Predicting diabetes progression with Machine Learning

Federico Trotta

MACHINE LEARNING PROJECT

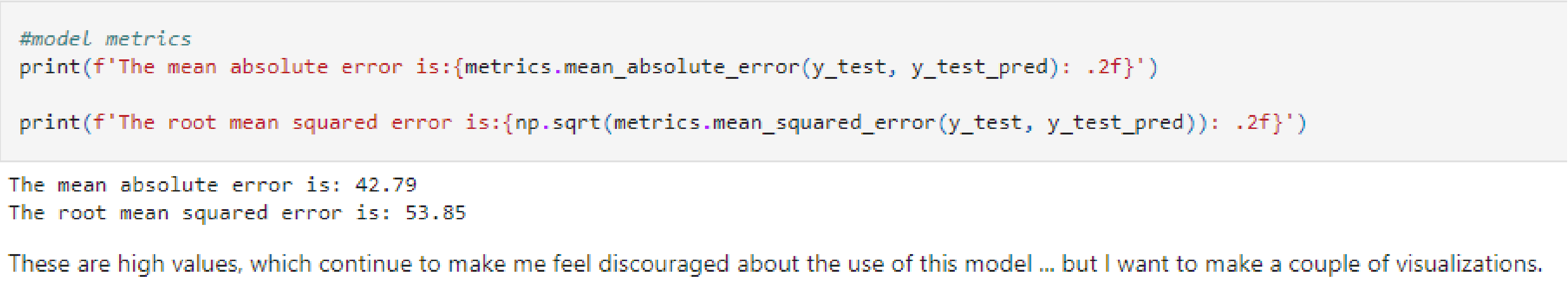
Description of the project

The specifics of the project were to analyze the "diabetes" dataset of the scikit-learn library and to choose a Machine Learning model for predicting the progression of the disease, with the only constraint of choosing a model with at least one hyperparameter to be validated.

I was not satisfied, and a study of different Machine Learning models was born to become familiar with the results of the metrics; I, therefore, learned to understand how to evaluate the numerical results of the various metrics used for model validation.

The Simple Linear Regression Model

I started from the simple linear regression model, studying the MAE, the RMSE, and the coefficient of determination as metrics. I then used the KDE graph for final validation. The results weren't too satisfying, so I switched to the regularized model.

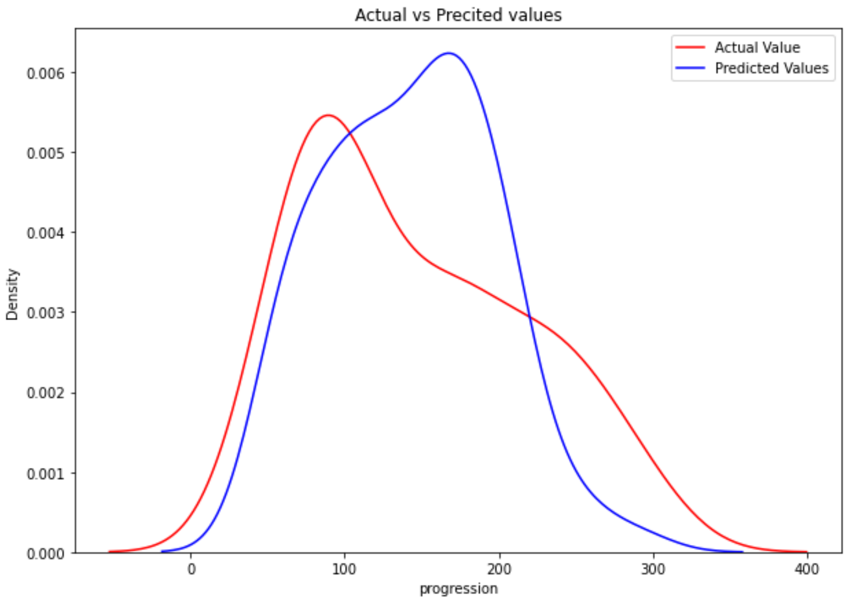


```
#model metrics
print(f'The mean absolute error is:{metrics.mean_absolute_error(y_test, y_test_pred): .2f}')

print(f'The root mean squared error is:{np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)): .2f}')
```

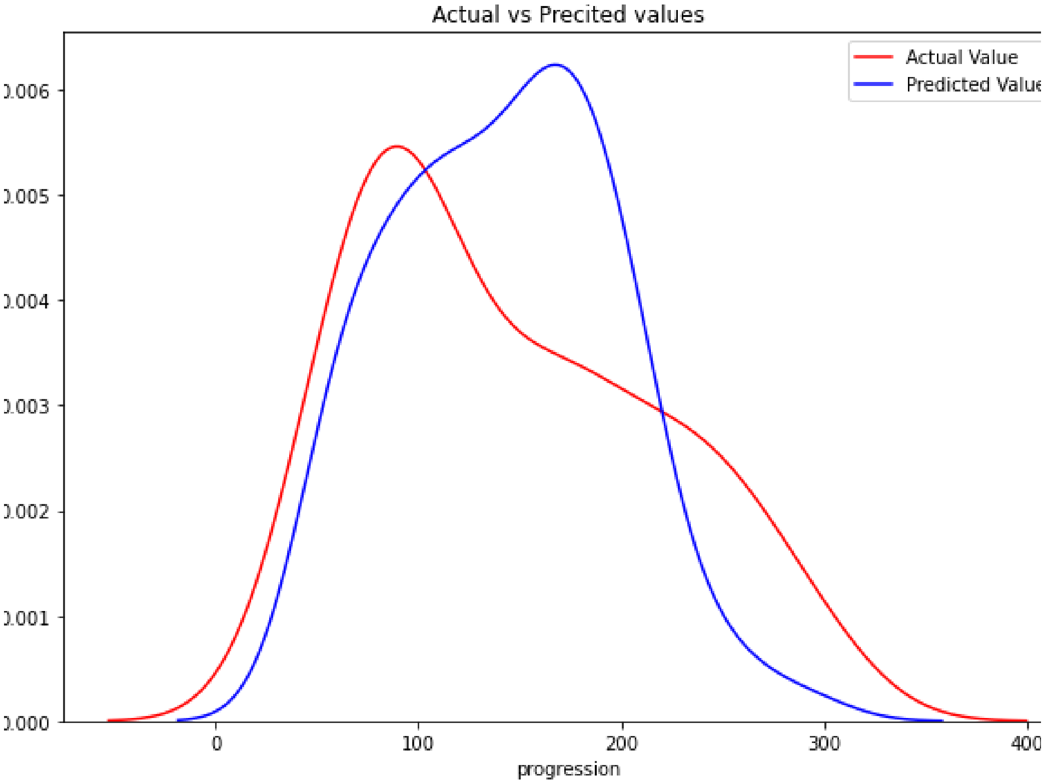
```
The mean absolute error is: 42.79
The root mean squared error is: 53.85
```

These are high values, which continue to make me feel discouraged about the use of this model ... but I want to make a couple of visualizations.



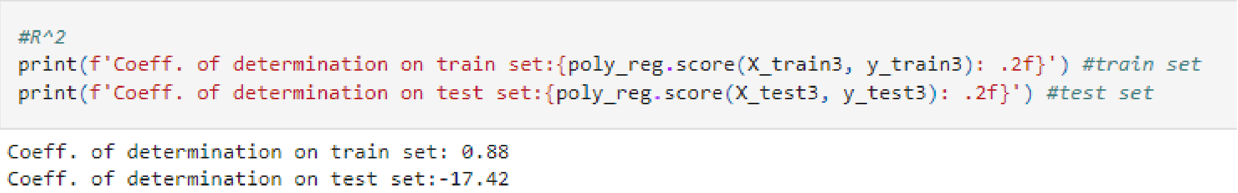
The Lasso-type regularized linear regression model

I tried to apply the Lasso type regularization, but the results obtained were almost identical to those of the non-regularized model.



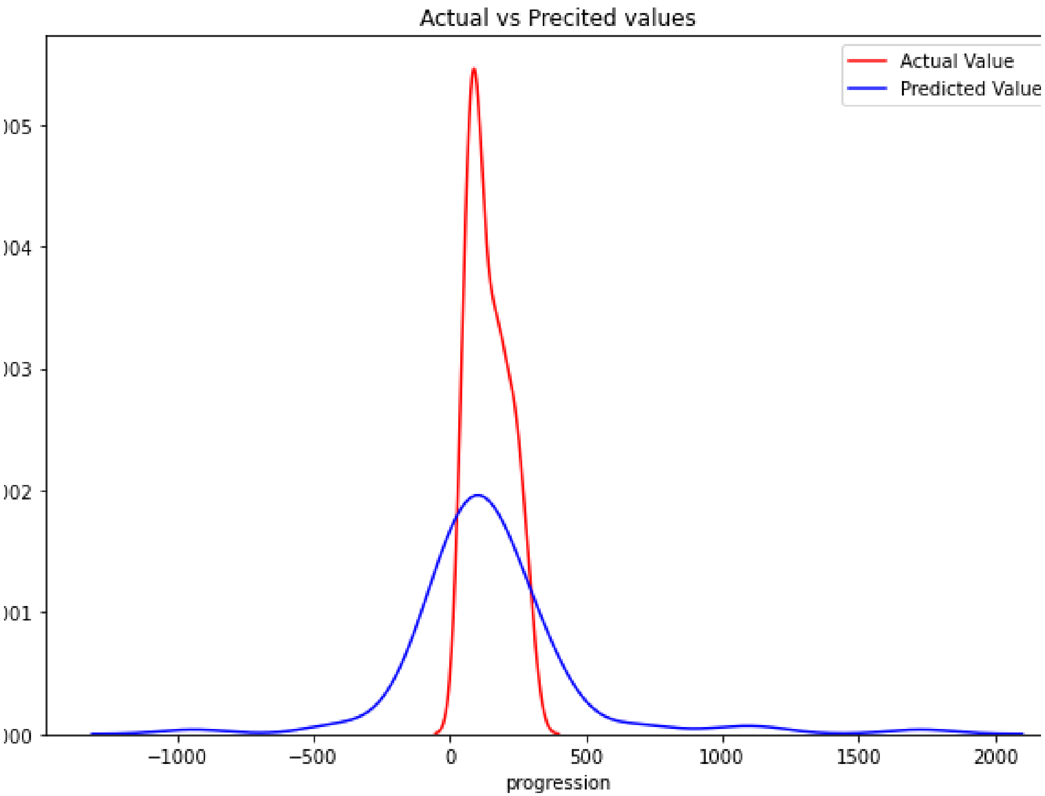
The polynomial regression model

Next, I tried the third degree polynomial regression model, but I got overfitting



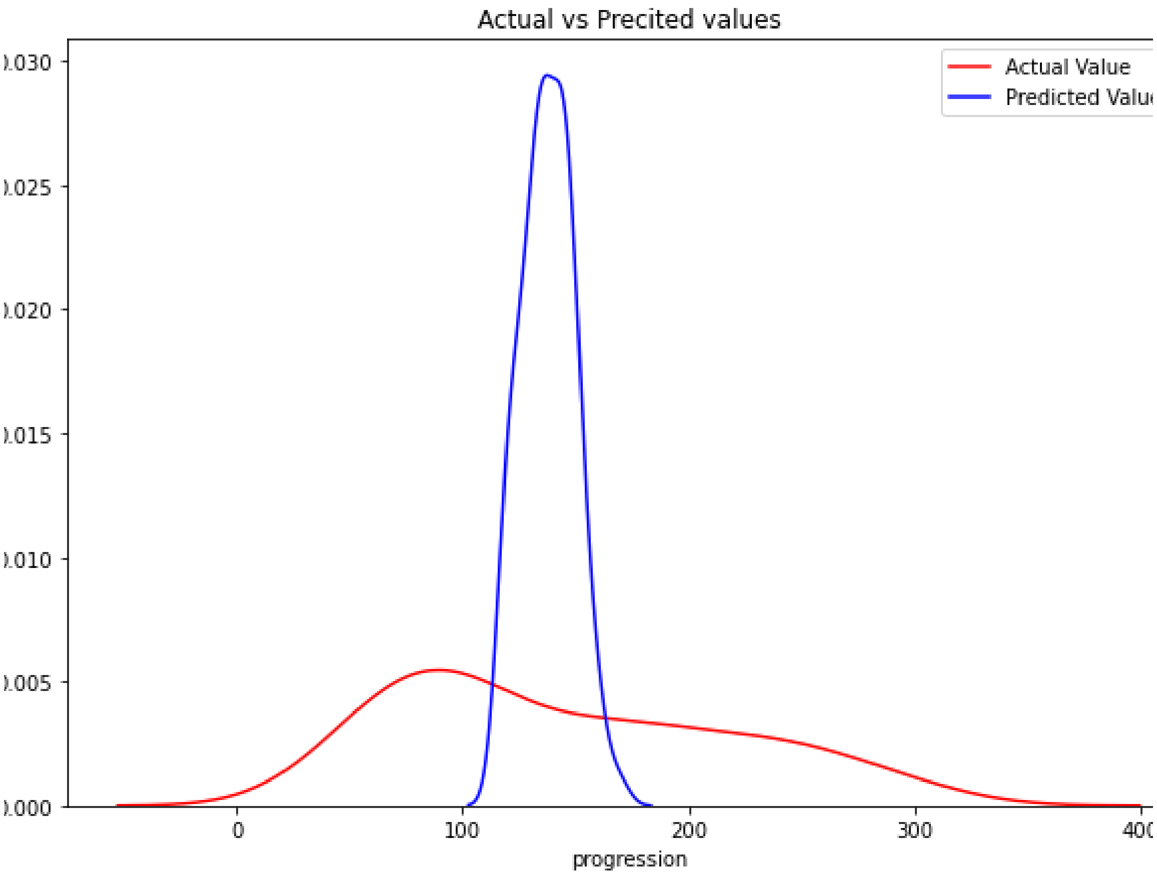
```
#R^2
print(f'Coeff. of determination on train set:{poly_reg.score(X_train3, y_train3): .2f}') #train set
print(f'Coeff. of determination on test set:{poly_reg.score(X_test3, y_test3): .2f}') #test set
```

```
Coeff. of determination on train set: 0.88
Coeff. of determination on test set:-17.42
```



The Support Vector Regression (SVR) model

Finally, I tried the SVR model which, however, was immediately the least suitable, as I obtained practically null values of the coefficient of determination. The KDE graphic then confirmed this initial idea.



Conclusions

This project was a way to become familiar with the various metrics typical of regression problems.

I learned to validate hyperparameters and compare various ML models based on metrics.

Moreover, the simple linear regression model turns out to be not too optimal, but in the end, it turned out to be the best of all the ones I have tried: this taught me to learn how to manage my bias in the study of an ML problem.