NASA SPACE APP CHALLENGE

DETECTION OF EXOPLANETS

- ESSIL AMAIMI
- FERHAT AIT GOUGAM
- AARON BARRALES
- YOANE NTSIKA MAKOUANGOU KENEITH

INTRODUCTION

This project is developed as part of the NASA Space App Challenge. It aims to create an AI/ML model capable of identifying exoplanets from transit data (Kepler, K2, TESS). A web interface allows users to upload data, receive predictions, and visualize the results.

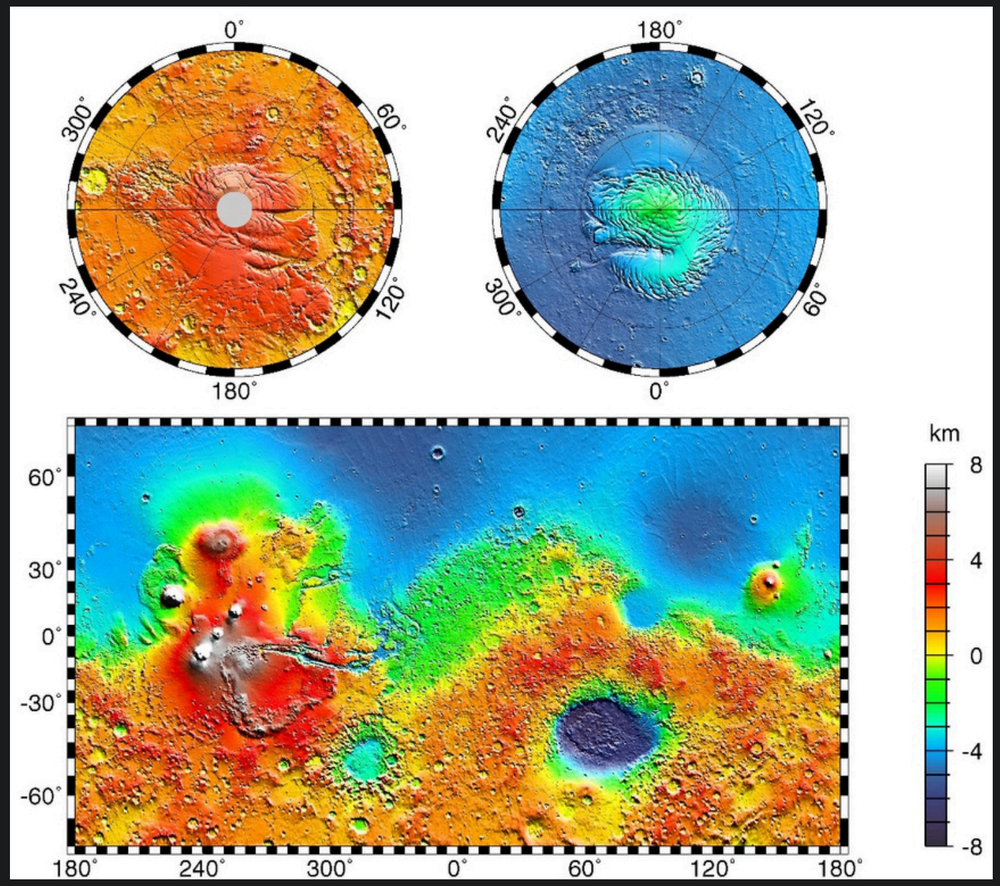
OBJECTIFS

1. CREATE AN AI/ML MODEL TO IDENTIFY EXOPLANETS FROM NASA DATA (KEPLER, K2, TESS)
2. DEVELOP A WEB INTERFACE FOR USER INTERACTION.
3. OPTIMIZE VARIABLE PROCESSING TO IMPROVE MODEL ACCURACY.
4. USE SUITABLE OPEN-SOURCE TOOLS FOR THE PROJECT.

VISUAL OF THE INTERFACE

HERE ARE THE PURPOSES THE GOALS OF 3D SURFACE VISUALIZATION

3D SURFACE VISUALIZATION RECONSTRUCTING TERRAIN RELIEF (ELEVATION, TOPOGRAPHY) TO BETTER UNDERSTAND THE LANDFORMS, CRATERS, VALLEYS



TESTING MODELS

Visual of the code

THE CODE ALLOWS US TO USE THE CSV SAMPLES OF THE DIFFERENT SATELLITES TO CALCULATE THE EFFICIENCY OF THE MODEL ACCORDING TO THE COLLECTED DATA

IN THIS CASE, IT WAS TESTED WITH TWO MODELS WITH THE HIGHEST PERFORMANCES:

- BOOST
- RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score, StratifiedKFold
import numpy as np

# Define the model
rf = RandomForestClassifier(random_state=42)

# Define 5-fold stratified cross-validation
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Perform cross-validation on the *resampled* training set
scores = cross_val_score(rf, X_train_res, y_train_res, cv=cv, scoring='accura

print("Cross-validation scores:", scores)
print("Mean accuracy:", np.mean(scores))
```

```
Cross-validation scores: [0.9728799  0.96943607 0.961257   0.97243755 0.97631
Mean accuracy: 0.9704648085300608
```

```
# Fit the model on the oversampled training set
rf.fit(X_train_res, y_train_res)

# Predict on the untouched test set
y_pred = rf.predict(X_test)

# Evaluate performance
from sklearn.metrics import accuracy_score, classification_report, confusion_

test_accuracy = accuracy_score(y_test, y_pred)
print("Test set accuracy:", test_accuracy)

print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

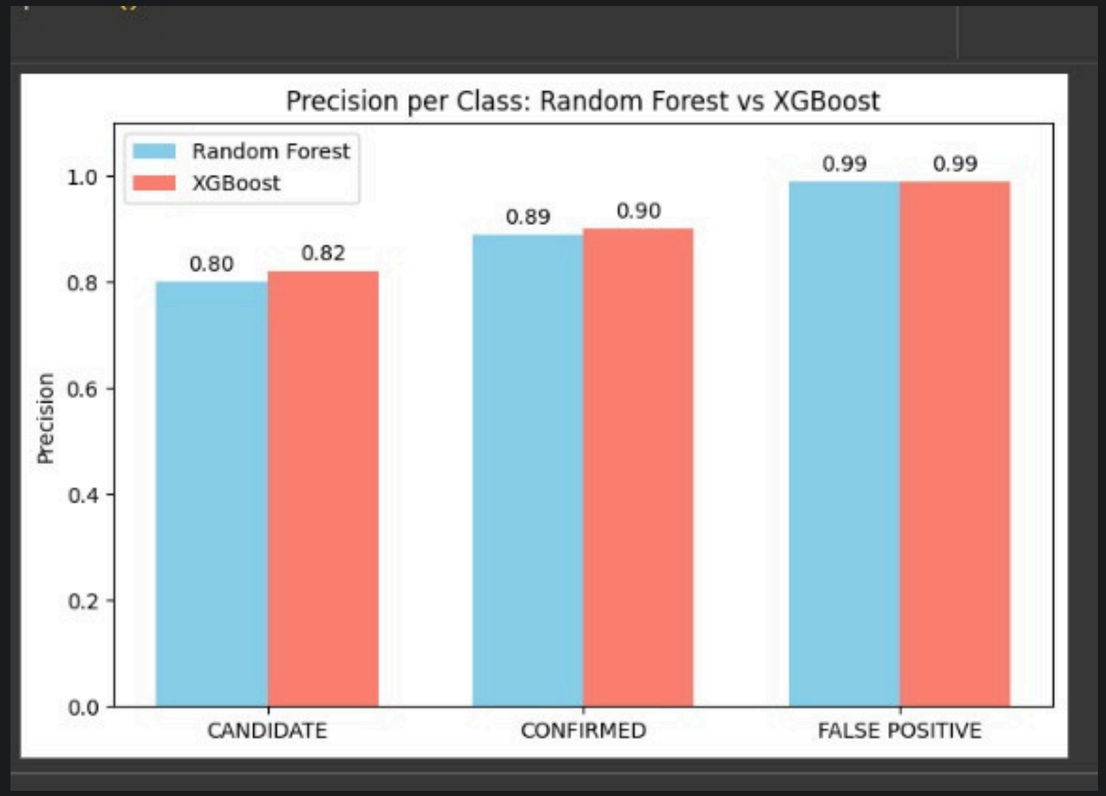
TESTING MODELS

XGBoost:

- PERCENTAGE OF 82 % OF SUSPECTING OF EXOPLANETS
- PERCENTAGE OF 9O % OF SUSPECTS IN THE CONFIRMATION OF THE PLANET
- ALMOS A SERVITUDE WITH 99% OF AFFIRMATION THE AIT4S NOT A PLANET

Random Forest:

- PERCENTAGE OF 8O % OF SUSPECTING OF EXOPLANETS
- PERCENTAGE OF 89 % OF SUSPECTS IN THE CONFIRMATION OF THE PLANET
- ALMOS A SERVITUDE WITH 99% OF AFFIRMATION THE AIT4S NOT A PLANET



UPGRADING POINTS

1. LINK THE BACKENDED FRONTENDED AND MODEL
2. UPGRADING THE DATA PRESENTED