balaji-creator / BLENDED_LEARNING_EXP5

Code Pull requests Actions Projects Wiki Security Insights Settings

Watch 0

BSD-3-Clause license

0 stars 153 forks 0 watching Branches Activity Tags

Public repository · Forked from AkilaMohan/BLENDED_LEARNING_Implementation-of-Ridge-Lasso-and-ElasticNet-Regularization

1 Branch 0 Tags Go to file t Go to file Add file Code

This branch is 1 commit ahead of AkilaMohan/BLENDED_LEARNING_Implementation-of-Ridge-Lasso-and-ElasticNet-Regularization:main .

Contribute Sync fork

balaji-creator success 6e2cdca · 8 minutes ago

LICENSE	Initial commit	2 years ago
README.md	success	8 minutes ago
Screenshot 2026-02-25 084...	success	8 minutes ago
Screenshot 2026-02-25 084...	success	8 minutes ago

README License

BLENDED_LEARNING

Implementation of Ridge, Lasso, and ElasticNet Regularization for Predicting Car Price

AIM:

To implement Ridge, Lasso, and ElasticNet regularization models using polynomial features and pipelines to predict car price.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Jupyter notebook

Algorithm

1. Data Collection & Preprocessing Collect car price data with features like mileage, brand, year, and engine size. Clean the dataset by handling missing values, encoding categorical variables, and scaling numerical features.
2. Train-Test Split Divide the dataset into training and testing subsets, typically using an 80/20 ratio. This ensures the model is trained on one portion and evaluated on unseen data.
3. Model Initialization Set up three regression models: Ridge (L2), Lasso (L1), and ElasticNet (L1+L2). These models help reduce overfitting and improve prediction accuracy.
4. Hyperparameter Tuning Use cross-validation to find the best alpha (λ) values for each model. For ElasticNet, also tune the mixing parameter to balance L1 and L2 penalties
5. Model Training & Evaluation Train Ridge, Lasso, and ElasticNet models on the training set. Evaluate them using metrics like Mean Squared Error (MSE), R^2, and Mean Absolute Error (MAE).
6. Comparison & Selection Compare performance across the three models to identify the most effective one. Select the best model for deployment in predicting car prices reliably.

Program:

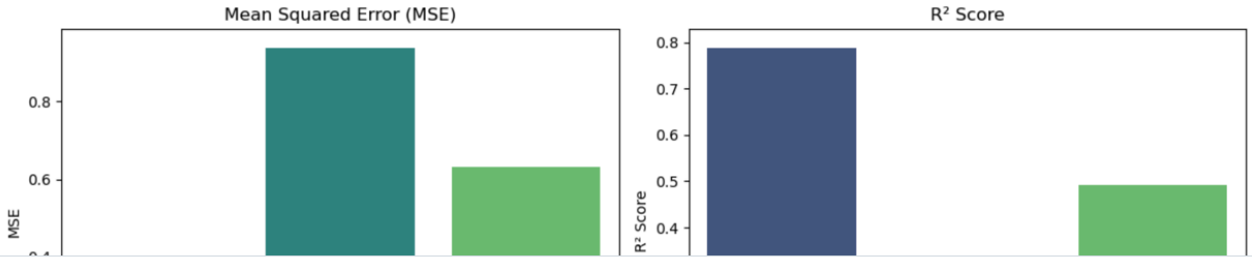
```
/*
Program to implement Ridge, Lasso, and ElasticNet regularization using pipelines.
Developed by: BALAJI B
RegisterNumber: 212225040040
*/
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge, Lasso, ElasticNet
from sklearn.preprocessing import PolynomialFeatures, StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error,r2_score

data =pd.read_csv("encoded_car_data")
data.head()

data=pd.get_dummies(data,drop_first=True)

x=data.drop('price',axis=1)
y=data['price']

scaler=StandardScaler()
x=scaler.fit_transform(x)
y=scaler.fit_transform(y.values.reshape(-1,1))

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

models = {
    "Ridge": Ridge(alpha=1.0),
    "Lasso": Lasso(alpha=1.0),
    "ElasticNet": ElasticNet(alpha=1.0, l1_ratio=0.5)
}
results = {}
```

```
for name, model in models.items():

    pipeline = Pipeline([
        ('poly', PolynomialFeatures(degree=2)),
        ('regressor', model)
    ])

    pipeline.fit(X_train, y_train)

    predictions = pipeline.predict(X_test)

    mse = mean_squared_error(y_test, predictions)
    r2 = r2_score(y_test, predictions)

    results[name] = {'MSE': mse, 'R2 Score': r2}
print('Name:Balaji B')
print('Reg. No:212225040040')
for model_name, metrics in results.items():
    print(f"{model_name} - Mean Squared Error: {metrics['MSE']:.2f}, R² Score:
{metrics['R2 Score']:.2f}")



results_df = pd.DataFrame(results).T
results_df.reset_index(inplace=True)
results_df.rename(columns={'index': 'Model'}, inplace=True)

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.barplot(x='Model', y='MSE', data=results_df, palette='viridis')
plt.title('Mean Squared Error (MSE)')
plt.ylabel('MSE')
plt.xticks(rotation=45)


plt.subplot(1, 2, 2)
sns.barplot(x='Model', y='R2 Score', data=results_df, palette='viridis')
plt.title('R² Score')
plt.ylabel('R² Score')
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```

Output:

```
Name:Balaji B
Reg. No:212225040040
Ridge - Mean Squared Error: 0.26, R² Score: 0.79
Lasso - Mean Squared Error: 0.94, R² Score: 0.25
ElasticNet - Mean Squared Error: 0.63, R² Score: 0.49
```



Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package