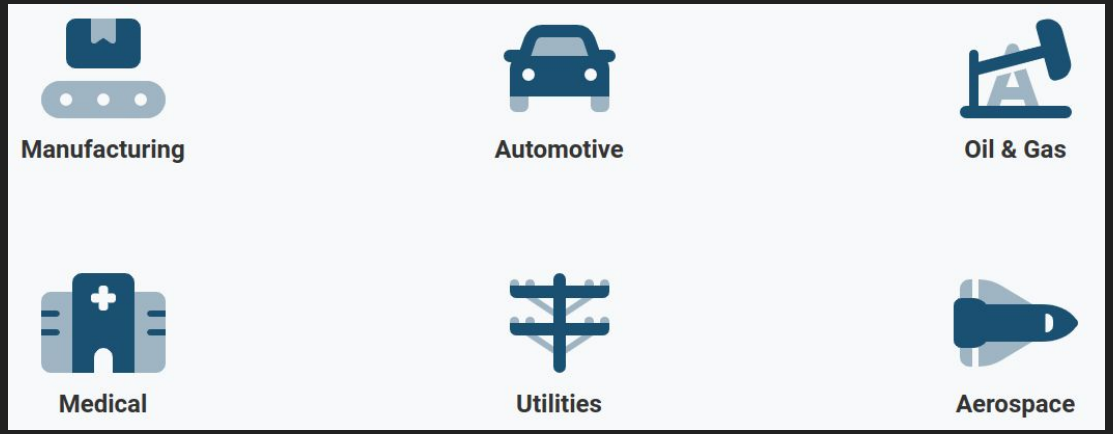
Can AI spot risky software in critical infrastructure?

Background

- Company aims to vet software packages used in critical infrastructure using FACT
- Clients need some way to ensure the safety of their operations

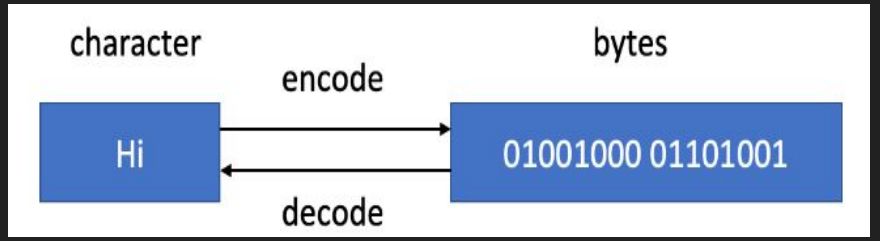


Background

Our Project

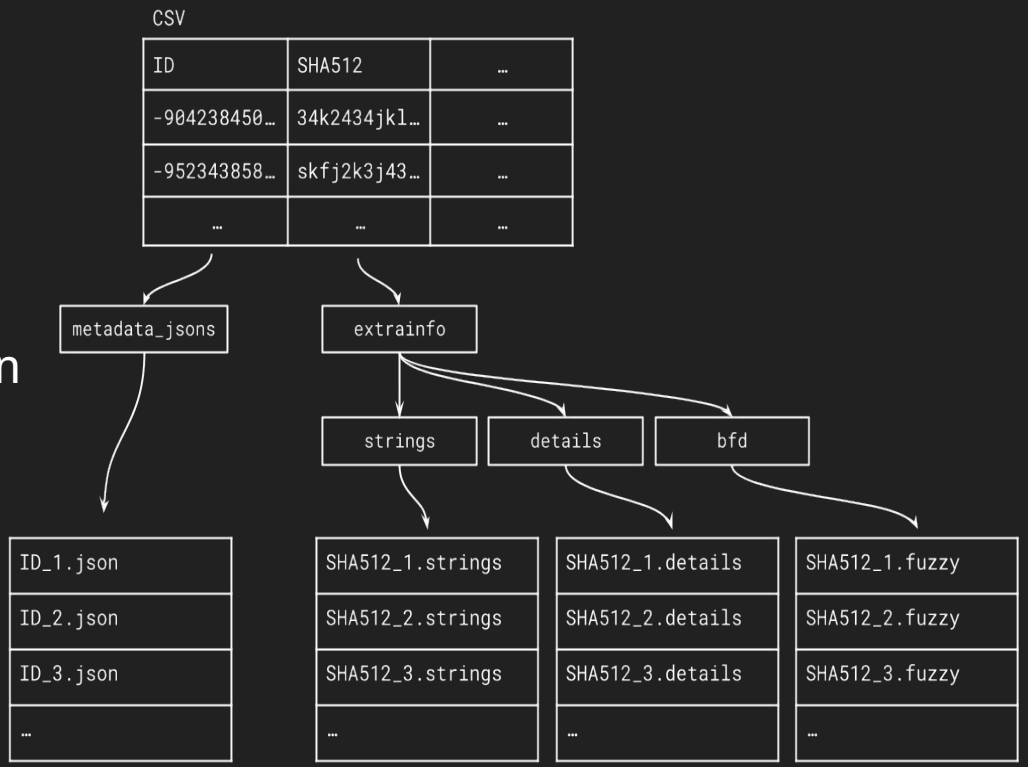
- Given Executable and Linkable Format files (ELFs)
- Binary files can be decoded to strings and other information
- Extracted information turned into metadata that the FACT system can use

