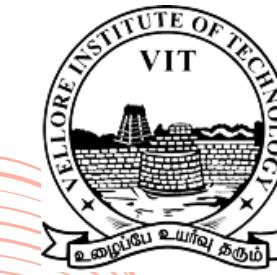


Project Review - 3



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

WINTER SEMESTER 2023-2024

CAPSTONE PROJECT

Course code - SWE1904

Guide

Dr. CHANDRASEGAR. T

Student

VIVEK R - 19MIS0184

TITLE OF THE PROJECT

**COMPREHENSIVE APPROACH
OF STATIC AND
DYNAMIC DATA ANALYTICS
USING AUTOML**

PROPOSED METHODOLOGY

- Developing automated solutions for data preprocessing, model training, algorithm selection, and hyperparameter optimization using AutoML techniques.
- Addressing class imbalance, redundant records, and missing guidelines in the CICIDS2017 dataset to improve its effectiveness as a benchmark for intrusion detection systems.
- Investigating techniques for handling IoT data and developing automated model updating procedures to maintain model performance over time.
- Comparing the performance of AutoML models with traditional machine learning
- Demonstrating the feasibility and effectiveness of AutoML in improving the efficiency and accuracy of data analytics tasks in both static and dynamic environments.

MODULE DESCRIPTION

Data Preprocessing Module

- Data Encoding
- Data Imputation
- Data Normalization
- Data Balancing
- Train-Test Split

MODULE DESCRIPTION

Model Learning Module

- Model Training
 - List of machine learning models:
 - Naive Bayes, KNN, Random Forest, LightGBM, ANN
- Model Evaluation
 - Performance metrics:
 - Accuracy, Precision, Recall, F1-score

MODULE DESCRIPTION

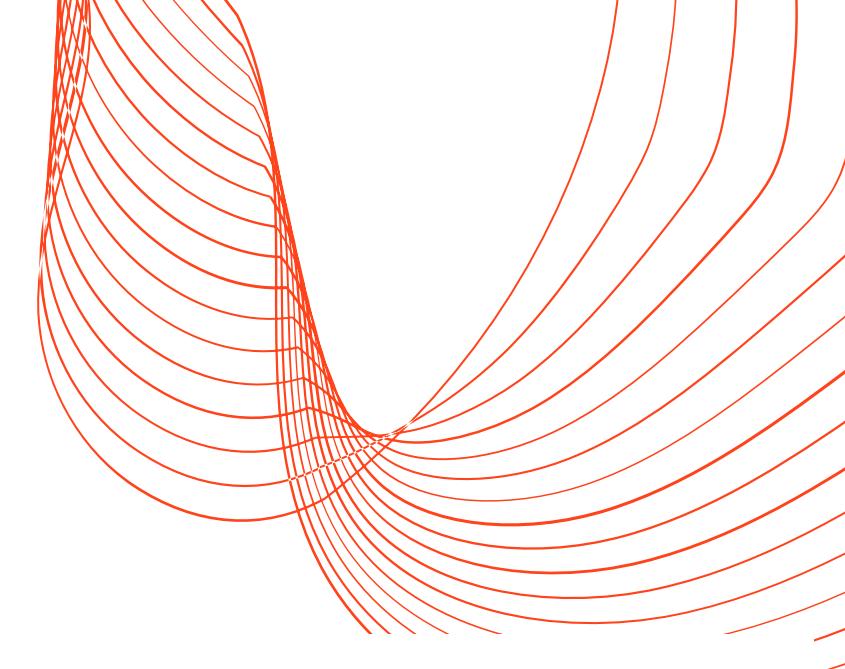
Model Selection Module

- Grid Search

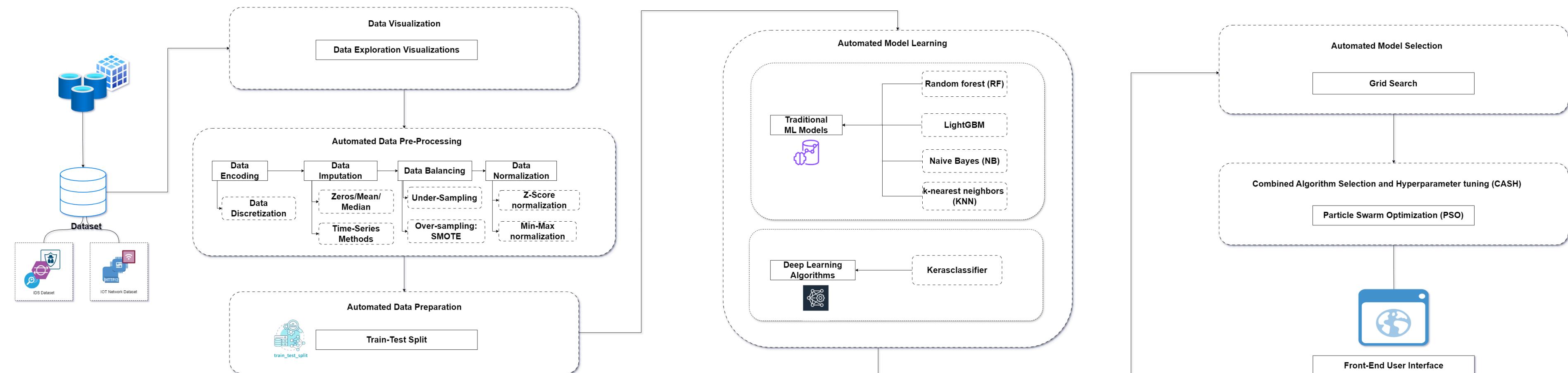
Combined Algorithm Selection and Hyperparameter Tuning (CASH) Module

- Particle Swarm Optimization (PSO)

SYSTEM ARCHITECTURE:



Architecture of Static and Dynamic Datasets



PROJECT IMPLEMENTATION

The screenshot shows a browser window with two tabs open, both titled "AutoML-Capstone Project". The left tab displays a sidebar with sections for "Notebooks", "Integrations >", "Files >", "Terminals", "Table of contents >", "Machine" (showing 34s 103% 1.8GB), and "Incoming connections" (set to Off). The right tab shows a notebook titled "AutoML-Capstone Project" with the status "Running 8 blocks". A button to "Stop execution" is visible. The main content area is titled "Automated Machine Learning - Dataset 1: CICIDS2017" and contains a section titled "Importing Libraries" with the following Python code:

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split,cross_val_score
4 import lightgbm as lgb
5 from sklearn.metrics import classification_report,confusion_matrix,accuracy_score, precision_score, recall_score, f1_score
6 from sklearn.preprocessing import LabelEncoder
7 from sklearn.ensemble import RandomForestClassifier
8 from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
9 from sklearn.neighbors import KNeighborsClassifier
10 from sklearn.svm import SVC
11 from sklearn.naive_bayes import GaussianNB
12 from sklearn.pipeline import Pipeline
13 from sklearn.model_selection import GridSearchCV
14 from scipy.stats import shapiro
15 from imblearn.over_sampling import SMOTE
16 import time
17 import warnings
18 warnings.filterwarnings('ignore')
```

PROJECT IMPLEMENTATION

ProjectWork AutoML-Capstone Project

Running 6 blocks

Stop execution

Share Create app

Notebooks +

Init ①

1.AutoML-Static-Data-Analytics

2.AutoML-Dynamic-Data-Analytics

Integrations > +

Files > 0 +

Terminals +

Table of contents >

Machine 1m 55% 10B

Basic 2 vCPUs, 5 GB memory

Python 3.11

Stop machine ...

Incoming connections Off

SQL cache Disabled

Auto shutdown After 1h

1 df = pd.read_csv("cic_0.01km.csv")
2 df

Flow Duration int64
-4 - 119999943

Total Length of F... 0 - 719406

Fwd Packet Length... 0 - 24820

Bwd Packet Length... 0 - 10000

Bwd Packet Length... 0 - 10000

Flow IUI mean no... 0 - 10000

Visualize

	Flow Duration	Total Length of F...	Fwd Packet Length...	Bwd Packet Length...	Bwd Packet Length...	Flow IUI mean no...
0	50833	0	0	0	0	50833
1	49	0	0	0	0	49
2	306	6	6	6	6	306
3	63041	65	65	124	124	63041
4	47682	43	43	59	59	47682
5	23896	68	44	100	100	7965.333333
6	343	62	31	159	159	114.333333
7	25271	164	41	57	57	3610.142857
8	1060	79	39	104	104	353.333333
9	224	82	41	169	169	74.66666667

28303 rows, showing 10 per page << < Page 1 of 2831 > >>

Data Preprocessing

PROJECT IMPLEMENTATION

The screenshot shows a Jupyter Notebook interface titled "AutoML-Capstone Project". The left sidebar displays a project structure with notebooks, integrations, files, terminals, and machine settings. The main area shows two code cells: one for LGBM Classifier Algorithm and one for Random Forest Algorithm.

LGBM Classifier Algorithm

