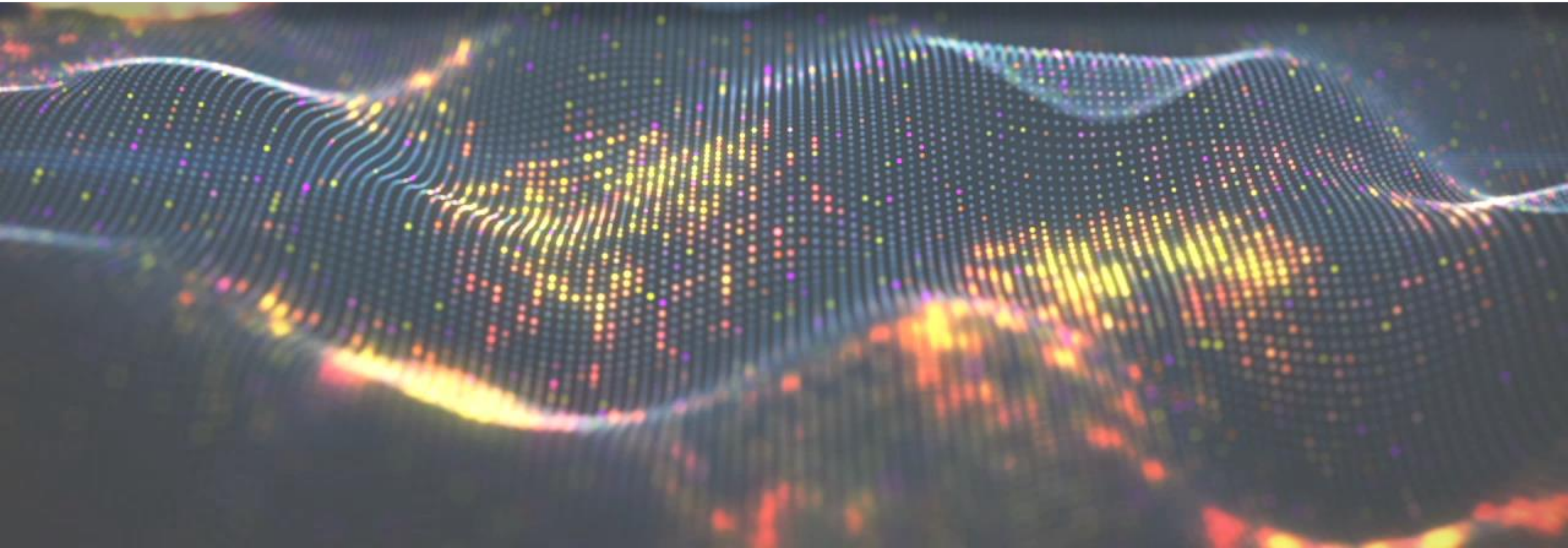
```
metrics.precision_score(predictions,y_final_test,average="weighted",zero_division=0)
```

```
0.7850126869124439
```

```
metrics.f1_score(predictions,y_final_test,average="weighted",zero_division=0)
```

```
0.7679508258568465
```

