Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei

Lucrare de laborator

Disciplina: *Inteligență Artificială*

Tema: *Linear Regression*

A efectuat: st. gr. SI-221M Postu Ivan

A verificat: Gavrilița Mihail

Chișinău – 2023

Task 1 Import your data. Analyze it via common statistical approaches. Cleanse the data if necessary.

```
import pandas as pd
import statistics

# Load data into pandas DataFrame
data = pd.read_csv(
    "/home/ivan/Desktop/ArtificialIntelligence/Lab3/apartmentComplexData.txt",
    header=None,
    usecols=[2, 3, 4, 5, 6, 8],
    names=[
        "complexAge",
        "totalRooms",
        "totalBedrooms",
        "complexInhabitants",
        "apartmentsNr",
        "medianComplexValue",
    ],
)
print("Informatii privind setul de date")
print(data.info())

print("Media virstei complexelor:")
print(statistics.mean(data.complexAge))
print("Media numarului total de camere:")
print(statistics.mean(data.totalRooms))
print("Media numarului total de camere de baie:")
print(statistics.mean(data.totalBedrooms))
print("Media numarului de locuitori:")
print(statistics.mean(data.complexInhabitants))
print("Media numarului de apartamente in bloc:")
print(statistics.mean(data.apartmentsNr))
print("Meida valoare complexă:")
print(statistics.mean(data.medianComplexValue))
```

Output:

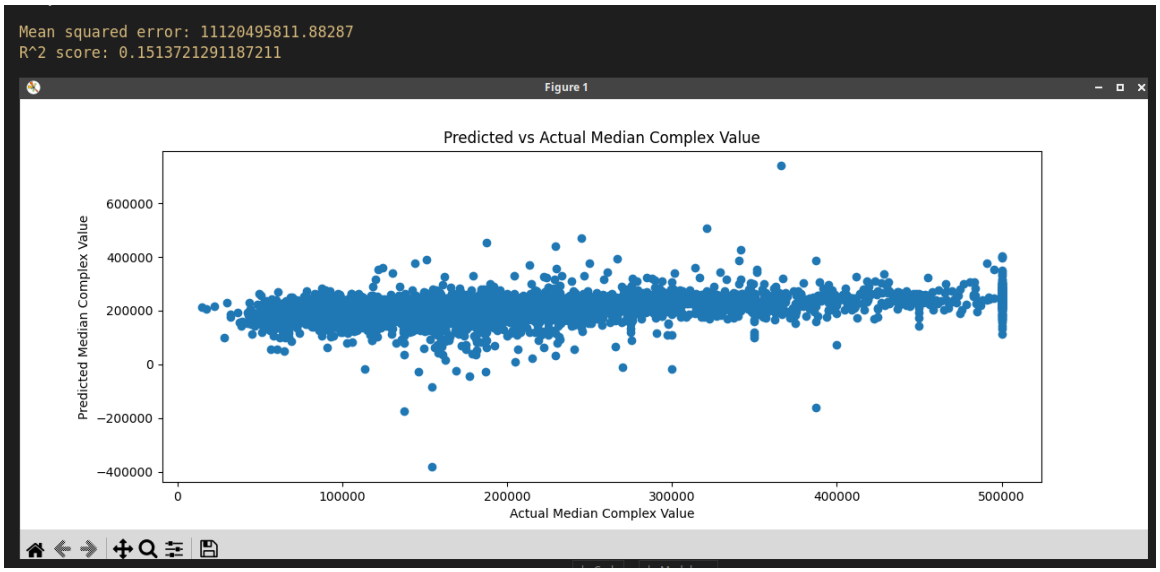
```
Informatii privind setul de date
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   complexAge          20640 non-null  float64
 1   totalRooms          20640 non-null  float64
 2   totalBedrooms       20640 non-null  float64
 3   complexInhabitants  20640 non-null  float64
 4   apartmentsNr        20640 non-null  float64
```

```
 5   medianComplexValue  20640 non-null  float64
dtypes: float64(6)
memory usage: 967.6 KB
None
Media virstei complexelor:
28.639486434108527
Media numarului total de camere:
2635.7630813953488
Media numarului total de camere de baie:
537.8980135658915
Media numarului de locuitori:
1425.4767441860465
Media numarului de apartamente in bloc:
499.5396802325581
Meida valoare complexă:
206855.81690891474
```

