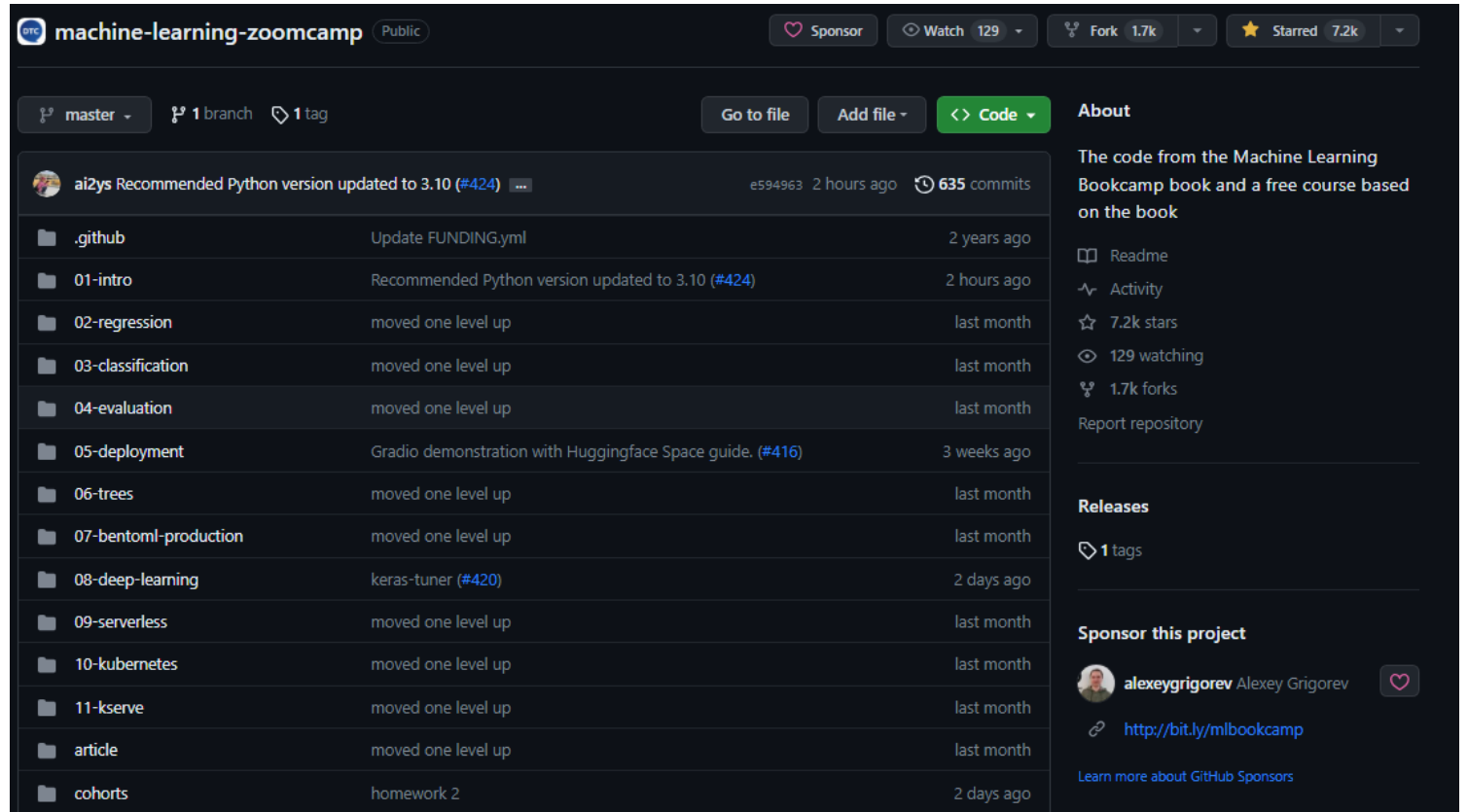


Machine Learning Zoomcamp

Data Talks Club: [Link](#)