```
1 lg = lgb.LGBMClassifier(verbose = -1)
2 lg.fit(X_train,y_train)
3 t1=time.time()
4 predictions = lg.predict(X_test)
5 t2=time.time()
6 print("Accuracy: "+str(round(accuracy_score(y_test,predictions),5)*100)+"%")
7 print("Precision: "+str(round(precision_score(y_test,predictions),5)*100)+"%")
8 print("Recall: "+str(round(recall_score(y_test,predictions),5)*100)+"%")
9 print("F1-score: "+str(round(f1_score(y_test,predictions),5)*100)+"%")
10
```

Output:

```
Accuracy: 99.806%
Precision: 99.292%
Recall: 99.733%
F1-score: 99.512%
```

Random Forest Algorithm

```
1 rf = RandomForestClassifier()
2 rf.fit(X_train,y_train)
```

PROJECT IMPLEMENTATION

ProjectWork > AutoML-Capstone Project

Running 2 blocks

Share Create app Stop execution

Model Selection

Method: Grid Search

```
1 # Create a pipeline
2 pipe = Pipeline([('classifier', GaussianNB())])
3
4 # Create space of candidate learning algorithms and their hyperparameters
5 search_space = [{"classifier": [GaussianNB()],
6                  "classifier": [KNeighborsClassifier()],
7                  "classifier": [RandomForestClassifier()],
8                  "classifier": [lgb.LGBMClassifier(verbose=-1)],
9                  "classifier": [KerasClassifier(build_fn=ANN, verbose=0)]},
10                 ]
11 clf = GridSearchCV(pipe, search_space, cv=5, verbose=0)
12 clf.fit(X, y)
```

GridSearchCV

```
GridSearchCV(cv=5, estimator=Pipeline(steps=[('classifier', GaussianNB())]),
param_grid=[{"classifier": [GaussianNB()],
("classifier": [KNeighborsClassifier()],
("classifier": [RandomForestClassifier()],
("classifier": [LGBMClassifier(verbose=-1)],
("classifier": [

estimator: Pipeline



```
GaussianNB()
GaussianNB()
```



Notebooks +



- Init (1)
- 1.AutoML-Static-Data-Analytics
- 2.AutoML-Dynamic-Data-Analytics



Integrations > +



Files > +



Terminals > +



Table of contents >



Machine 5% 4% 1.2GB



Basic  
2 vCPUs, 5 GB memory



Python 3.11



Stop machine



Incoming connections Off



SQL cache Disabled



Auto shutdown After 1h


```

PROJECT IMPLEMENTATION

ProjectWork v AutoML-Capstone Project

Notebooks +

- Init ⓘ
- 1.AutoML-Static-Data-Analytics
- 2.AutoML-Dynamic-Data-Analyti...

Integrations > +

Files > +

Terminals +

Table of contents >

Machine 12m 0% 1.3GB

Basic 2 vCPUs, 5 GB memory

Python 3.11

Stop machine ...

Incoming connections Off

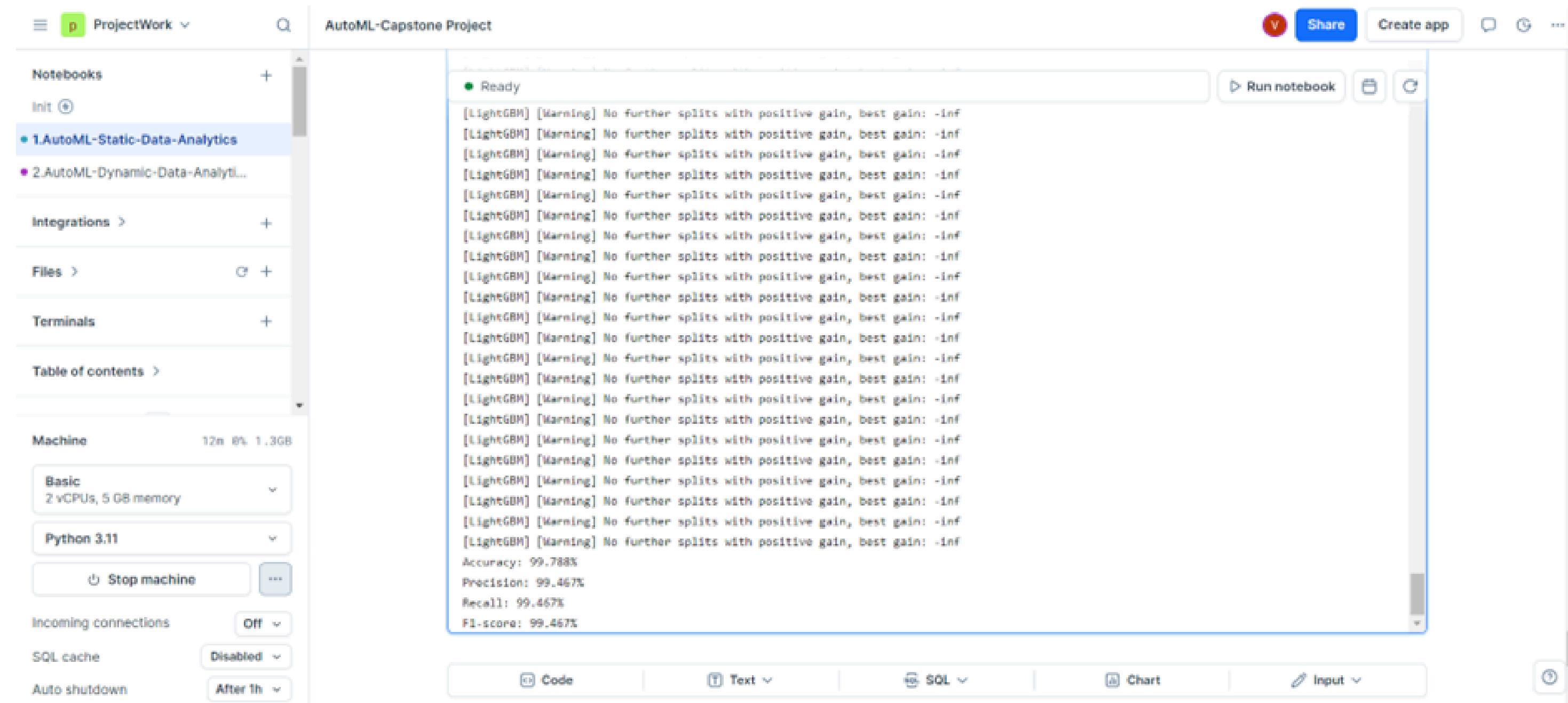
SQL cache Disabled

Auto shutdown After 1h

Ready

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
Accuracy: 99.788%
Precision: 99.467%
Recall: 99.467%
F1-score: 99.467%
```

Code Text SQL Chart Input



PERFORMANCE METRICS: ACCURACY

Accuracy measures how many predictions were correct overall.

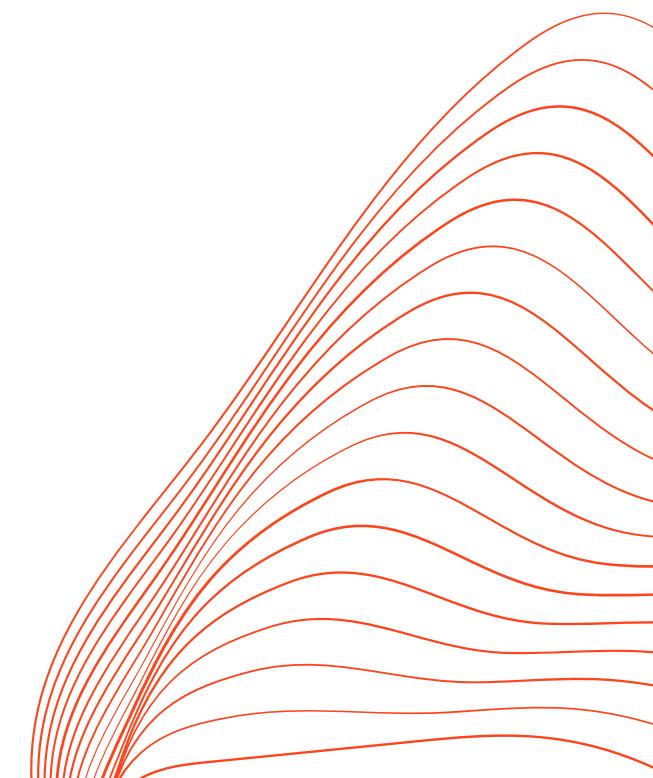
$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Static Dataset

Model Accuracy	Percentage
LGBM Classifier	99.753 %
Random Forest	75.729 %
Naive Bayes	98.728 %
k-nearest neighbors (KNN)	92.475 %
KerasClassifier Model	99.753 %

Dynamic Dataset

Model Accuracy	Percentage
LGBM Classifier	99.92%
Random Forest	99.839%
Naive Bayes	70.184%
k-nearest neighbors (KNN)	99.280%
KerasClassifier Model	92.475 %



PERFORMANCE METRICS: PRECISION

Precision measures proportion predictions that were actually correct.

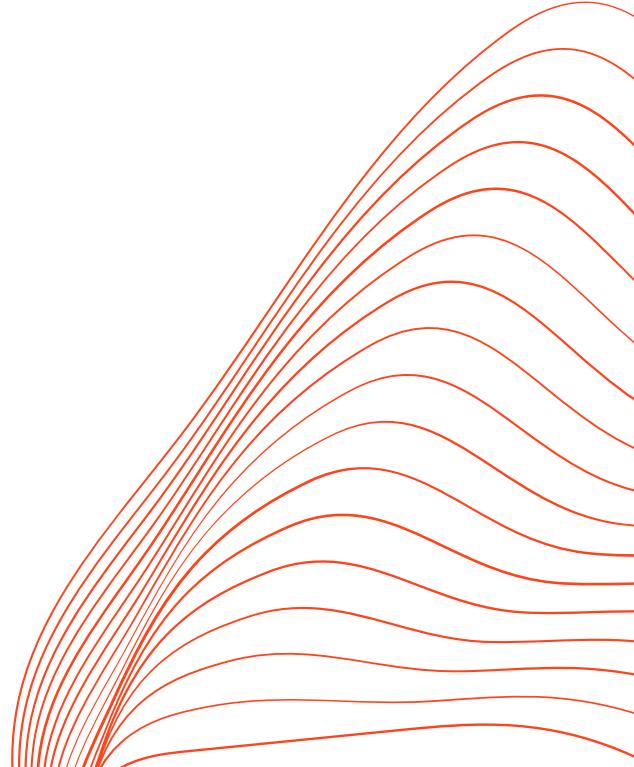
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Static Dataset

Model Precision	Percentage
LGBM Classifier	99.378 %
Random Forest	99.554 %
Naive Bayes	44.891 %
k-nearest neighbors (KNN)	95.584 %
KerasClassifier Model	73.378 %

Dynamic Dataset

Model Precision	Percentage
LGBM Classifier	99.914%
Random Forest	99.83%
Naive Bayes	99.875%
k-nearest neighbors (KNN)	99.744%
KerasClassifier Model	92.475 %



PERFORMANCE METRICS: RECALL

Recall is a measure of proportion of actual positives that were correctly predicted.

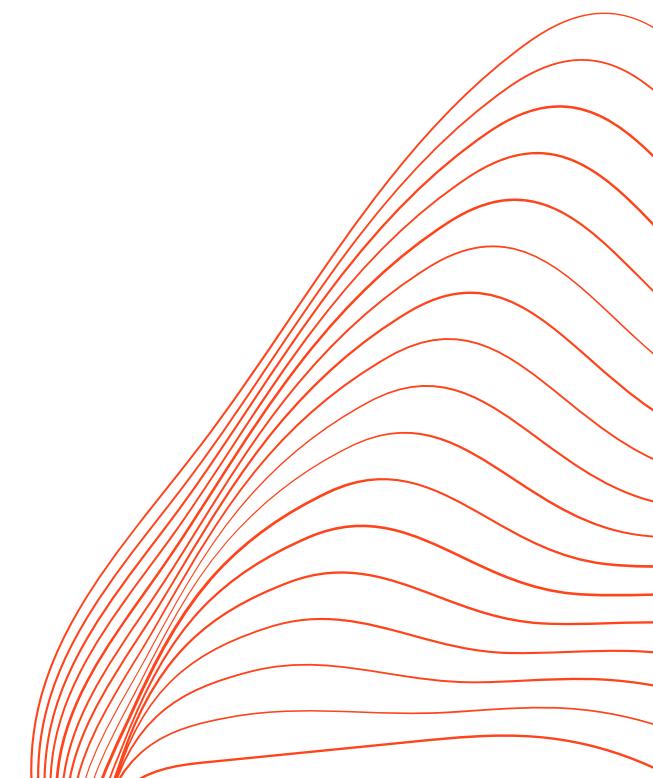
$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Static Dataset

Model Accuracy	Percentage
LGBM Classifier	99.753 %
Random Forest	75.729 %
Naive Bayes	98.728 %
k-nearest neighbors (KNN)	92.475 %
KerasClassifier Model	99.753 %

Dynamic Dataset

Model Accuracy	Percentage
LGBM Classifier	99.92%
Random Forest	99.839%
Naive Bayes	70.184%
k-nearest neighbors (KNN)	99.280%
KerasClassifier Model	92.475 %



PERFORMANCE METRICS: F1 SCORE

F1 score is harmonic mean of the Recall and Precision scores, therefore balancing their respective strengths

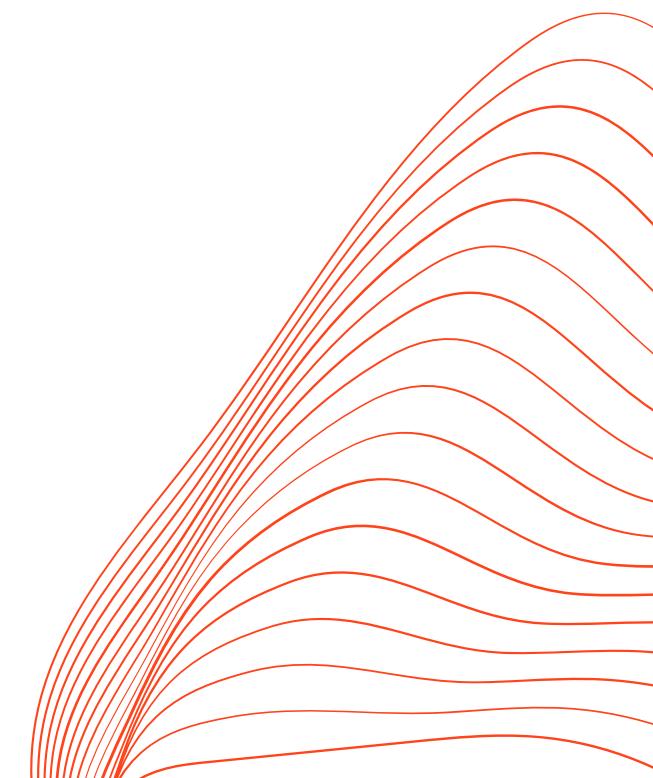
$$F1 = 2 \times TP / 2 \times TP + FP + FN$$

Static Dataset

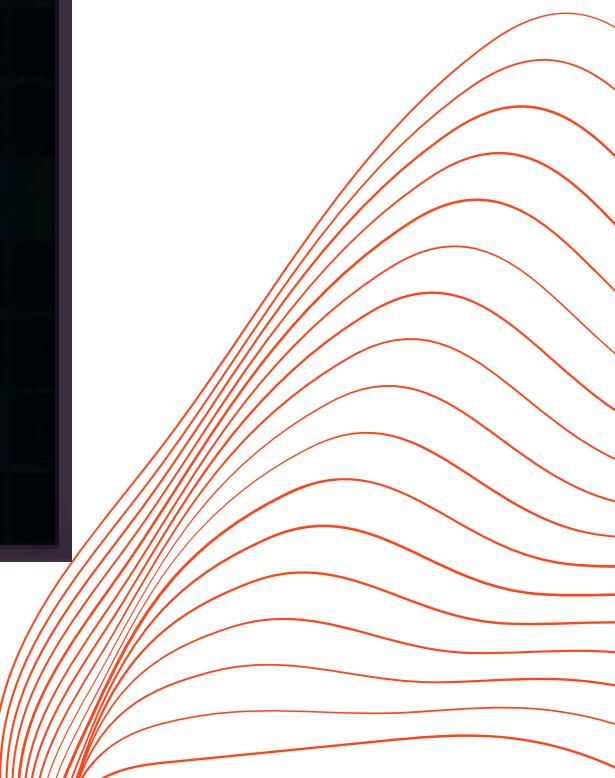
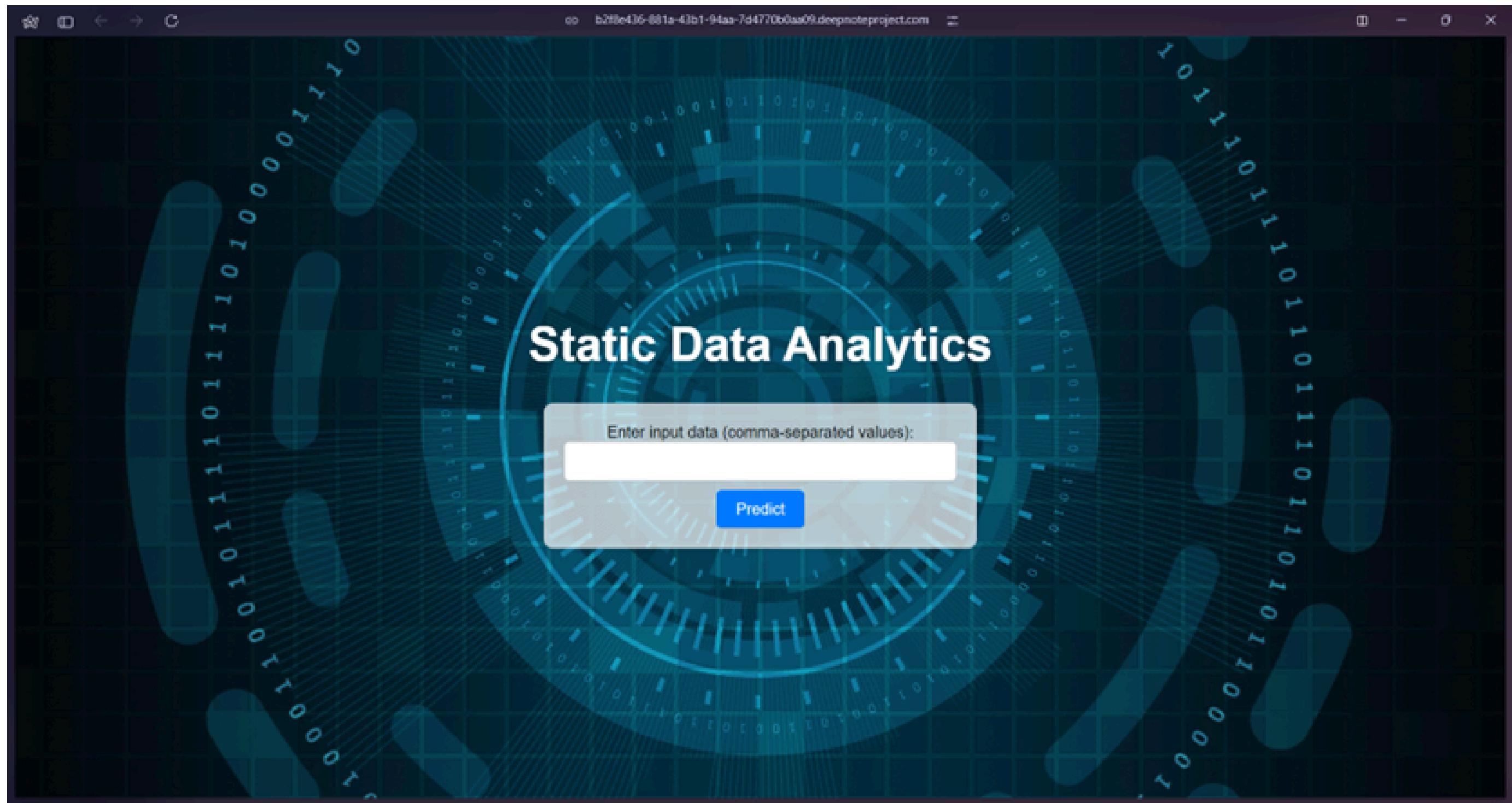
Model F1 score	Percentage
LGBM Classifier	99.788 %
Random Forest	99.753 %
Naive Bayes	61.426 %
k-nearest neighbors (KNN)	96.842%
KerasClassifier Model	83.740%

Dynamic Dataset

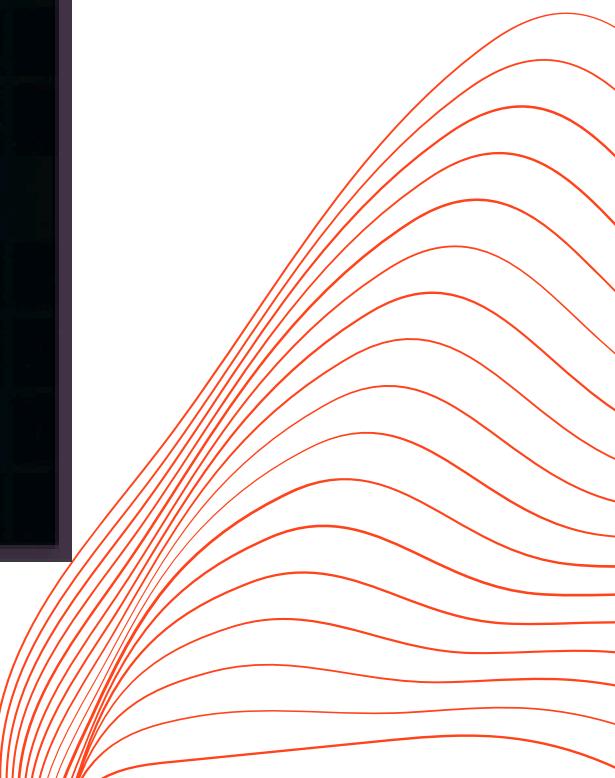
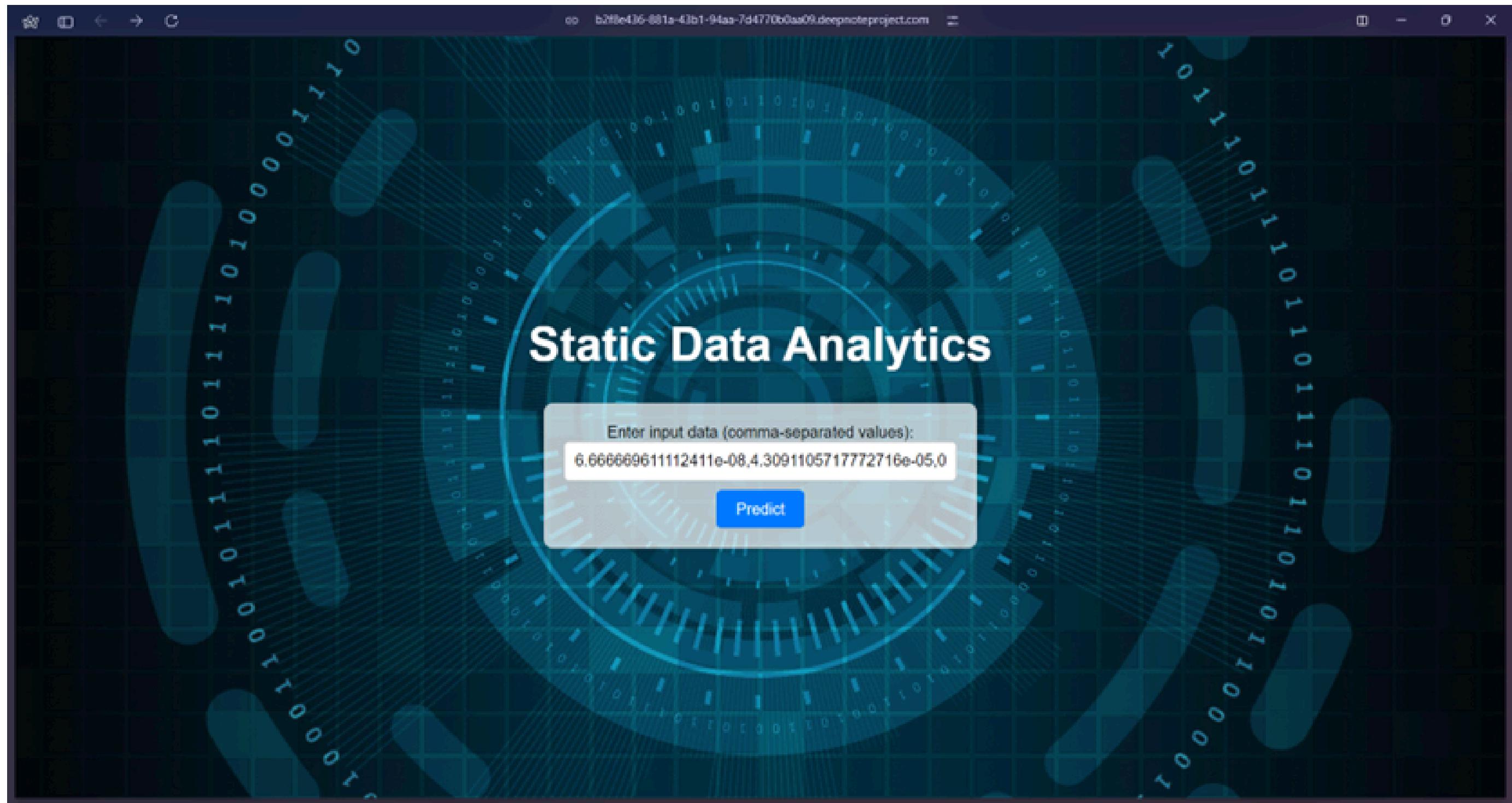
Model F1 score	Percentage
LGBM Classifier	99.957%
Random Forest	99.914%
Naive Bayes	81.133%
k-nearest neighbors (KNN)	99.616%
KerasClassifier Model	92.475 %



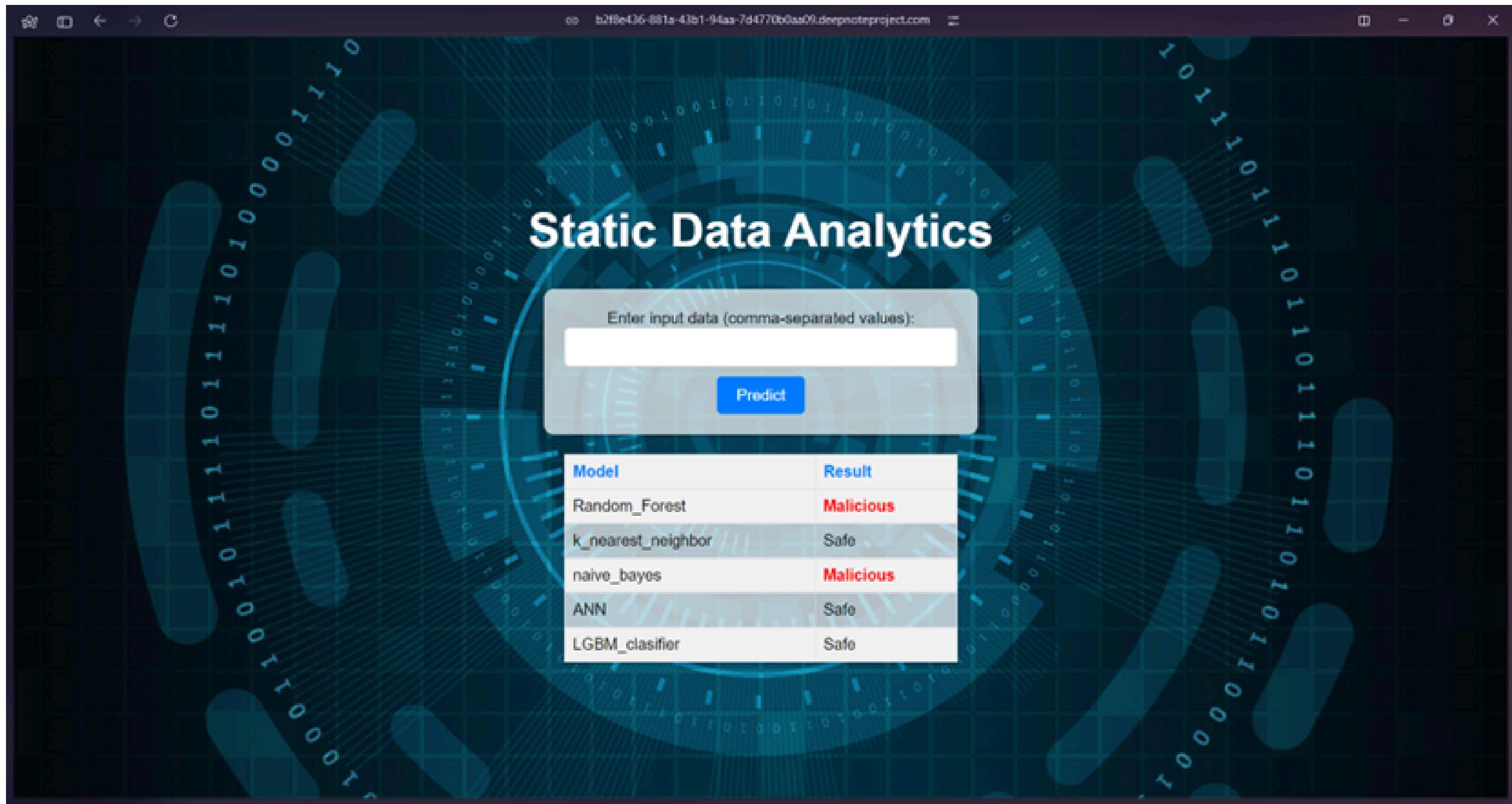
STATIC DATA ANALYTICS – FRONTEND USER INTERFACE (UI)



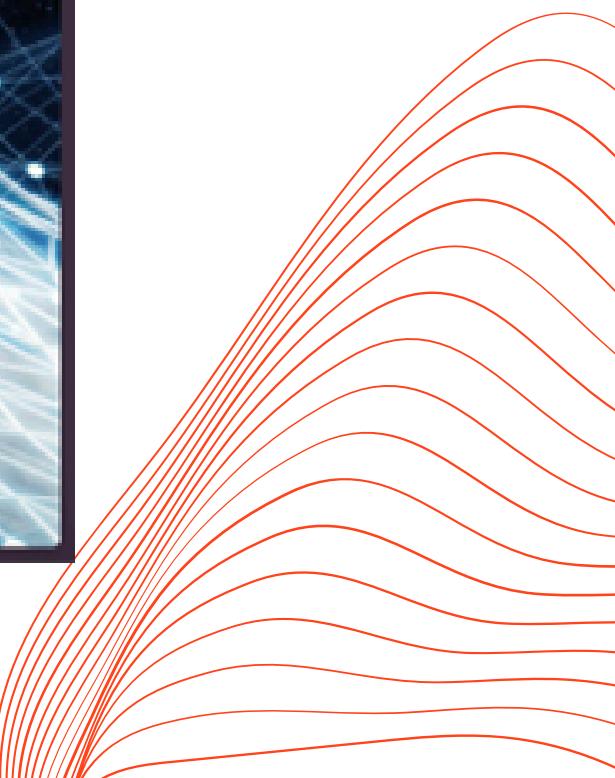
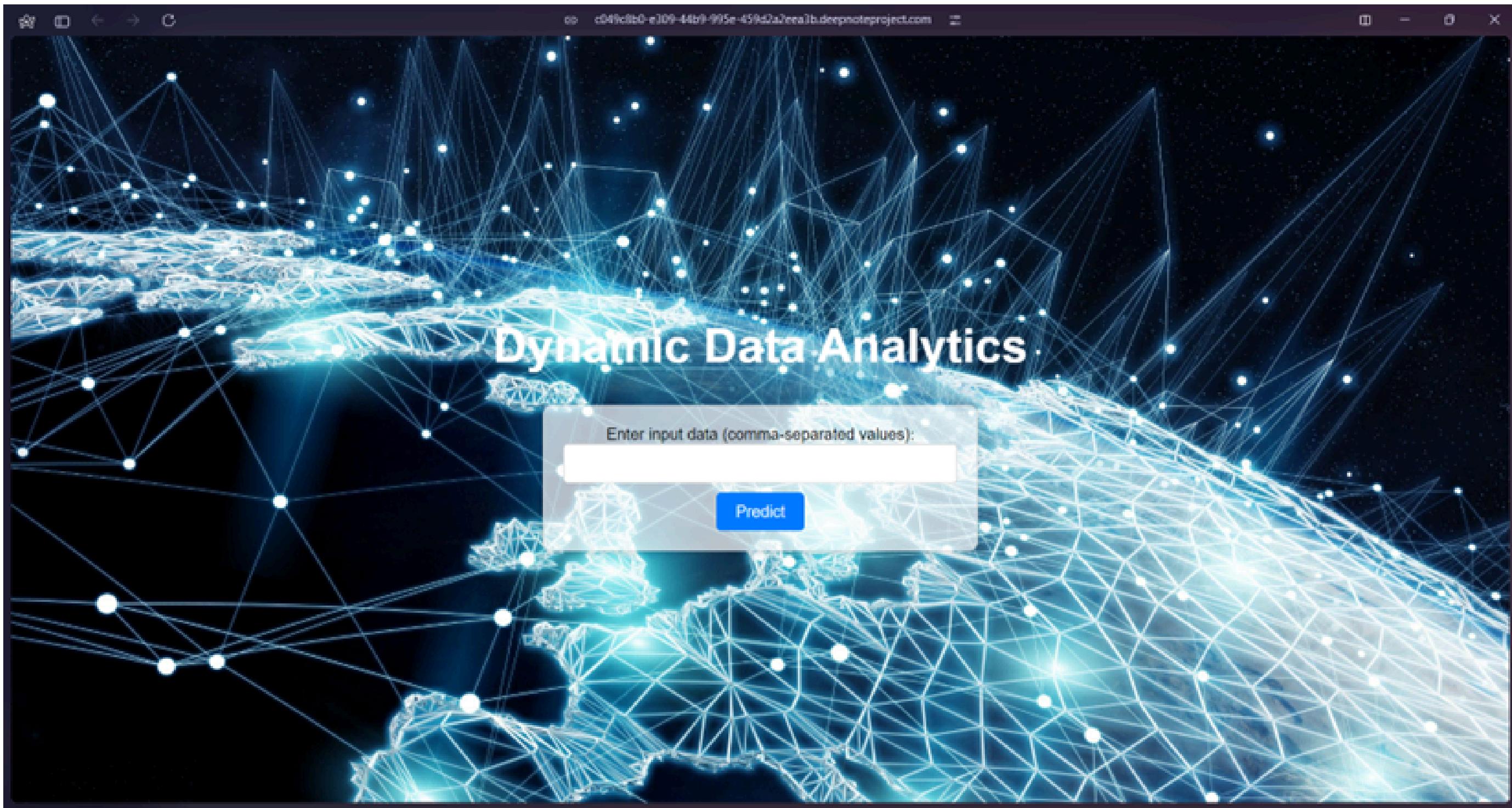
STATIC DATA ANALYTICS – FRONTEND USER INTERFACE (UI)



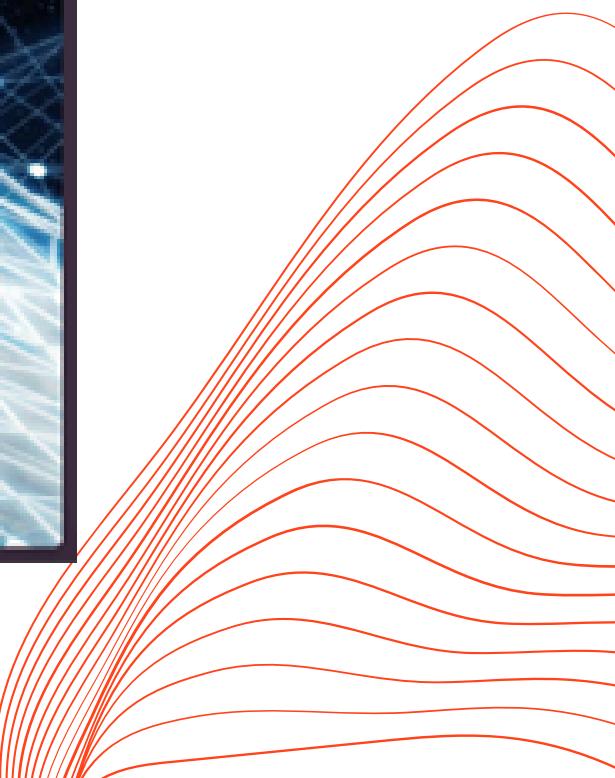
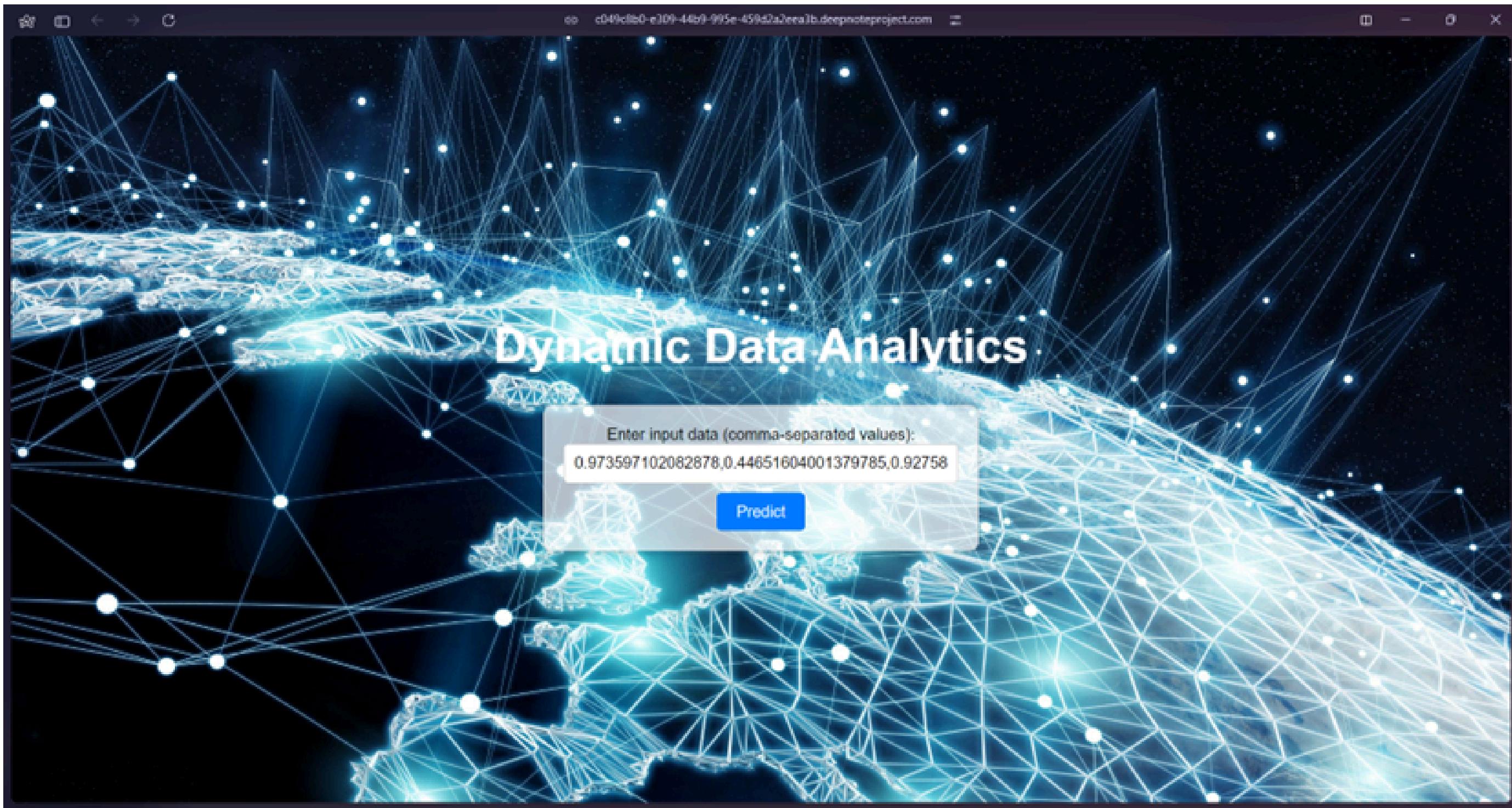
STATIC DATA ANALYTICS – FRONTEND USER INTERFACE (UI)



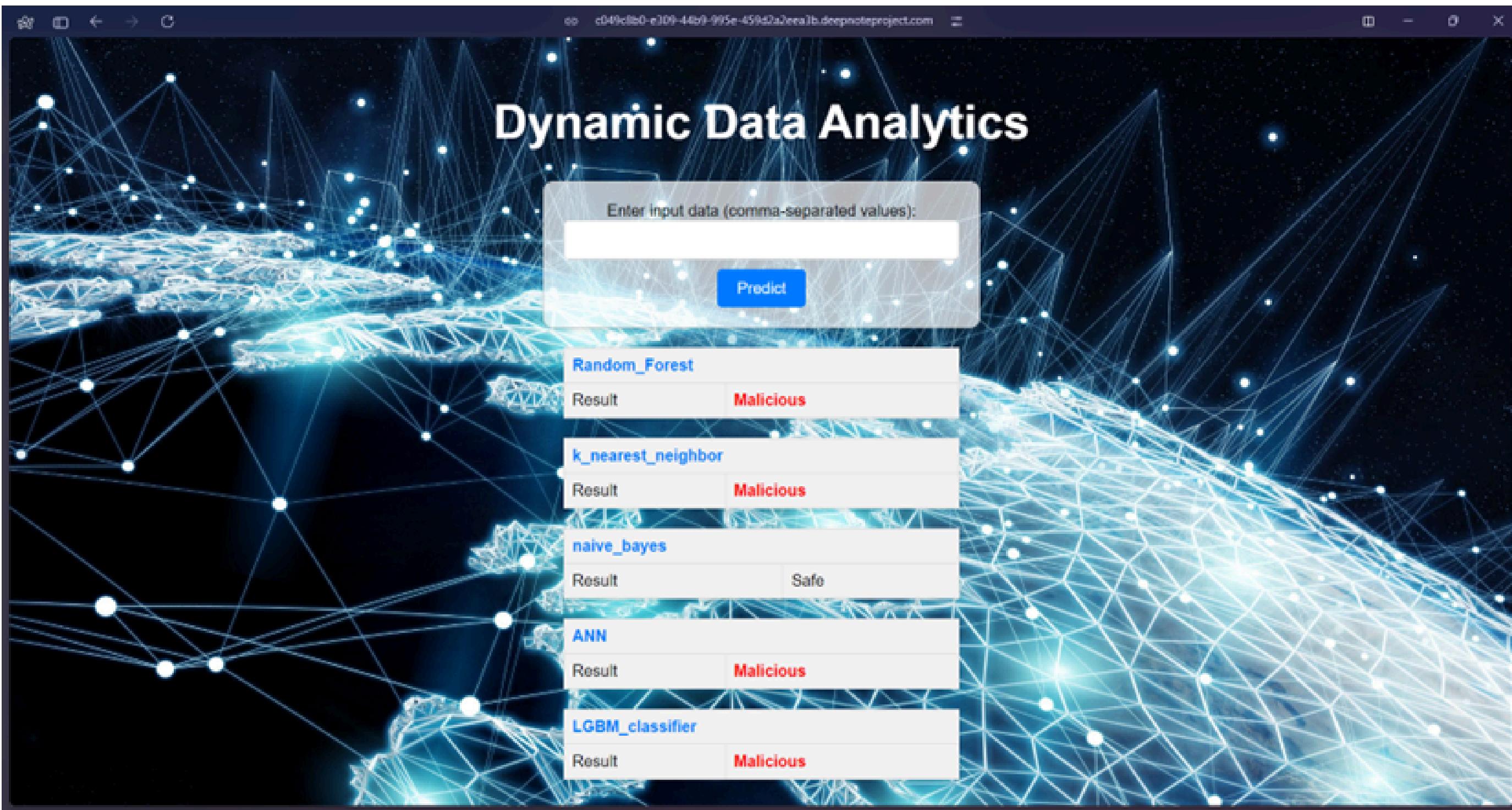
DYNAMIC DATA ANALYTICS – FRONTEND USER INTERFACE (UI)



DYNAMIC DATA ANALYTICS – FRONTEND USER INTERFACE (UI)

