3rd phase project

Glass prediction

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
glass_data = pd.read_csv('https://github.com/FlipRoboTechnologies/ML-Datasets/raw/main/Glass%20Identification/Glass%20Identification.csv')

# Separate features and target
X = glass_data.drop('Type of glass', axis=1)
y = glass_data['Type of glass']

# Encode target variable
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and train the model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
```

```
print('Accuracy:', accuracy_score(y_test, y_pred))

print('Classification Report:\n', classification_report(y_test, y_pred,
target_names=label_encoder.classes_))
```

Student grade prediction

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset

grades_data = pd.read_csv('https://github.com/FlipRoboTechnologies/ML-
Datasets/raw/main/Grades/Grades.csv')


# Separate features and target

X = grades_data.drop(['Seat No', 'CGPA'], axis=1)

y = grades_data['CGPA']


# Split data into train and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Create and train the model

model = LinearRegression()

model.fit(X_train, y_train)


# Make predictions on the test set

y_pred = model.predict(X_test)


# Evaluate the model

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse:.2f}')

print(f'R-squared: {r2:.2f}')
```