We need to extract the most relevant information (package names, versions, etc) from these files!



Data

- CSV -> json files + extrainfo
- Current solution is manual regex and verification
- Not all files conform to industry standards from (package names in specific places)

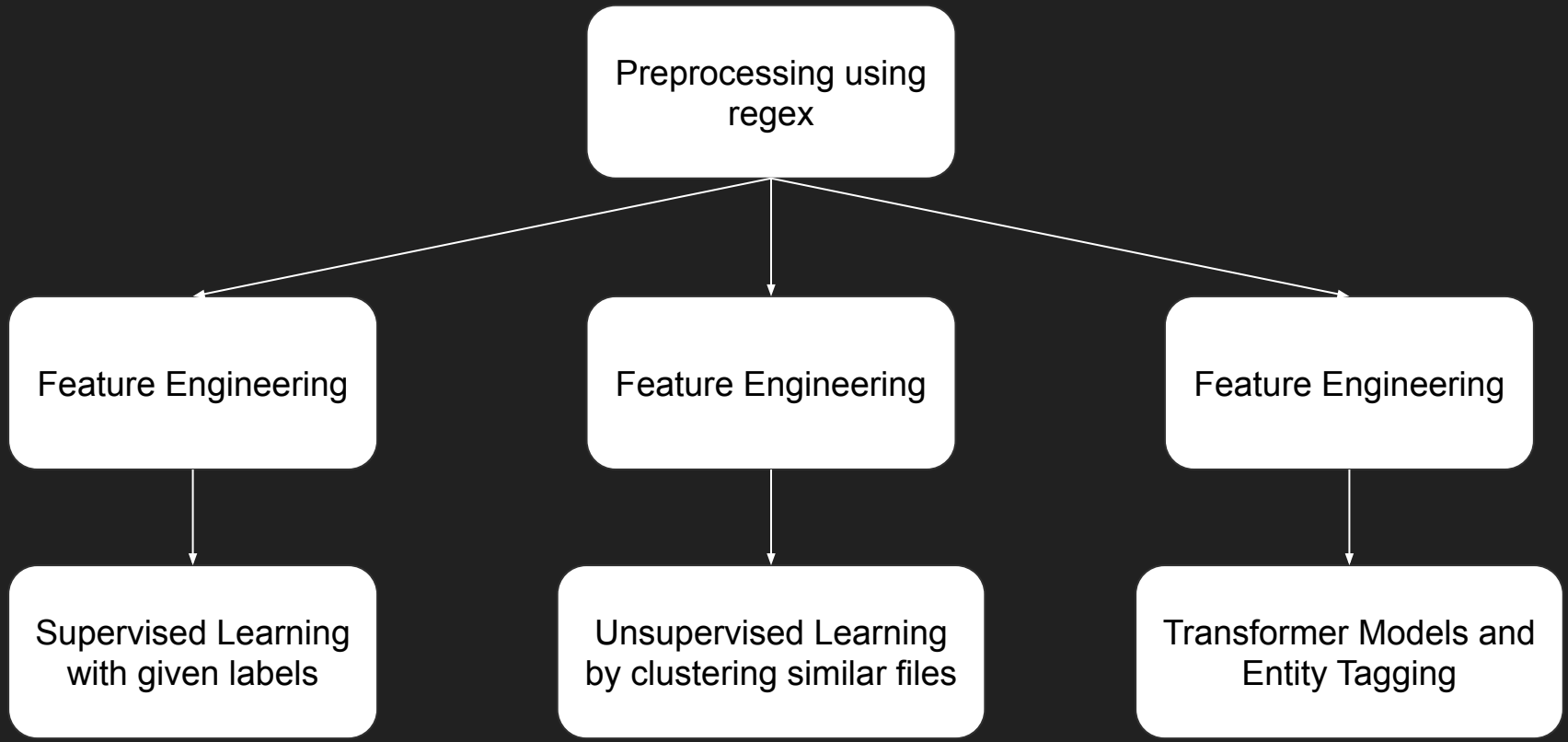


Products for the Partners

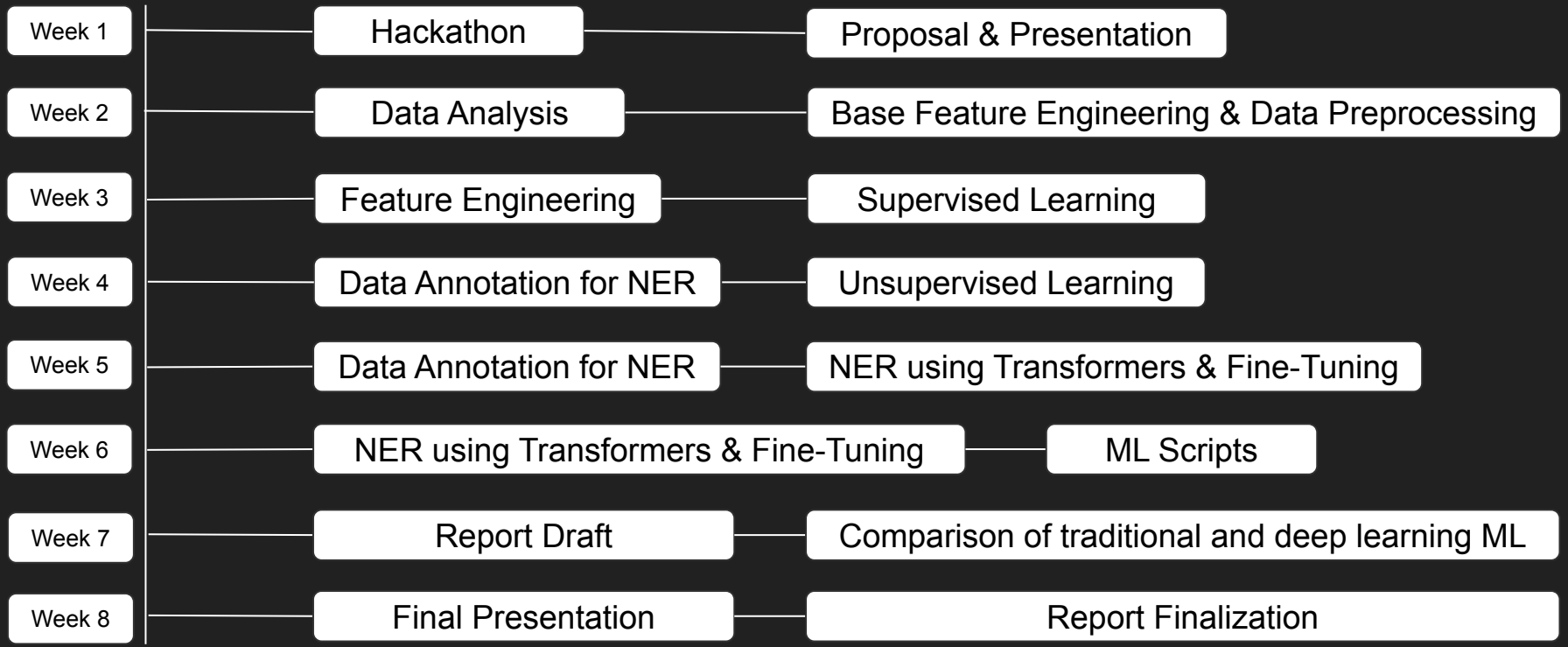
- Documentation on
  - Engineered features
  - Research feasibility of models to predict
  - Recommendations for further research
- Python Scripts (time-permitting)

