Deployment report

Housing Price Prediction by

Zhaklin Yanakieva

Date: 06/04/2021

Table of contents

Versioning table

Time	Changes	Comments	Version
Week 6			
Week 10			

Abstract

Machine learning research typically focuses on optimization and testing on a few criteria, but deployment in a public policy setting requires more. For machine learning models to have real-world benet and impact, effective deployment is difficult. In this report, I describe the implementation of the deployment. My decision of how the project will be delivered to production is further explained and the evaluation is made after that .

Introduction

The goal of building a machine learning model is to solve a problem, and a machine learning model can only do so when it is in production and actively in use by consumers. As such, model deployment is as important as model building. [1] Deployment is part of the Machine Learning Lifecycle. Broadly, the entire machine learning lifecycle can be described as a combination of 6 stages:

Stage 1: Problem Definition

The first and most important part of any project is to define the problem statement. Here, we want to describe the aim or the goal of our project and what we want to achieve at the end.

Stage 2: Hypothesis Generation

Once the problem statement is finalized, we move on to the hypothesis generation part. Here, we try to point out the factors/features that can help us to solve the problem at hand.

[1] https://stackoverflow.blog/2020/10/12/how-to-put-machine-learning-models-into-production/

Stage 3: Data Collection

After generating hypotheses, we get the list of features that are useful for a problem. Next, we collect the data accordingly. This data can be collected from different sources

Stage 4: Data Exploration and Pre-processing

After collecting the data, we move on to explore and pre-process it. These steps help us to generate meaningful insights from the data. We also clean the dataset in this step, before building the model

Stage 5: Model Building

Once we have explored and pre-processed the dataset, the next step is to build the model. Here, we create predictive models in order to build a solution for the project.

Stage 6: Model Deployment

Once you have the solution, you want to showcase it and make it accessible for others. And hence, the final stage of the machine learning lifecycle is to deploy that model.

Modelling

Machine learning is about predicting and recognizing patterns and generating suitable results after understanding them. ML algorithms study patterns in data and learn from them. An ML model will learn and improve on each attempt. To gauge the effectiveness of a model, it’s vital to split the data into training and test sets first. So before training our models, we split the Lending Club data into a Training set which was 80% of the whole dataset and Test set which was the remaining 20%. Then it was important to implement a selection of performance metrics to the predictions made by our model. In this case, we tried to identify whether an individual is going to default on a loan or not.

Model accuracy might not be the sole metric to identify how our model performed - the F1 score and confusion matrix should be important metrics to analyse as well. What is important is that the right performance measures are chosen for the right situations. We used 3 algorithms for the modelling purpose: [2]

1. Linear Regression:

   In statistics, linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables).[3]

2. Lasso Regression:

   Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters).[4]

3. Ridge Regression:

   Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where independent variables are highly correlated. It has uses in fields including econometrics, chemistry, and engineering.[5]

[2] https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012042/pdf
[3] https://en.wikipedia.org/wiki/Linear_regression
[4] https://www.statisticshowto.com/lasso-regression/
[5] https://en.wikipedia.org/wiki/Ridge_regression

4. Random forest regressor:

Random Forest Regression is a supervised learning algorithm that uses ensemble learning methods for regression. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.[6]

Results:

1. Linear Regression:

```
MAE: 0.3410280435512464
MSE: 0.13885727231548534
RMSE: 0.37263557575127654
```

2. Lasso Regression:

```
MAE: 0.32357276994617423
MSE: 0.1302860917523951
RMSE: 0.3609516473883934
```

[6] https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

3. Ridge Regression:

```
MAE: 0.34128080786593337
MSE: 0.1389431130217031
RMSE: 0.37275073845896417
```

4. Random forest regressor:

```
Mean Absolute Error: 0.2867295244901386
Mean Squared Error: 0.09942510658144314
Root Mean Squared Error: 0.3153174695151589
```

Evaluation of the methods

In this paper, I used three machine learning algorithms: Linear, Lasso and Ridge regression, and Random Forest to find the model with the best performance that will work for the housing price prediction.

The results of both the models are shown in the modelling part above and to get a better understanding of the scores of the two models, I will explain more about them here. The results show that the Random forest is the best fit for this project and should be used for the deployment.

From the exploring of the models accuracy:

- Linear Regression score: 0.80 (80%)
- Lasso score: 0.82 (82%)
- Ridge score: 0.86 (86%)
- Random forest score: 98.13 %

From the exploring of the models RMSE:

- Linear Regression score: 0.2003 (0.1887)
- Lasso score: 0.5 (0.4675)
- Ridge score: 0.2 (0.1877)
- Random forest score: 0.2372

> RMSE values between 0.2 and 0.5 shows that the model can relatively predict the data accurately. All of the models showed values in this range.

Random forest turns out to be the more accurate model for predicting the house price.

All of the models showed RMSE values between 0.2 and 0.5 so that they show relatively accurate predictions of the data.

In the end, I tried three different models and evaluated them using Mean Absolute Error. I chose MAE because it is relatively easy to interpret and outliers aren't particularly bad for this type of model. The one I will be using for the deployment is the Random forest.

From the exploring of the models cross-validation:

- Linear Regression score: R2: 0.7308604883584712
- Lasso score: R2: 0.6532616143265344
- Ridge score: R2: 0.7310756447849953
- Random forest: R2: 0.7742740242196954

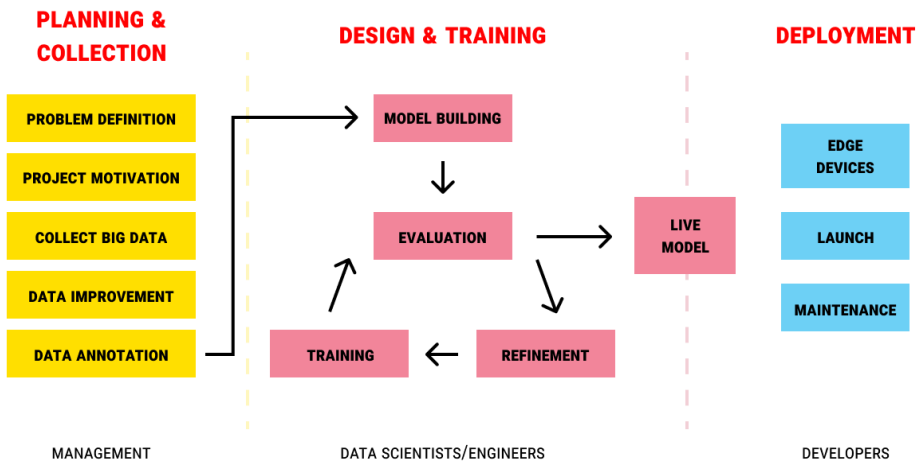
Deployment

The last step before the AI project is done is the deployment. After the training of models, one is selected, the one with the best performance. And to do the deployment, I will use Streamlit, which is a recent and the simplest way of building web apps and deploying machine learning and deep learning models.

First, I will explain the tool I am using for the deployment and after that, I will explain the deployment itself.

What is deployment?

Model deployment generally contains two parts, frontend, and backend. The backend is generally a working model, a machine learning model, which is built-in python. And the front end part, which generally requires some knowledge of other languages like java scripts, etc. I decided to deploy the project as a web app using Flask.



What is flask API?

Flask is a web framework for Python, meaning that it provides functionality for building web applications, including managing HTTP requests and rendering templates.[7] It is designed as a web framework for RESTful API development.

Environment and tools:

1. scikit-learn
2. pandas
3. numpy
4. flask

Summary

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient component. This application is working properly and will meet all people's expectations. This component can be easily plugged in many other systems. There have been cases of computer glitches, errors in content and most important weight of features is fixed in an automated prediction system, so in the near future the software could be made more secure, reliable and dynamic weight adjustment. In near future this module of prediction can be integrate with the module of automated processing systems.

[7] https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask