Near-Earth-Objects (NEO): Are We Doomed?

An analysis by Jacqueline Tsodikova, Alejandra Magana, Robert Janke, Michael Albers, Fred Jambor

What are we analyzing?

Our group is researching the likelihood an asteroid or comet will potentially harm our planet. Using historical data, we will determine what thresholds are used to predict which NEOs are the most hazardous to Earth.

Why this topic?

A new movie on Netflix, *Don't Look Up*, was just released that had to do with a comet approaching Earth and scientists trying to warn the public about it.

QUESTIONS

1. Which NEOs will be the closest to approach Earth?
2. Can we predict potentially hazardous objects in the future?
3. Which NEOs are the most potentially hazardous?

Technologies Used:

Dashboard

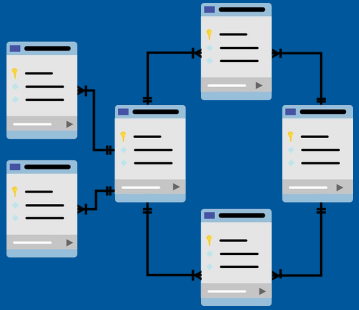
Store Data

Cleaning the Tables

Test Overall Performance

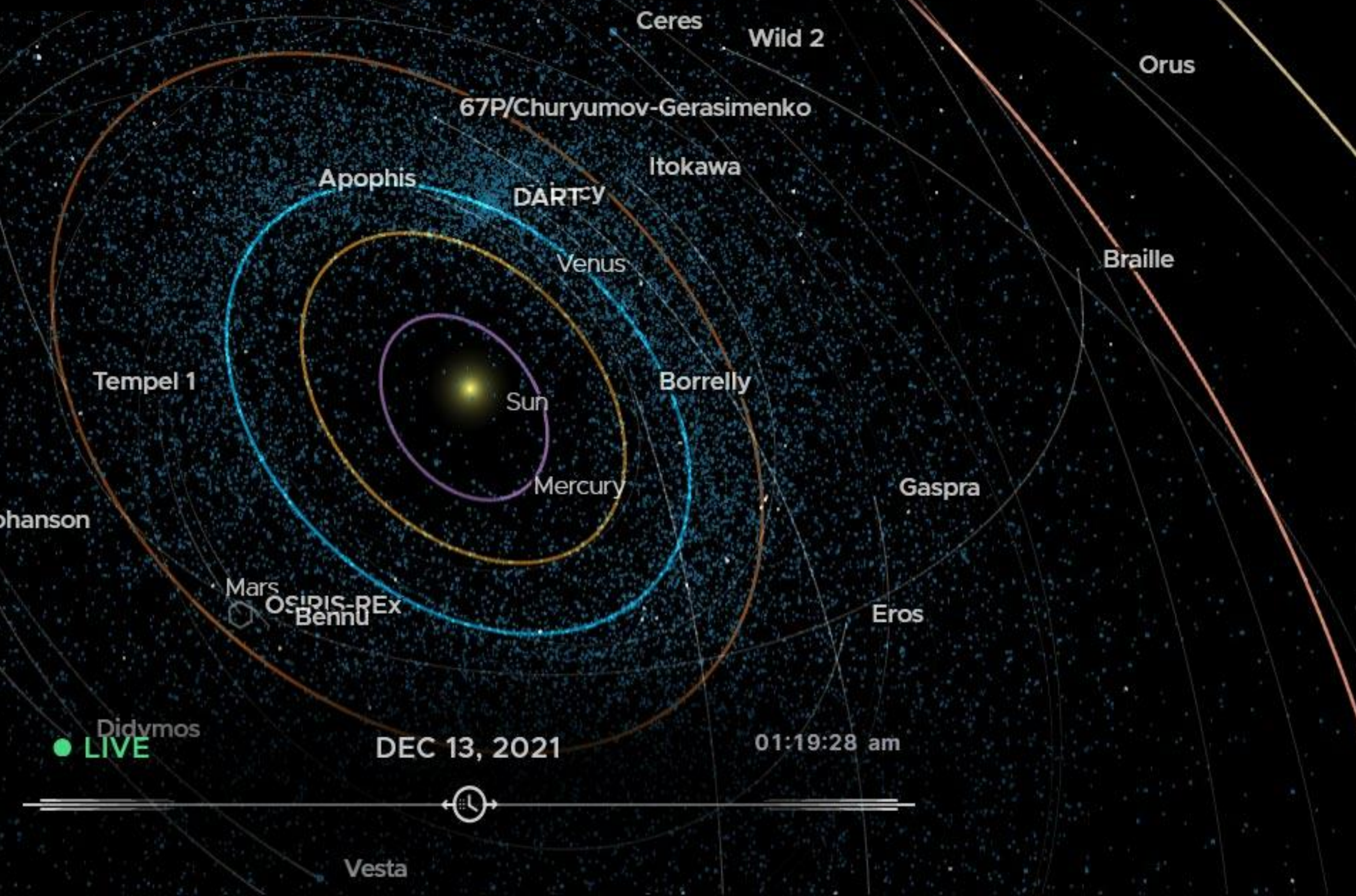
Store Large Data Safely

For Range of Libraries



Model Stored Data

Our Data Source



Our data comes from the Center For Near Earth Object Studies (CENOS) at NASA.

Data Exploration Phase

Database consisted of 29,052 rows and 36 columns.

We dropped string columns containing names, IDs, equinox and PC.

Eliminated columns with a null value more than 50% of the total number of rows and replaced the other null values with 0.

Analysis Phase

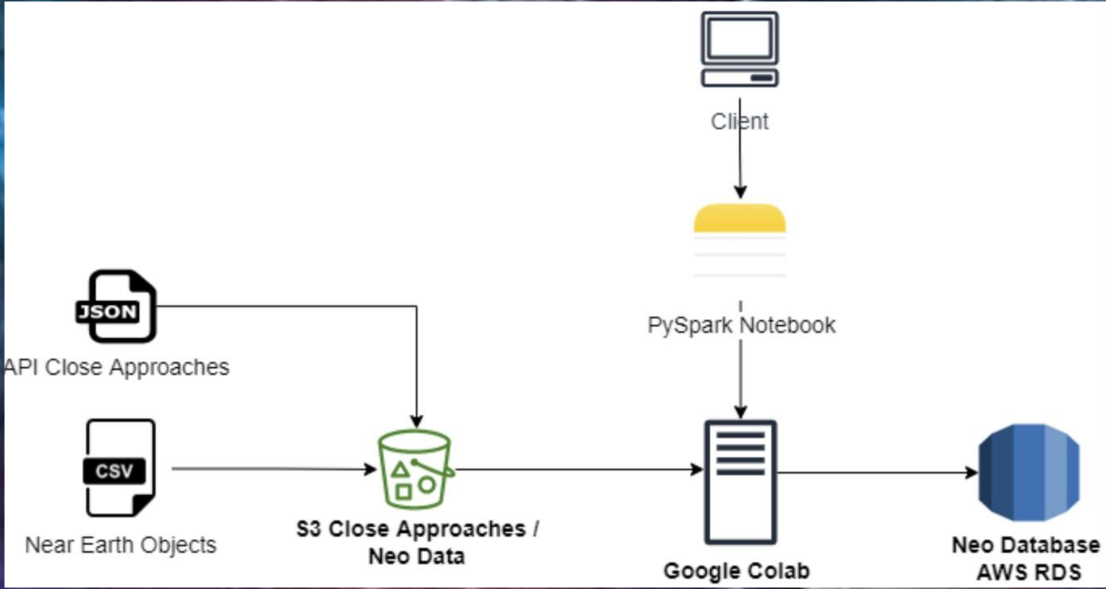
We have a 78% accuracy score

- Random Forest
- RandomOverSampler
- Over and Under Sampling for DS2
- SMOTE
- Neural Networks

Database

Since we had over 28,000 rows, we decided to use PostgreSQL because it stores large and sophisticated data safely and we could visually see the relationships between our data.

ETL Process: JSON Data



Machine Learning

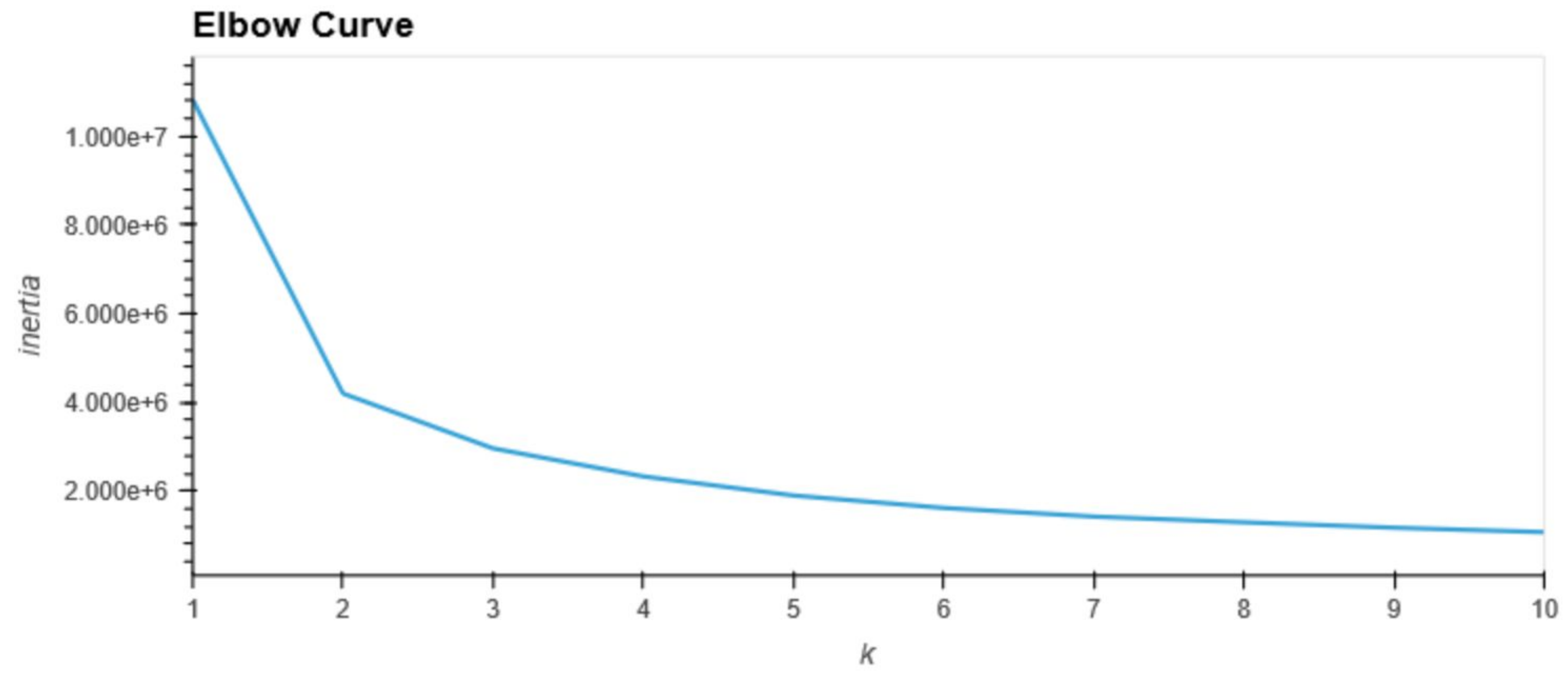
Performing Resampling

- Over/Under Sampling classified 40% of actual impacts wrong (228 predicted wrong, 325 predicted correctly.

- RandomOverSampler and SMOTE failed to have any accurate predictions.

- Neural Networks generated most accurate predictions with 99.8% accuracy.

Clustering



Elbow Curve
1.000e+7
8.000e+6
6.000e+6
4.000e+6
2.000e+6
inertia
1
2
3
4
5
6
7
8
9
10
k

Choosing Our Model

<u>Online research on impact analysis:</u> Incorporated 2 more features V_inf & V_rel

<u>Neural Networks</u>: **99.8% accuracy on test data, Highest accuracy score and lowest loss.**

<u>Using RandomForest</u> : Narrowed down our features variable to 3. False predictions on Hazardous asteroids was not satisfactory.

```
# sorting the features by their importance.
sorted(zip(rf_model.feature_importances_, X.columns), reverse=True)
```

```
[(0.19644519683387338, 'moid_ld'),
 (0.19087137867726245, 'moid'),
 (0.18111737382035517, 'h'),
 (0.03258768133302332, 'sigma_i'),
 (0.03195676846955056, 'sigma_ma'),
 (0.02706870349954681, 'sigma_e'),
 (0.022797875135387183, 'sigma_n'),
```

Confusion Matrix

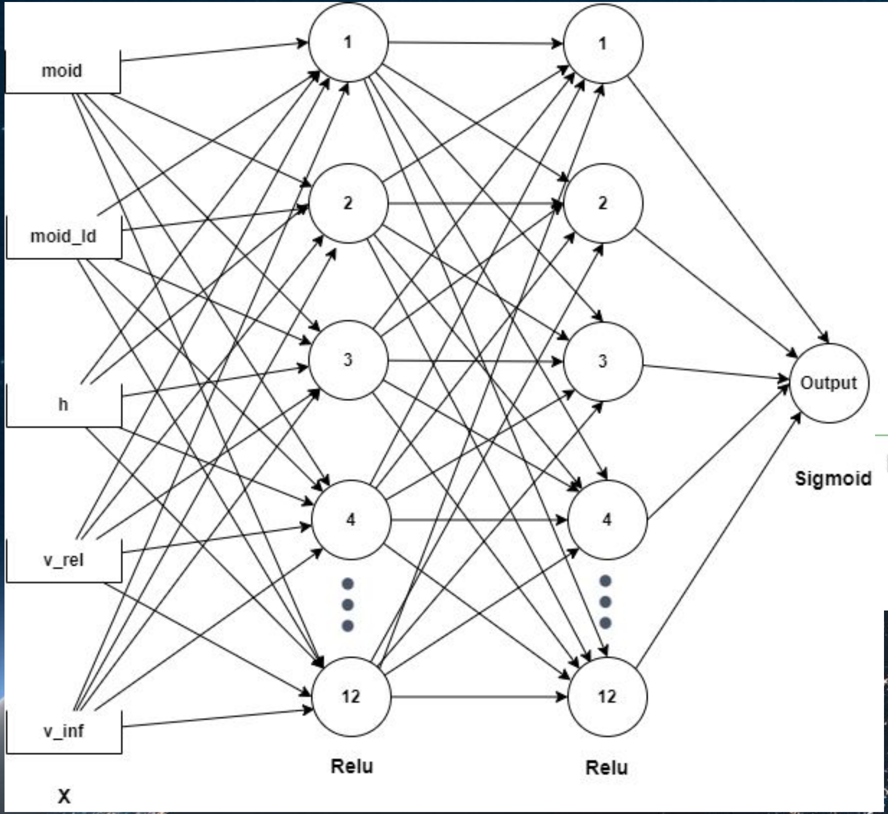
	Predicted 0	Predicted 1
Actual 0	5090	9
Actual 1	7	518

```
Accuracy Score : 0.9971550497866287
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5099
           1       0.98      0.99      0.98       525

    accuracy                           1.00      5624
   macro avg       0.99      0.99      0.99      5624
weighted avg       1.00      1.00      1.00      5624
```

Neural Networks Model




```
# Check the structure of the model
nn.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 12)                72

 dense_1 (Dense)             (None, 12)                156

 dense_2 (Dense)             (None, 1)                 13

=================================================================
Total params: 241
Trainable params: 241
Non-trainable params: 0
_________________________________________________________________
```

```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
176/176 - 1s - loss: 0.0055 - accuracy: 0.9979 - 1s/epoch - 8ms/step
Loss: 0.005502276588231325, Accuracy: 0.9978662729263306
```

Tableau Story - With Interactive Elements

-

Neural Networks Predictions Results

Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	8440	1082
Actual 1	1442	406

Accuracy Score : 0.7780123131046613

Classification Report

	precision	recall	f1-score	support
0	0.85	0.89	0.87	9522
1	0.27	0.22	0.24	1848
accuracy			0.78	11370
macro avg	0.56	0.55	0.56	11370
weighted avg	0.76	0.78	0.77	11370


QUESTIONS?