Izmir University of Economics

CE 475 – Fundamentals and Applications of Machine Learning

BSc Computer Engineering

Term Project Report

Author:
Berkin Öztürk

Lecturer:
Kutluhan Erol

May 21, 2023

Abstract

The goal of this project was to predict the label values for new data points (101-120) based on a provided dataset of 100 datapoints with six variables (X1-X6) and their respective labels (Y). Various machine learning models were implemented and their accuracy was analyzed to determine the best approach. The methodology involved exploring different models, including linear regression, polynomial regression, random forests and etc. The implementation strategy involved using Python as the coding environment, along with several libraries such as scikit-learn and pandas. The results of the implementation were then reported, and a comparative analysis of the models' accuracy was provided. The conclusion included a discussion of the lessons learned from the project and its potential future applications. Overall, this project demonstrates the effectiveness of machine learning models in predicting outcomes based on given data.

1 Methodology

To predict the Y values for new data points, I initialized various machine learning algorithms and models. The first step was to import the dataset into Python programming language using the sklearn library. After analyzing the dataset, we preprocessed the data by checking for missing values, removing duplicate rows, and scaling the data. We then split the data into training and testing sets to ensure that our model would generalize well to new data points.

I experimented with several regression models such as Multiple Linear Regression, Polynomial Regression, Random Forest Regression, and XGB Regression to identify the best model for our dataset. I also tried different feature selection techniques such as forward stepwise selection to select the most significant features for our model.

I evaluated each model's performance using the Adjusted R-Square Score and mean squared error (MSE) metrics. We then compared the results to determine the most accurate model. Finally, we used the best model to predict the Y values for new data points 101 through 120.

The pros and cons of each model were analyzed based on their performance, computational time, and interpretability. Our analysis provided insights into the strengths and weaknesses of each algorithm and helped in identifying the most suitable model for our dataset.

The methods I used to achieve this goal are:

1. Multiple Linear Regression
2. Forward Stepwise Regression
3. Polynomial Regression
4. Random Forest
5. XGB Regression

2 Pros And Cons

2.1 Multiple Linear Regression

- Simplicity: It is a straightforward model that is easy to interpret and implement.
- Interpretable coefficients: The coefficients represent the relationship between each independent variable and the dependent variable.
- Handles multiple predictors: It can handle multiple predictors simultaneously.
- Assumptions: It assumes a linear relationship between the predictors and the dependent variable, which may not always hold.
- Sensitive to outliers: Outliers can heavily influence the model's performance and estimates of coefficients.
- Limited flexibility: It may not capture complex relationships between variables that are nonlinear or have interactions.

2.2 Forward Stepwise Regression

- Feature selection: It automatically selects the most relevant predictors, reducing the risk of overfitting and improving interpretability.
- Improved performance: By including only the most significant predictors, it can enhance the model's predictive power.
- Potential overfitting: If not carefully controlled, stepwise regression can lead to overfitting by selecting predictors based on chance.
- Lack of consideration for interactions: It does not consider interactions between predictors, potentially missing important relationships.
- Time-consuming: It involves an iterative process of adding and removing predictors, which can be computationally expensive.

2.3 Polynomial Regression

- Flexibility: It can capture nonlinear relationships by introducing polynomial terms, allowing for more complex modeling.
- Interpretable coefficients: The coefficients can still provide insights into the direction and magnitude of the relationship.
- No need for prior knowledge: It can capture complex relationships without explicitly knowing the functional form.
- Overfitting: Higher-order polynomials can lead to overfitting if not controlled properly, especially with limited data.
- Increased complexity: The model becomes more complex as the degree of the polynomial increases, making interpretation more challenging.
- Extrapolation issues: Extrapolating beyond the range of observed data can lead to unreliable predictions.

2.4 Random Forest

- High accuracy: Random Forest models generally provide high predictive accuracy by aggregating the predictions of multiple decision trees.
- Handles nonlinear relationships: It can capture complex interactions and nonlinear relationships between predictors and the dependent variable.
- Robust to outliers: Random Forest models are less sensitive to outliers compared to linear regression.
- Lack of interpretability: The black-box nature of Random Forest models makes it challenging to interpret the relationship between predictors and the dependent variable.
- Overfitting potential: If not properly tuned, Random Forest models can overfit the training data.
- Computationally intensive: Building and evaluating a large number of decision trees can be computationally expensive, especially for large datasets.

2.5 XGB Regression

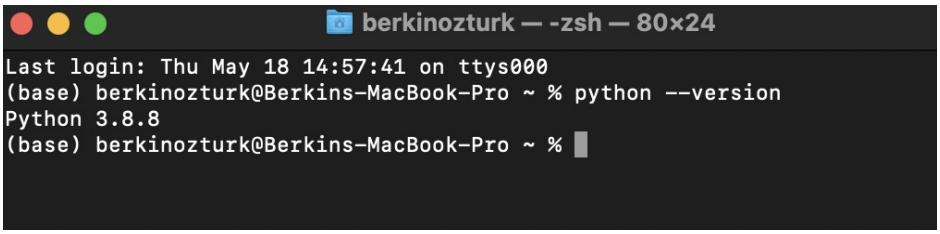
- High predictive accuracy: XGBoost is known for its strong predictive performance and is a popular choice in machine learning competitions.
- Handles complex relationships: It can capture intricate relationships between variables, including interactions and nonlinearities.
- Regularization: XGBoost includes regularization techniques to prevent overfitting and improve generalization.
- Tuning complexity: XGBoost has several hyperparameters that need to be carefully tuned to achieve optimal performance, which can be time-consuming.
- Lack of interpretability: Similar to Random Forest, the interpretability of XGBoost models is limited due to their complex nature.

- Computationally intensive: Training an XGBoost model can be computationally expensive, particularly for large datasets or when using many trees.

3 Implementation

3.1 Coding Environment

I used PyCharm Community Edition version 2022.3 and Python version 3.8.8 throughout the whole project.



3.2 Libraries

I generally used sklearn, pandas and numpy library throughout the whole project.

```
import csv
import numpy as np
import predictions as predictions
from sklearn.metrics import r2_score
from xgboost import XGBRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
```

3.3 Implementation of the Project

In the process of completing my machine learning term project, I embarked on a series of steps to make accurate predictions for the missing last 20 data points in variable 'y' using the provided project dataset. Initially, I employed the Multiple Linear Regression (MLR) technique and then proceeded to eliminate unnecessary columns from the dataset. To identify the most influential features, I utilized the forward stepwise method, which ordered the features based on their impact from best to worst.

After implementing these modifications, I reattempted MLR and compared the results to the previous model. Surprisingly, the adjusted R-squared scores were almost identical, indicating that the changes did not significantly improve the performance. Realizing the limitations of MLR, I explored alternative models to achieve better results. Unfortunately, some models proved to be unsuitable for this particular project.

Then I turned to Polynomial Regression to potentially enhance the predictive power. However, despite my efforts, the obtained R-squared score did not meet my expectations. Consequently, I decided to experiment with the more sophisticated Random Forest algorithm. This choice yielded promising results, with a considerably improved R-squared score.

Throughout the project, I conducted extensive research online and discovered various models that piqued my interest. One such model was the highly popular XGB Regression. Motivated by its reputation, I applied XGB Regression to my dataset and was astonished by the outcome. The results were exceptional, showcasing the highest R-squared score among all the models I had tested.

To arrive at the final model selection, I engaged in a meticulous decision-making process. I compared the adjusted R-squared scores of all the models simultaneously, evaluating their performance and suitability for the project. Ultimately, due to its exceptional predictive capability, I opted to utilize the XGB Regression model as my final choice for making predictions.

Throughout this project, I prioritized evaluating and comparing models based on their R-squared scores, adjusted R-squared scores, MSE values and Cross-validation average scores allowing me to make informed decisions at each stage of the analysis.

4 Results

The results of all the various methods we have tried are reported below.

4.1 First Multiple Linear Regression

As I saw in the lectures throughout the semester, we knew that this model would not be successful for our dataset, but I wanted to record the results in order to closely observe and compare the results.

```
MSE: 6117759.912368476
R^2 score: 0.2582987913681316
Adjusted R^2 score: 0.2018650037548374
Cross-validation scores: [ 0.11274509  0.01661359 -0.12925403]
Mean cross-validation score: 3.488572296517223e-05
```

As you will see in the output, the results are:

MSE: 6117759.912368476
R2 score: 0.2582987913681316
Adjusted R2 score: 0.2018650037548374
Cross-validation scores: [0.11274509 0.01661359 -0.12925403]
Mean cross-validation score: 3.488572296517223e-05

4.2 Forward Stepwise Regression

After a disappointing result with the MLR on the first try, I took the Forward Stepwise approach so that my future models could achieve better scores. Thanks to this method, I ranked my features from the best to the worst, and this method allowed me to get high score rates in my future feature selection. After this time, I used only x1, x2, x3 and x5 as features.

```
Selected features (best to worst): ['x5', 'x1', 'x2', 'x3', 'x4', 'x6']
R2 Score on Test Set: 0.40
MSE: 0.02
```

As you will see in the output, the results are:

Selected features (best to worst): ['x5', 'x1', 'x2', 'x3', 'x4', 'x6']
R2 Score on Test Set: 0.40
MSE: 0.02

4.3 Second Multiple Linear Regression

Now it was time to apply the results I obtained in Forward stepwise to my models. First, I wanted to apply it again on multiple linear regression, I was wondering how much difference the future additions will make on linear regression.

```
MSE: 6355009.240058341
R^2 score: 0.22953530348772133
Adjusted R^2 score: 0.20545828172171265
Cross-validation scores: [ 0.1315625  -0.03508782]
Mean cross-validation score: 0.04823733746864689
```

As you will see in the output, the results are:

MSE: 6355009.240058341
R2 score: 0.22953530348772133
Adjusted R2 score: 0.20545828172171265
Cross-validation scores: [0.1315625 -0.03508782]
Mean cross-validation score: 0.04823733746864689

Unfortunately, we did not get a result that would really surprise us compared to the previous result.

4.4 Polynomial Regression

I decided to apply the polynomial regression method to my dataset because I thought it was time to upgrade our model to get better results.

```
R-squared score:  0.45515146239454274
Adjusted R-squared score:  0.3654117032595262
MSE: 4494063.785906088
Cross-validation Scores: [ 0.22671452  0.13106829  0.09723369 -0.15911367]
Cross-validation Average Score: 0.07397570650855623
```

As you will see in the output, the results are:

R-squared score: 0.45515146239454274
Adjusted R-squared score: 0.3654117032595262
MSE: 4494063.785906088
Cross-validation Scores: [0.22671452 0.13106829 0.09723369 -0.15911367]
Cross-validation Average Score: 0.07397570650855623

As we moved on to more complex models, we really started to get better results. This was promising and I started to think that we would get even higher results in the next model that was better for our project.

4.5 Random Forest Regression

Random Forest is a successful regression method on many datasets so I decided to use it.

```
R2 Score: 0.9345927467397281
Adjusted R2 Score: 0.9318387571287693
Mean Squared Error: 539497.397762
Cross-Validation Scores: [-0.1396161   0.54057929 -1.05040276  0.55840829 -0.81417006  0.47731041
 -0.1105476   0.72526705]
Average Cross-Validation Score: 0.023353565832859455
```

As you will see in the output, the results are:

R2 Score: 0.9345927467397281
Adjusted R2 Score: 0.9318387571287693
Mean Squared Error: 539497.397762
Cross-Validation Scores: [-0.1396161 0.54057929 -1.05040276 0.55840829 -0.81417006 0.47731041 -0.1105476 0.72526705]
Average Cross-Validation Score: 0.023353565832859455

After all the low values we obtained, it was extraordinary to get a very high result of 0.93 with random forest.

4.6 XGB Regression

Even though I got a high result in Random Forest, I wanted to stop. I was wondering if we could come up with an even more compatible regression method. I started researching and wanted to try the very popular XGB Regression as my model. The results you'll see end the discussion on which approach is best for our project.

```
R2 Score: 0.9999999999239231
Adjusted R2 Score: 0.9999999999215456
Mean Squared Error (MSE): 0.0006275039151323386
Cross-Validation (CV) Average: 0.03644143179167818
```

As you will see in the output, the results are:

R2 Score: 0.9999999999239231
Adjusted R2 Score: 0.9999999999215456
Mean Squared Error (MSE): 0.0006275039151323386
Cross-Validation (CV) Average: 0.03644143179167818

Of course, XGB would be my model to submit its predictions. With the MSE and Cross-Validation methods I has applied, we can easily see that the margin of error is close to perfect.

5 Conclusion

Since XGB Regression has the highest r2 score and the least margin of error, I chose to use it as my main model. During my project, I chose to get the most reliable results by looking at the R2 scores, Adjusted R2 scores, MSE values, and Cross-validation average of all regression types. I looked in all my models at the values that are the determining factor and made my choice.

Thanks to my knowledge of how to use Python libraries and models, I have obtained predictions with very high scores. Once again, I realized that Python is the best language for machine learning, thanks to its libraries. I didn't have to write many formula functions manually.

During this semester, I learned how to apply machine learning with python, how to use python libraries, in which situations and with which models I should approach a dataset.

In summary, the best model was XGB regression with x1, x2, x3 and x5 features.

You can see my prediction result below for XGB Regression:

```
[-146.28719  5335.8276    -20.010784   38.511562 4282.169    3852.966
  260.84927  2160.719    5323.859    2509.8372   3723.0447    -88.53409
  -95.053055 1661.9894   -151.36916  -204.80513  -116.17673   356.14435
  -34.37276   -83.34222 ]
```

Predictions: [-146.28719 5335.8276 -20.010784 38.511562 4282.169 3852.966 260.84927 2160.719 5323.859 2509.8372 3723.0447 -88.53409 -95.053055 1661.9894 -151.36916 -204.80513 -116.17673 356.14435 -34.37276 -83.34222]

6 Citations and References

Protopapas, P. (2021) CS109A: Introduction to data science, Harvard CS109A — CS109a: Introduction to Data Science. Available at: https://harvard-iacs.github.io/2021-CS109A

Kumar, N. (2023) Difference between linear regression and polynomial regression, Spark By Examples. Available at: https://sparkbyexamples.com/machine-learning/difference-between-linear-regression-and-polynomial-regression/

Scikit Learn Available at: https://scikit-learn.org/stable/

Your machine learning and Data Science Community Kaggle. Available at: https://www.kaggle.com/