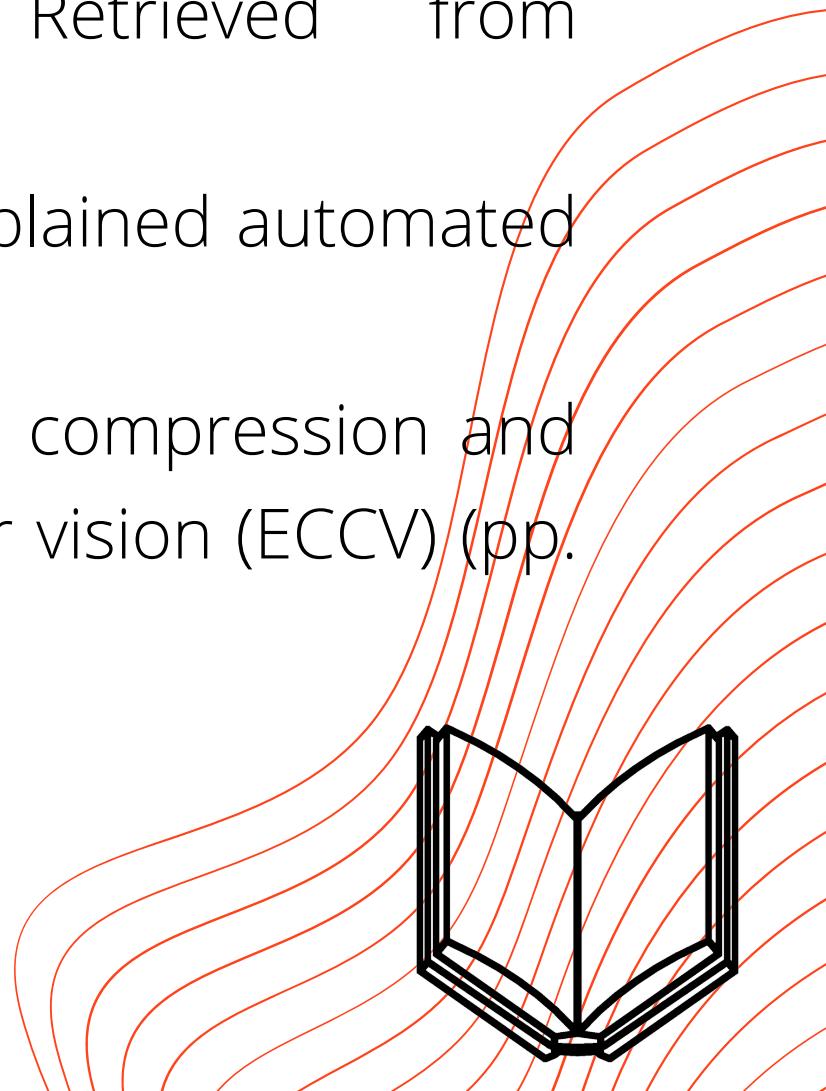


DYNAMIC DATA ANALYTICS – FRONTEND USER INTERFACE (UI)



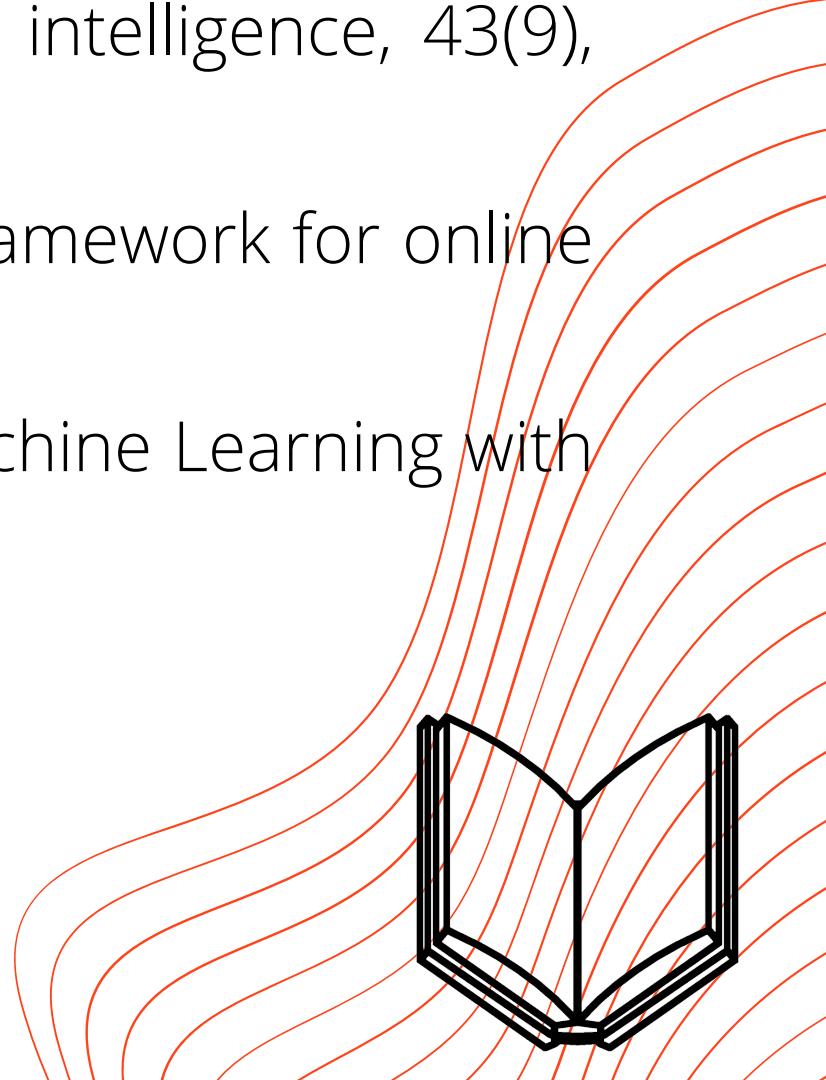
REFERENCES:

- [1] Yang, L., & Shami, A. (2022). IoT data analytics in dynamic environments: From an automated machine learning perspective. *Engineering Applications of Artificial Intelligence*, 116, 105366.
- [2] Singh, A., Amutha, J., Nagar, J., Sharma, S., & Lee, C. C. (2022). AutoML-ID: Automated machine learning model for intrusion detection using wireless sensor network. *Scientific Reports*, 12(1), 9074.
- [3] Lindstedt, H. (2022). Methods for network intrusion detection: Evaluating rule-based methods and machine learning models on the CIC-IDS2017 dataset (Dissertation). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-479347>
- [4] Garouani, M., Ahmad, A., Bouneffa, M., & Hamlich, M. (2022). AMLBID: an auto-explained automated machine learning tool for big industrial data. *SoftwareX*, 17, 100919.
- [5] He, Y., Lin, J., Liu, Z., Wang, H., Li, L. J., & Han, S. (2018). Amc: Automl for model compression and acceleration on mobile devices. In Proceedings of the European conference on computer vision (ECCV) (pp. 784-800).



REFERENCES:

- [6] He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 106622.
- [7] Lee, J., Ahn, S., Kim, H., & Lee, J. R. (2022). Dynamic Hyperparameter Allocation under Time Constraints for Automated Machine Learning. *Intelligent Automation & Soft Computing*, 31(1).
- [8] Wever, M., Tornede, A., Mohr, F., & Hüllermeier, E. (2021). AutoML for multi-label classification: Overview and empirical evaluation. *IEEE transactions on pattern analysis and machine intelligence*, 43(9), 3037-3054.
- [9] Celik, B., Singh, P., & Vanschoren, J. (2023). Online automl: An adaptive automl framework for online learning. *Machine Learning*, 112(6), 1897-1921.
- [10] Zhang, S., Gong, C., Wu, L., Liu, X., & Zhou, M. (2023). AutoML-GPT: Automatic Machine Learning with GPT. *arXiv preprint arXiv:2305.02499*



THANK YOU