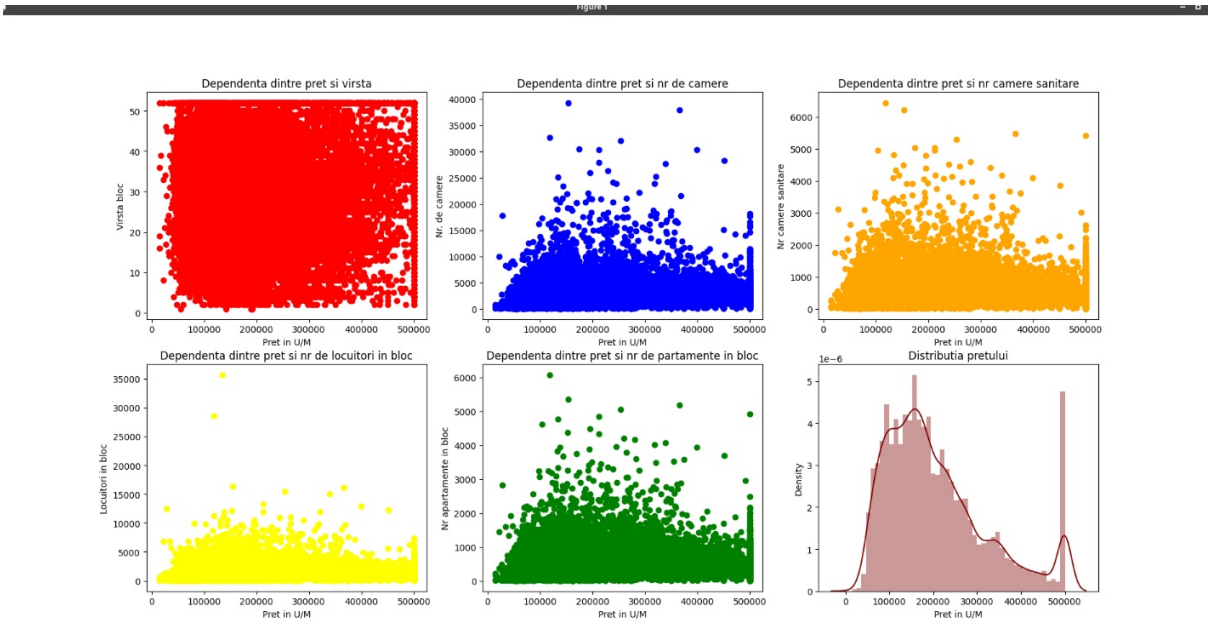
Task 2 - Train your model by applying linear regression.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Load data into pandas DataFrame
data = pd.read_csv(
    "/home/ivan/Desktop/ArtificialIntelligence/Lab3/apartmentComplexData.txt",
    header=None,
    usecols=[2, 3, 4, 5, 6, 8],
    names=[
        "complexAge",
        "totalRooms",
        "totalBedrooms",
        "complexInhabitants",
        "apartmentsNr",
        "medianComplexValue",
    ],
)

X = data.drop("medianComplexValue", axis=1)
Y = data['medianComplexValue']
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
reg = LinearRegression()
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)

print("Mean squared error:", mean_squared_error(y_test, y_pred))
```

```
print("R^2 score:", r2_score(y_test, y_pred))

plt.scatter(y_test, y_pred)
plt.xlabel("Actual Median Complex Value")
plt.ylabel("Predicted Median Complex Value")
plt.title("Predicted vs Actual Median Complex Value")
plt.show()
```



Task 3 - Show the prediction power of your model by attempting to predict the price of a new house.




```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

```
import seaborn as sb  # visualization

# Load data into pandas DataFrame
dataSet = pd.read_csv(
    "/home/ivan/Desktop/ArtificialIntelligence/Lab3/apartmentComplexData.txt",
    header=None,
    usecols=[2, 3, 4, 5, 6, 8],
    names=[
        "complexAge",
        "totalRooms",
        "totalBedrooms",
        "complexInhabitants",
        "apartmentsNr",
        "medianComplexValue",
    ],
)

plt.subplot(2, 3, 1)
plt.scatter(dataSet["medianComplexValue"], dataSet["complexAge"], color="red")
plt.title("Dependenta dintre pret si virsta")
plt.ylabel("Virsta bloc")
plt.xlabel("Pret in U/M")
plt.subplot(2, 3, 2)
plt.scatter(dataSet["medianComplexValue"], dataSet["totalRooms"], color="blue")
plt.title("Dependenta dintre pret si nr de camere")
plt.ylabel("Nr. de camere")
plt.xlabel("Pret in U/M")
plt.subplot(2, 3, 3)
plt.scatter(dataSet["medianComplexValue"], dataSet["totalBedrooms"], color="orange")
plt.title("Dependenta dintre pret si nr camere sanitare")
plt.ylabel("Nr camere sanitare")
plt.xlabel("Pret in U/M")
plt.subplot(2, 3, 4)
plt.scatter(
    dataSet["medianComplexValue"], dataSet["complexInhabitants"], color="yellow"
)
plt.title("Dependenta dintre pret si nr de locuitori in bloc")
plt.ylabel("Locuitori in bloc")
plt.xlabel("Pret in U/M")
plt.subplot(2, 3, 5)
plt.scatter(dataSet["medianComplexValue"], dataSet["apartmentsNr"], color="green")
plt.title("Dependenta dintre pret si nr de partamente in bloc")
plt.ylabel("Nr apartamente in bloc")
plt.xlabel("Pret in U/M")
plt.subplot(2, 3, 6)
sb.distplot(dataSet["medianComplexValue"], color="maroon")
plt.title("Distributia pretului")
plt.xlabel("Pret in U/M")
```

```
plt.subplot(2, 3, 6)
plt.show()
```