The screenshot shows the GitHub repository page for `machine-learning-zoomcamp`. The repository is public and has 129 watchers, 1.7k forks, and 7.2k stars. It is currently on the `master` branch with 1 branch and 1 tag. The repository contains a list of files and folders, including `.github`, `01-intro`, `02-regression`, `03-classification`, `04-evaluation`, `05-deployment`, `06-trees`, `07-bentoml-production`, `08-deep-learning`, `09-serverless`, `10-kubernetes`, `11-kserve`, `article`, and `cohorts`. The repository is sponsored by Alexey Grigorev, who is also the author of the Machine Learning Bookcamp book and a free course based on the book. The repository is also available as a Readme, Activity, and 7.2k stars.

machine-learning-zoomcamp Public

Sponsor Watch 129 Fork 1.7k Starred 7.2k

master 1 branch 1 tag Go to file Add file Code

ai2ys Recommended Python version updated to 3.10 (#424) e594963 2 hours ago 635 commits

.github	Update FUNDING.yml	2 years ago
01-intro	Recommended Python version updated to 3.10 (#424)	2 hours ago
02-regression	moved one level up	last month
03-classification	moved one level up	last month
04-evaluation	moved one level up	last month
05-deployment	Gradio demonstration with Huggingface Space guide. (#416)	3 weeks ago
06-trees	moved one level up	last month
07-bentoml-production	moved one level up	last month
08-deep-learning	keras-tuner (#420)	2 days ago
09-serverless	moved one level up	last month
10-kubernetes	moved one level up	last month
11-kserve	moved one level up	last month
article	moved one level up	last month
cohorts	homework 2	2 days ago

About

The code from the Machine Learning Bookcamp book and a free course based on the book

Readme Activity 7.2k stars 129 watching 1.7k forks Report repository

Releases

1 tags

Sponsor this project

alexeygrigorev Alexey Grigorev <http://bit.ly/mlbookcamp> Learn more about GitHub Sponsors

- ML is the process of extracting patterns from data
- Data is of 2 types Features and Target(Prediction from Model)



Supervised learning

- The goal of SL is to come up with a model g which takes a **feature** matrix 'x' as input and learn to **predict** the output as close to the **target** example 'y'

1. Predict a numerical value: Regression

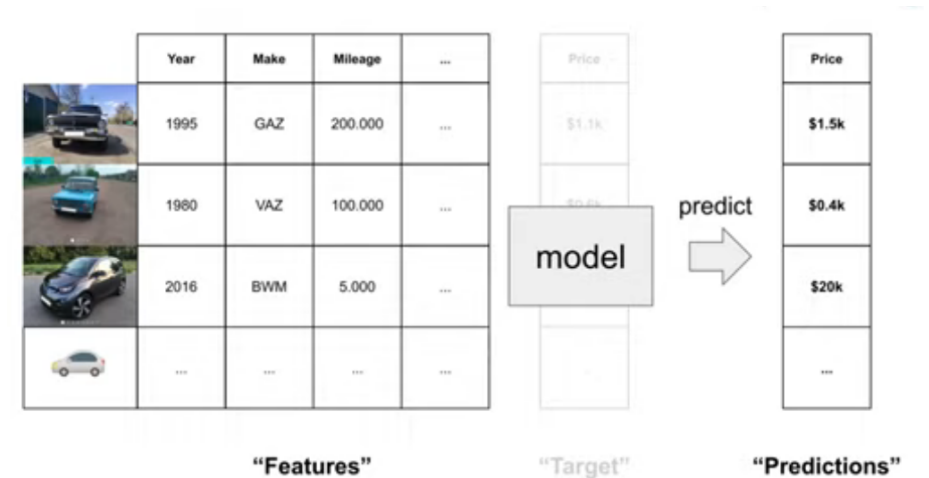
- Predicts a continuous number
- Eg: Price , Area, Height etc.

2. Predict a category: Classification

- Predicts a category
- Binary classification only gives one of 2 output categories
- Multicategory Classification has more than 2 options as classification output
- Eg: Spam or not, True false,

3. Ranking: Recommender Systems

- Rank the result based on the score assigned
- Eg: E commerce, Google search



$$g(\mathbf{x}) \approx y$$

model FEATURES TARGET

CRISP-DM

Cross Industry Standard Process for Data Mining by IBM

1. Business Understanding

- Understand the extent of problem and decide suitability of ML based system
- Define Goal
- