```python
# ==============================================================================
# AQUA-ANALYTICS: PREDICTIVE MODEL SCRIPT
# This script will connect to Google Sheet, train a model, find the key
# drivers, and save the results with predictions.
# ==============================================================================

# Step A: Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from google.colab import auth
import gspread
from google.auth import default

# Step B: Authenticate and connect to Google Sheet
print("--> Step 1: Connecting to Google Sheets...")
try:
    auth.authenticate_user()
    creds, _ = default()
    gc = gspread.authorize(creds)
    print("Authentication successful!")
except Exception as e:
    print(f"Authentication failed. Please try running this cell again. Error:
    {e}")

# --- !!! IMPORTANT CONFIGURATION !!! ---
# THE FULL URL of Google Sheet.
SPREADSHEET_URL = "https://docs.google.com/spreadsheets/d/
1rQp2pRQBmPPhenSPoOgE7njHZbm8XUr6EZ7BHnGfTYE/edit?gid=460422318#gid=460422318"
# The name of the worksheet that contains my final, clean data.
WORKSHEET_NAME = "Master Sheet"
# The name of the column I want to predict.
TARGET_COLUMN = "Certification_Deficit_Pct"
# --- END OF CONFIGURATION ---

try:
    # To open the spreadsheet and the specific worksheet
    worksheet = gc.open_by_url(SPREADSHEET_URL).worksheet(WORKSHEET_NAME)

    # To get all the data from the sheet and convert it into a pandas DataFrame
    data = worksheet.get_all_records()
    df = pd.DataFrame(data)
    print(f"Successfully loaded {len(df)} rows from '{WORKSHEET_NAME}'.")
    print("\nData Preview:")
    print(df.head())
except Exception as e:
    print(f"Failed to load data. Please check your URL and Worksheet Name.
    Error: {e}")


# Step C: To prepare the data for the Machine Learning model
print("\n--> Step 2: Preparing data for the model...")
# To define the features (inputs) and the target (what I want to predict)
features = df.drop(columns=['State/UT', TARGET_COLUMN]) # Using all columns
except the state name and the target itself
target = df[TARGET_COLUMN]

# Handle any non-numeric data by converting it to numbers (one-hot encoding)
features = pd.get_dummies(features)
print("Data preparation complete.")


# Step D: Training the Predictive Model
print("\n--> Step 3: Training the predictive model...")
# I am using RandomForestRegressor, which is a powerful and reliable model for
this kind of task.
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(features, target)
print("Model training complete!")


# Step E: Analyzing the Key Drivers (Feature Importance)
print("\n--> Step 4: Analyzing the Key Drivers of Implementation Risk...")
# Getting the importance of each feature from the trained model
importances = model.feature_importances_
feature_names = features.columns

# Creating a DataFrame to display the results clearly
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Importance':
importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance',
ascending=False)

print("\n==========================================================")
print("           *** KEY DRIVERS ANALYSIS ***")
print("==========================================================")
print("This table shows which factors have the biggest impact on predicting the
Certification Deficit.")
print("A higher importance score means the factor is more influential.")
print(feature_importance_df)
print("==========================================================")


# Step F: Making Predictions and Save the Results
print("\n--> Step 5: Making predictions and preparing the final output...")
# Using the trained model to make predictions on the entire dataset
predictions = model.predict(features)

# Add the predictions as a new column to the original DataFrame
df['Predicted_Certification_Deficit'] = predictions

# Preparing the file for download
output_filename = "predictions_output.csv"
df.to_csv(output_filename, index=False)

print(f"\nSUCCESS! A file named '{output_filename}' is ready for download.")
print("This file contains all of the original data plus the new AI predictions.
")
feature_importance_df.to_csv("feature_importance_output.csv", index=False)
print(f"\nSUCCESS! A file named {"feature_importance_output.csv"} is ready for
download.")
```

```
--> Step 1: Connecting to Google Sheets...
Authentication successful!
Successfully loaded 34 rows from 'Master Sheet'.

Data Preview:
                    State/UT  Overall_FHTC_Pct  Village_Inequality_Index  \
0  Andaman and Nicobar Islands              1.00                      0.00
1               Andhra Pradesh              0.74                      0.11
2            Arunachal Pradesh              1.00                      0.00
3                        Assam              0.81                      0.15
4                        Bihar              0.96                      0.03

   PWS_Infrastructure_Gap  Literacy_Rate_2011  Certification_Deficit_Pct
0                    0.00                85.0                       0.00
1                    0.00                60.0                       0.23
2                    0.00                60.0                       0.00
3                    0.02                69.0                       0.44
4                    0.02                60.0                       1.00

--> Step 2: Preparing data for the model...
Data preparation complete.

--> Step 3: Training the predictive model...
Model training complete!

--> Step 4: Analyzing the Key Drivers of Implementation Risk...

==========================================================
           *** KEY DRIVERS ANALYSIS ***
==========================================================
This table shows which factors have the biggest impact on predicting the Certification Deficit.
A higher importance score means the factor is more influential.
                    Feature  Importance
3        Literacy_Rate_2011    0.419329
0          Overall_FHTC_Pct    0.216872
1  Village_Inequality_Index    0.188521
2    PWS_Infrastructure_Gap    0.175278
==========================================================

--> Step 5: Making predictions and preparing the final output...

SUCCESS! A file named 'predictions_output.csv' is ready for download.
This file contains all of the original data plus the new AI predictions.

SUCCESS! A file named feature_importance_output.csv is ready for download.
```


30/08/2025, 18:36 Aqua-Analytics Model.ipynb - Colab
https://colab.research.google.com/drive/1JvfKQu8ysC1QVOgua_ZeeQb5VWTIsDCg#scrollTo=nDffqUGIynZA&uniqifier=2&printMode=true