Task 4 - Re-train your model. Use Ridge, Lasso or Elastic Net regularization.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score as r2
from sklearn.metrics import explained_variance_score as evs

# Load data into pandas DataFrame
data = pd.read_csv(
    "/home/ivan/Desktop/ArtificialIntelligence/Lab3/apartmentComplexData.txt",
    header=None,
    usecols=[2, 3, 4, 5, 6, 8],
    names=[
        "complexAge",
        "totalRooms",
        "totalBedrooms",
        "complexInhabitants",
        "apartmentsNr",
        "medianComplexValue",
    ],
)

X = data.drop("medianComplexValue", axis=1)
Y = data["medianComplexValue"]
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5, random_state=0)

ols = LinearRegression()
ols.fit(X_train, y_train)
ols_yhat = ols.predict(X_test)
print("Scorul calculat cu OLS: {}".format(evs(y_test, ols_yhat)))
print("R-Squared: {}".format(r2(y_test, ols_yhat)))

ridge = Ridge(alpha=0.5)
ridge.fit(X_train, y_train)
ridge_yhat = ridge.predict(X_test)
print("Scorul calculat cu Ridge: {}".format(evs(y_test, ridge_yhat)))
print("R-Squared: {}".format(r2(y_test, ridge_yhat)))

lasso = Lasso(alpha=0.01)
lasso.fit(X_train, y_train)
lasso_yhat = lasso.predict(X_test)
print("Scorul calculat cu Lasso: {}".format(evs(y_test, lasso_yhat)))
```

```
print("R-Squared: {}".format(r2(y_test, lasso_yhat)))

en = ElasticNet(alpha=0.01)
en.fit(X_train, y_train)
en_yhat = en.predict(X_test)

print("Scorul calculat cu Elastic Net: {}".format(evs(y_test, en_yhat)))
print("R-Squared: {}".format(r2(y_test, en_yhat)))
```

Output:

```
Scorul calculat cu OLS: 0.1630341005537369
R-Squared: 0.1630314176789679
Scorul calculat cu Ridge: 0.1630341011890245
R-Squared: 0.1630314183128354
Scorul calculat cu Lasso: 0.1630341005282263
R-Squared: 0.1630314176533212
Scorul calculat cu Elastic Net: 0.16303416606450027
R-Squared: 0.16303148304312431
```

Task 5 - Score and compare the scores of the models you have implemented. Interpret the result

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score as r2
from sklearn.metrics import explained_variance_score as evs

# Load data into pandas DataFrame
data = pd.read_csv(
    "/home/ivan/Desktop/ArtificialIntelligence/Lab3/apartmentComplexData.txt",
    header=None,
    usecols=[2, 3, 4, 5, 6, 8],
    names=[
        "complexAge",
        "totalRooms",
        "totalBedrooms",
        "complexInhabitants",
        "apartmentsNr",
        "medianComplexValue",
    ],
)

X = data.drop("medianComplexValue", axis=1)
Y = data["medianComplexValue"]
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5, random_state=0)
```

```
ols = LinearRegression()
ols.fit(X_train, y_train)
ols_yhat = ols.predict(X_test)

ridge = Ridge(alpha=0.5)
ridge.fit(X_train, y_train)
ridge_yhat = ridge.predict(X_test)

lasso = Lasso(alpha=0.01)
lasso.fit(X_train, y_train)
lasso_yhat = lasso.predict(X_test)

en = ElasticNet(alpha=0.01)
en.fit(X_train, y_train)
en_yhat = en.predict(X_test)

print("Scorul OLS: {}".format(evs(y_test, ols_yhat)))
print("Scorul Ridge: {}".format(evs(y_test, ridge_yhat)))
print("Scorul Lasso: {}".format(evs(y_test, lasso_yhat)))
print("Variatia ElasticNet: {}".format(evs(y_test, en_yhat)))
print("R-pătrat OLS: {} ".format(r2(y_test, ols_yhat)))
print("R-pătrat Ridge: {} ".format(r2(y_test, ridge_yhat)))
print("R-pătrat Lasso: {} ".format(r2(y_test, lasso_yhat)))
print("R-pătrat ElasticNet: {} ".format(r2(y_test, en_yhat)))
```

Output:

```
Scorul OLS: 0.1630341005537369
Scorul Ridge: 0.1630341011890245
Scorul Lasso: 0.1630341005282263
Variatia ElasticNet: 0.16303416606450027
R-pătrat OLS: 0.1630314176789679
R-pătrat Ridge: 0.1630314183128354
R-pătrat Lasso: 0.1630314176533212
R-pătrat ElasticNet: 0.16303148304312431
```

Concluzie:

În urma realizării lucrării de laborator a fost atins obiectivul de a analiza și vizualiza datele. A fost cu succes antrenat un model ce permite de a prezice ce preț va avea apartamentele. Cu succes au fost aplicate librariile Scikit-learn, Pandas, Numpy și Matplotlib.