Compararea modelelor de predictie aplicate pe un singur set de date

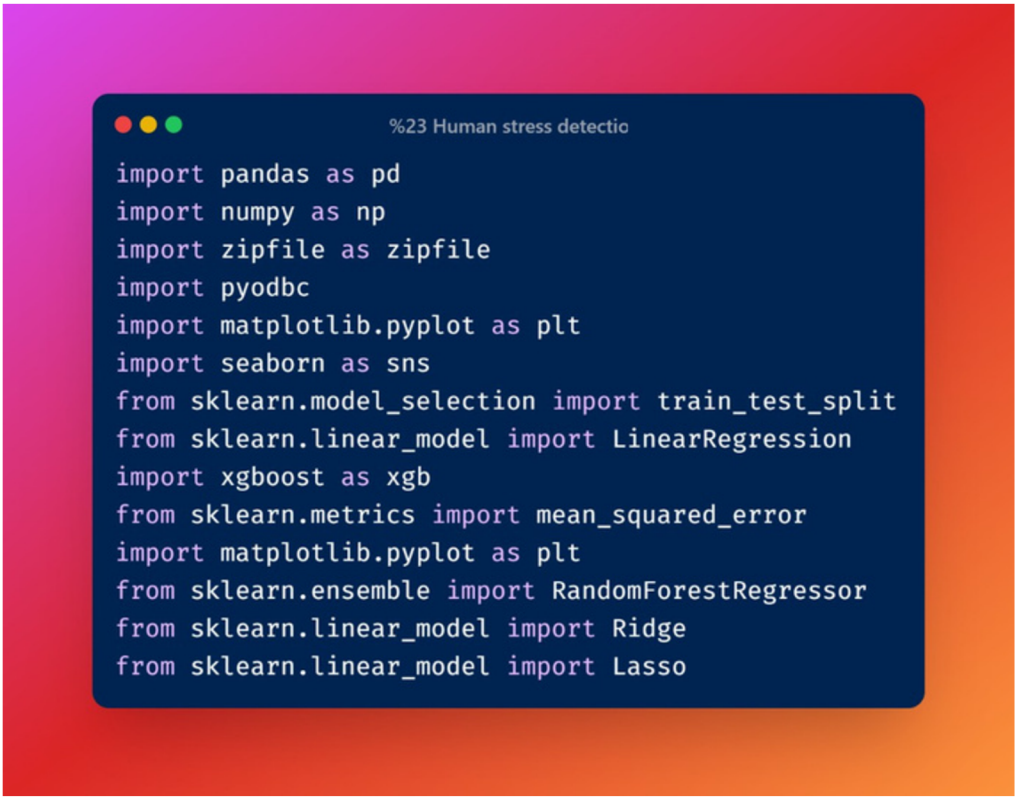
În acest proiect am vrut să arât diferența dintre acuratețea a cinci modele de predicție diferite, acestea fiind :

- **XGBoost**
- **RandomForest**
- **Regresia Ridge**
- **Regresia Lasso**
- **Regresia Lineară**

Proiectul este scris aproape în întregime între-un script de python și urmărește următoarea structură :

1.Procesarea Inițială a Datelor

- Importarea bibliotecilor necesare;