https://www.diagnosticimaging.com/view/radiology-comic-long-report-covers-everything

The Science.


Preprocessing using regex
Feature Engineering
Feature Engineering
Feature Engineering
Supervised Learning with given labels
Unsupervised Learning by clustering similar files
Transformer Models and Entity Tagging

Project Steps



Tested Baseline Feature Engineering

Toy string file

```
$$$@@@- sqlite$$^7@@@@
@import - ,  package$$$
!!!! 1[]iptables--
@@@@$$$$$$$ $h?p????%2
sqlite o ?o .?o 3?oY0HE?
H?5B21?????? Lo cationmem
gvfs cmp __fprintf ?H ??AR
H?t$8?k D$?D$ ?D$$H?D$(H?
Plt.data.bss?@??D?\ $8LP
@@@@@@@@@L?T$8L??$?$$$$$@@@
bzip2??t?I9?vA?H?@??
D?\/usr/share/BB B(D0A8
]iptables?QTP???$1A(B
BBB,"HD$HT$Ht$0L$PLlo
neTable","_invocation_name
```

Package Name	Number of occurrences
sqlite	2
gvfs	1
iptables	2
bzip2	1
...	...

Tested Baseline Feature Engineering

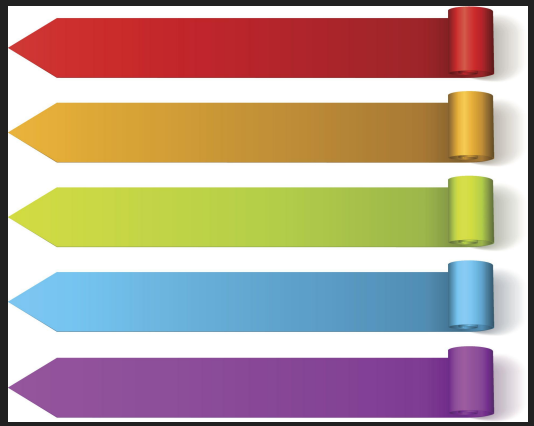
	Approach 1	Approach 2
Prediction Method for Package Label	Based on maximum occurrences	Logistic regression model
Prediction Accuracy	67.7%	96.8% (on the test set)

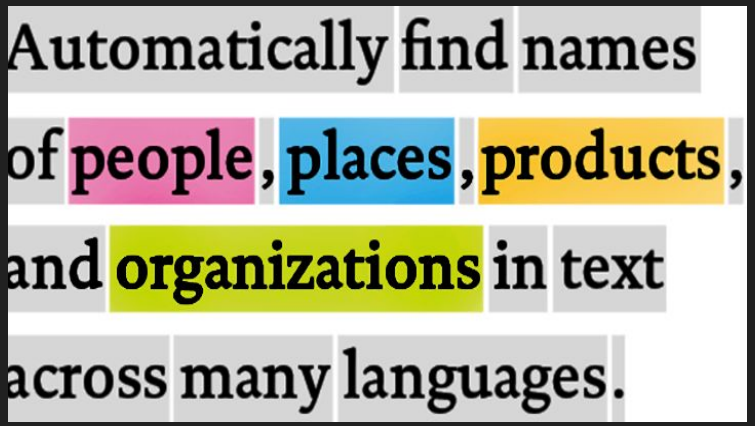
Challenges Faced

Heavy string preprocessing

Limited labels and package version numbers with baseline techniques

NER limitations





Need more sophisticated and advanced ML and NLP techniques!!!