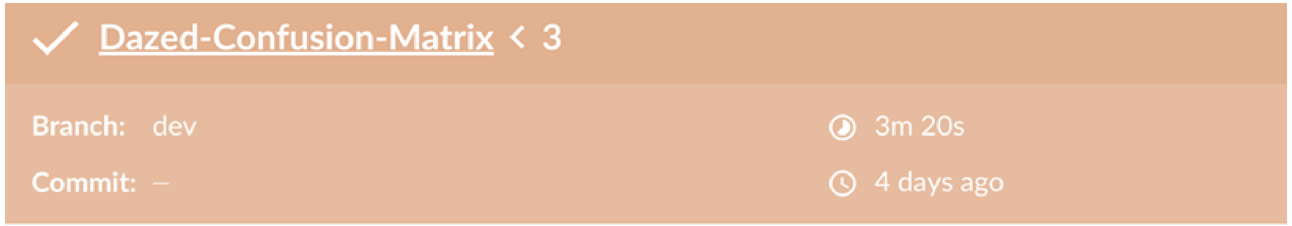
ARCHITECTURE



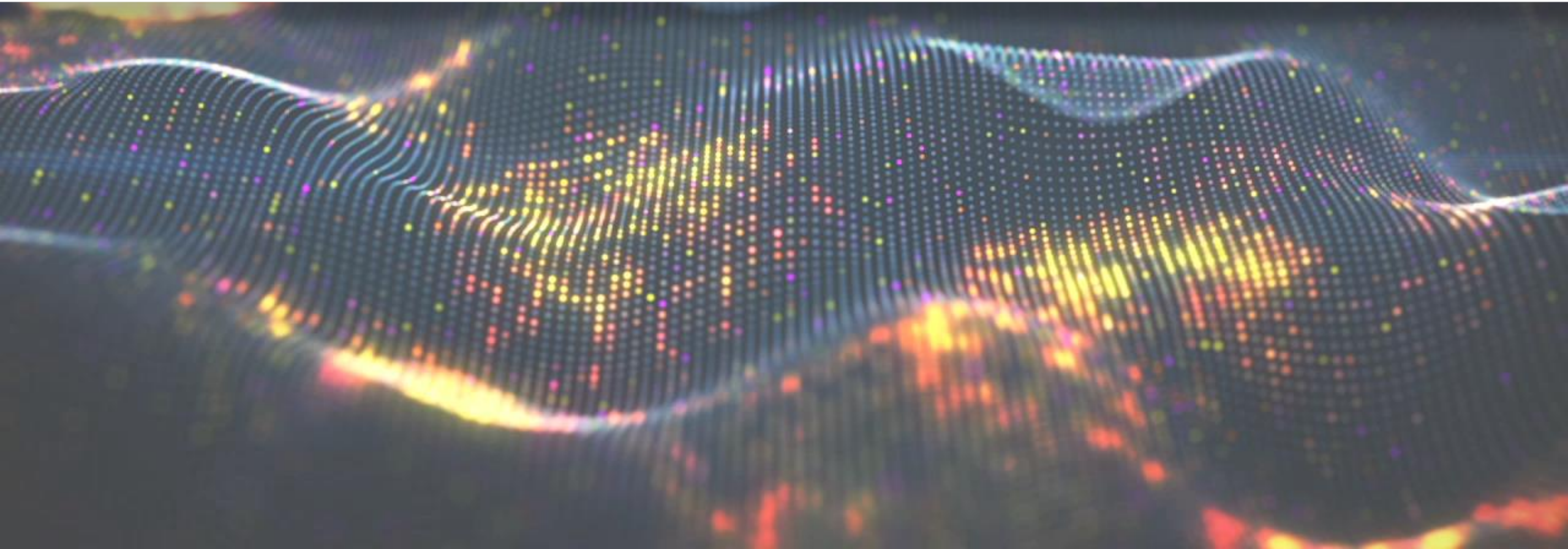
LIVE DEMO

LAUNCHING, MONITORING & MAINTENANCE

- Docker
  - Platform independent deployed
  - Experimented with Azure registry and web-apps
- Jenkins
  - Automated pipeline integrated with Git to make launching, monitoring, and maintenance easier
- PyTest
  - Lightweight Python library that can be used at bui time for test coverage
  - Experimented adding this as a check to inputs and modelling in the dashboard



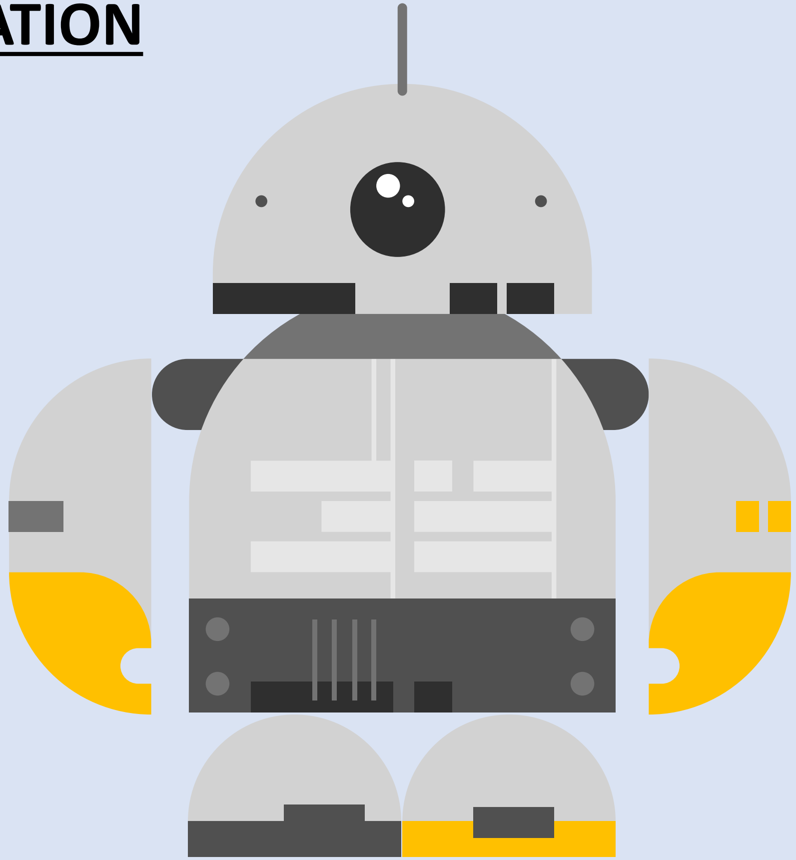
ACADEMIC



LESSONS LEARNED

APPLICATION

- Github
- ML Flow
- Hyperopt
- Jenkins
- Shap values
- CausalLIFT
- Auto keras
- Pomegranate
- Pytest
- Pycharm
- Docker
- Streamlit



USE CASE

- Bias is difficult to account for
- Predictive policing experiences a lot of criticism as unethical
- There is a lot of noise in this type of data
- Drift is a threat to validity here
- Data augmentation needs to continue for further improvements
- Additional work is required but would need cooperation from multiple levels of government organizations, and higher security clearance

FUTURE WORK

- Streams in real time
  - tensor processing unit
- Limit the bias within the model
- SME for the Bayesian Neural Network
- To advance the project would mean to acquire high security clearance data
  - End of open-source potential

Questions?
Questions?
Questions?
Questions?
Questions?
Questions?
Questions?
Questions?
Questions?
Questions?