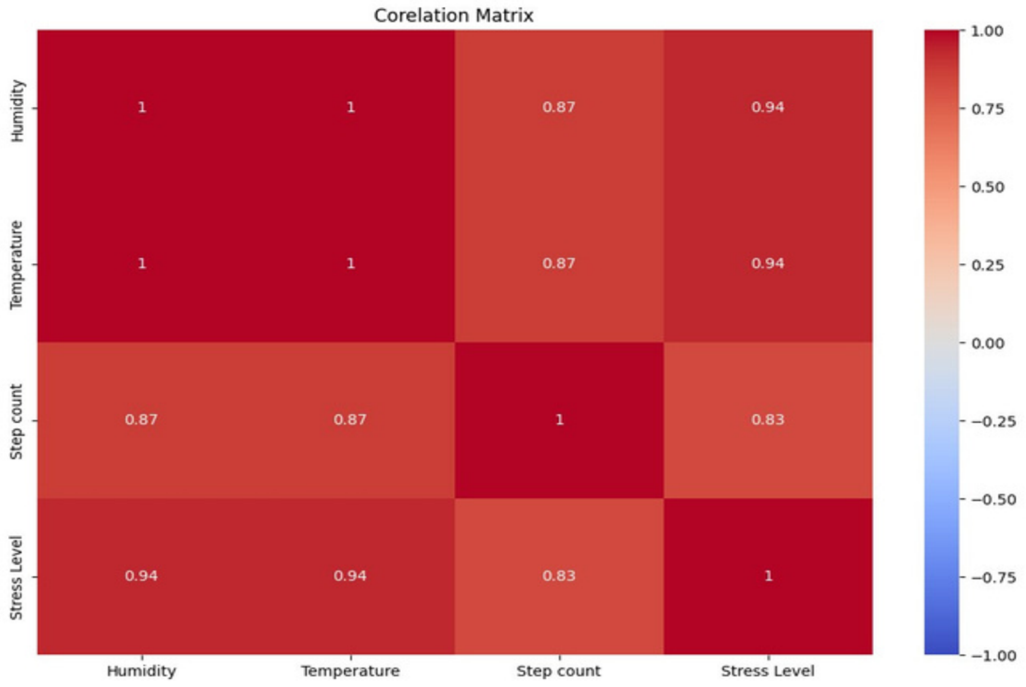


- Extracția datelor dintr-un fișier zip
- Transformarea valorilor temperaturii din Fahrenheit în Celsius
- Verificarea datelor pentru a asigura că nu există valori NA și tipurile de date sunt corecte
- Crearea de grafice pentru a vizualiza distribuția datelor și anomaliile
- Calcularea coeficienților de corelație Pearson și vizualizarea lor folosind un heatmap

Funcția pentru vizualizarea coeficientului de corelație si coeficientul mai jos:

```
%23 Human stress detectio
def pearson_correlation(dataset,x,y):
    X = dataset[x]
    Y = dataset[y]
    correlation_matrix = np.corrcoef(X, Y)
    correlation_coefficient = correlation_matrix[0, 1] #0.8326232433614356
    return correlation_coefficient

pearson_correlation(file,"Humidity","Stress Level") #0.9360362899114489
pearson_correlation(file,"Temperature","Stress Level") #0.9360362899114495
pearson_correlation(file,"Step count","Stress Level") #0.8326232433614356
pearson_correlation(file,"Step count","Humidity") #0.8704860484528991
pearson_correlation(file,"Temperature","Humidity") #1.0
pearson_correlation(file,"Step count","Temperature") #0.8704860484528993
```

Vizualizarea Heatmap pentru corelație:

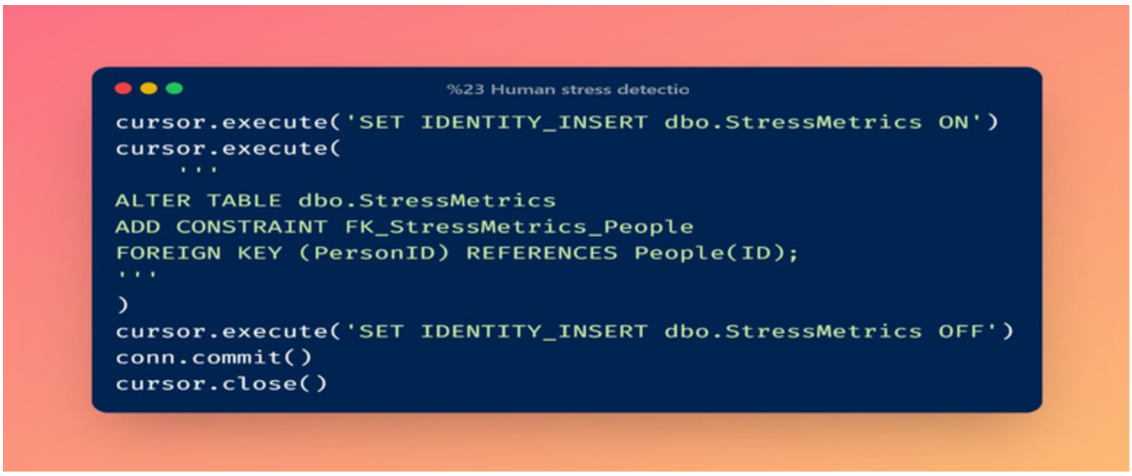


- Căutarea anomaliilor și eliminarea acestora

2.Crearea și Inserarea Datelor în Baza de Date:

- **Conectarea la o bază de date locală**

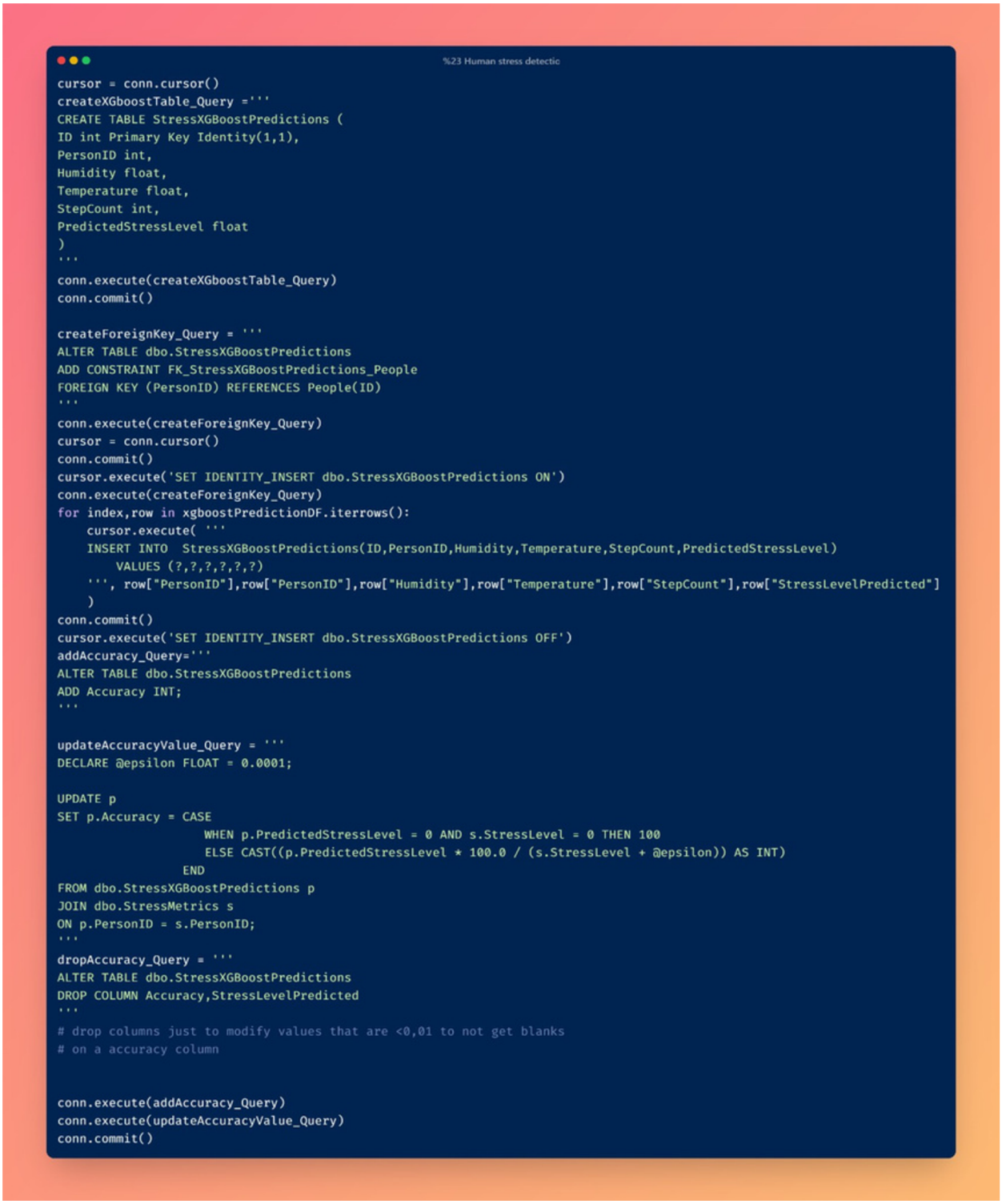
- **Crearea tabelelor StressMetrics și People**
- **Inserarea valorilor din fișierele CSV și Excel în tabelele respective**
- **Stabilirea unei relații de cheie străină între StressMetrics și People**



3.Modele Predictive:

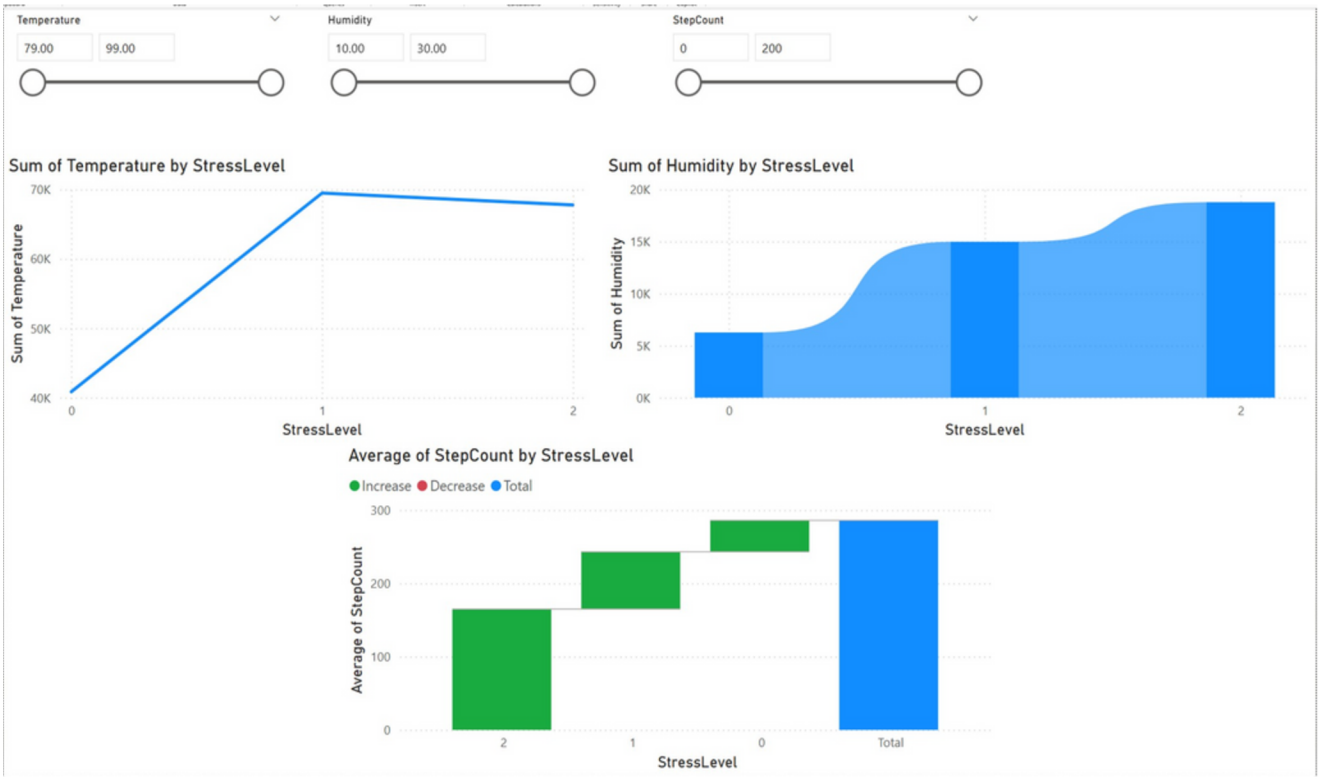
- **Pregatirea seturilor de date pentru antrenarea modelelor și alegem variabilele independente cu ajutorul cărora facem predicția ;**

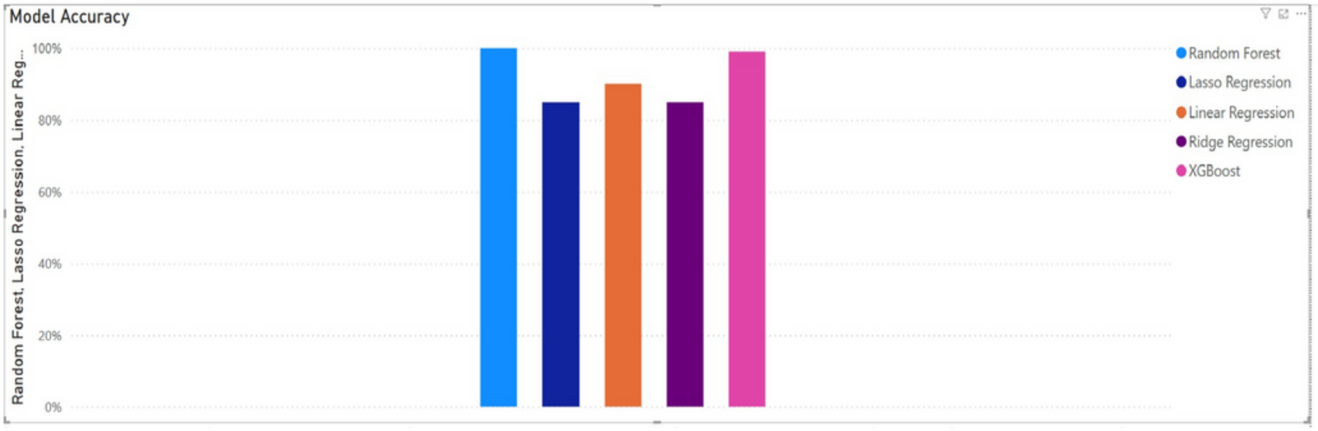
- **Calculăm eroarea medie patratică și coeficientul de determinare pentru a observa eficacitatea modelului înainte de a-l aplica pe setul de date;**
- **Vom crea un tabel nou pentru fiecare dintre predicțiile noastre la care vom adăuga un field nou de acuratețe prin care**
- **Calculam acuratețea modelului raportat la valorile din setul initial**



```
%23 Human stress detectic
cursor = conn.cursor()
createXGboostTable_Query ='''
CREATE TABLE StressXGBoostPredictions (
ID int Primary Key Identity(1,1),
PersonID int,
Humidity float,
Temperature float,
StepCount int,
PredictedStressLevel float
)
'''
conn.execute(createXGboostTable_Query)
conn.commit()

createForeignKey_Query = '''
ALTER TABLE dbo.StressXGBoostPredictions
ADD CONSTRAINT FK_StressXGBoostPredictions_People
FOREIGN KEY (PersonID) REFERENCES People(ID)
'''
conn.execute(createForeignKey_Query)
cursor = conn.cursor()
conn.commit()
cursor.execute('SET IDENTITY_INSERT dbo.StressXGBoostPredictions ON')
conn.execute(createForeignKey_Query)
for index,row in xgboostPredictionDF.iterrows():
    cursor.execute( '''
    INSERT INTO  StressXGBoostPredictions(ID,PersonID,Humidity,Temperature,StepCount,PredictedStressLevel)
        VALUES (?,?,?,?,?,?)
    ''', row["PersonID"],row["PersonID"],row["Humidity"],row["Temperature"],row["StepCount"],row["StressLevelPredicted"]
    )
conn.commit()
cursor.execute('SET IDENTITY_INSERT dbo.StressXGBoostPredictions OFF')
addAccuracy_Query='''
ALTER TABLE dbo.StressXGBoostPredictions
ADD Accuracy INT;
'''

updateAccuracyValue_Query = '''
DECLARE @epsilon FLOAT = 0.0001;

UPDATE p
SET p.Accuracy = CASE
                    WHEN p.PredictedStressLevel = 0 AND s.StressLevel = 0 THEN 100
                    ELSE CAST((p.PredictedStressLevel * 100.0 / (s.StressLevel + @epsilon)) AS INT)
                END
FROM dbo.StressXGBoostPredictions p
JOIN dbo.StressMetrics s
ON p.PersonID = s.PersonID;
'''
dropAccuracy_Query = '''
ALTER TABLE dbo.StressXGBoostPredictions
DROP COLUMN Accuracy,StressLevelPredicted
'''
# drop columns just to modify values that are <0,01 to not get blanks
# on a accuracy column


conn.execute(addAccuracy_Query)
conn.execute(updateAccuracyValue_Query)
conn.commit()
```

4.Crearea de vizualizări în PowerBI





Rezultate :
Random forest – 99.2%
XGBoost – 99.0%
Regresia Lasso – 85%
Regresia Liniara – 90%
Regresia Ridge – 86%

Diagramă Entitate-Relație