Machine Learning Engineer Nanodegree

Capstone Proposal

Catapult project

Contents

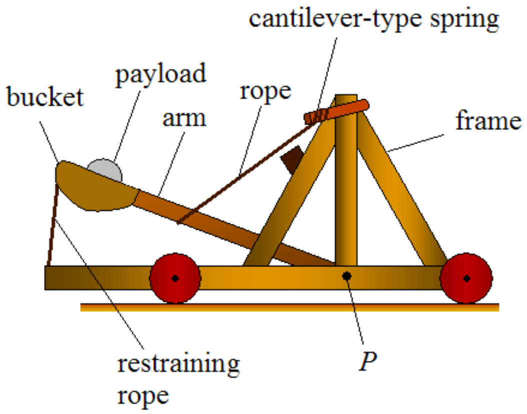
--

--

Domain Background

A catapult is a ballistic device used to launch a projectile a great distance without the aid of gunpowder or other propellants – particularly various types of ancient and medieval siege engines. A catapult uses the sudden release of stored potential energy to propel its payload. Most convert tension or torsion energy that was more slowly and manually built up within the device before release, via springs, bows, twisted rope, elastic, or any of numerous other materials and mechanisms. The counterweight trebuchet is a type of catapult that uses gravity. Machine Learning could be a good solution to predict the distance of a projectile, so you don't have to run the experiment whenever you change one of the inputs. Using Machine Learning you will save the money and the time that require to perform the experiment.

I don't know if there is an academic paper where machine learning was applied to this type of problem. The idea came to my mind when I was completing the final project and I saw there is an option that you can implement your project. I have studied the catapult as part of the Design of Experiment subject in the college and then I thought that I can build a Machine Learning model to predict the distance. It will be much easier.



Problem statement

The goal of the project is to predict the distance between the projectile when it will hit the ground and the catapult depending on several inputs which are release angle, firing angle, cup elevation, pin elevation, and bungee position. The output is the distance

Datasets and Inputs

For this project, the input data is collected from an online simulator of a catapult by changing the five inputs of the experiment (release angle, firing angle, cup elevation, pin elevation, and bungee position). Noise is considered and added to the experiment output. By taking the average of Distance without noise and Distance with noise. The average distance is the output data. About 460 experiments are done to collect the datasets for this project.

Example of the dataset with the noise distance.

Release angle	Firing angle	Cup elevation	Pin elevation	Bungee pos	Distance With out noise	Distance with noise	Average distance
185	90	300	200	100	196.99	207.89	202.44
185	95	300	200	100	243.40	251.52	247.16
185	100	300	200	100	287.92	305.08	296.5
185	105	300	200	100	331.42	345.32	338.37

Example of the dataset after taking the average of the distance.

Release angle	Firing angle	Cup elevation	Pin elevation	Bungee pos	distance
185	90	300	200	100	202.44
185	95	300	200	100	247.16
185	100	300	200	100	296.5
185	105	300	200	100	338.37

Simulator link: https://sigmazone.com/catapult/

Catapult grid interference to collect the data: https://sigmazone.com/catapult-grid/

Solution Statement

Trying different models and different hyperparameters are always a good solution to see and decide which model is fit better for the data. Using XGBoost and by adjusting the hyperparameters to find the best training job.

Benchmark Model

K-Nearst Neighbors model will be used as the benchmark model. Knn algorithm is an index-based algorithm. It uses a non-parametric method for classification or regression. For regression problems, the algorithm queries the *k* closest points to the sample point and returns the average of their feature values as the predicted value. Training with the k-NN algorithm has three steps: sampling, dimension reduction, and index building. Sampling reduces the size of the initial dataset so that it fits into memory

Evaluation Metrics

Root Mean Square Error will be the main metric used to test the XGBoost model and model after the hyperparameter Tuning. While accuracy will be metric to test the knn model

Project Design

Step 1:

- Collect the data using the catapult simulator.
- Take the average of the output distance using excel.

Step 2:

- Load the catapult dataset to the Jupiter notebook.
- Clean the data and remove any repeated or empty rows.
- Distribute the value for each feature.
- Separate the input data and the output data.
- Separate the data to train and test using Sklearn.

Step 3:

- Locate the data directory and save the test, train and validation to csv files.
- Upload the csv files to s3 to use them later in the model.
- Build a XGBoost model using SageMaker.
- Deploy the using the model for prediction.

Step 4:

- Use SageMaker hyperparameter tuning.
- Find the best training job.
- Deploy the adjusted model for prediction and compare it with old model

Step 5:

- Prepare the input dataset and scale it.
- Create the K-Nearst Neighbors model.
- Fit the model with input and validation data.
- Test the model and predict the output from the test datasets.

References

1.Catapult Background: https://en.wikipedia.org/wiki/Catapult
2.Catapult simulation for data collection: https://sigmazone.com/catapult
3.Knn Algorithm: https://docs.aws.amazon.com/sagemaker/latest/dg/k-nearest-neighbors.html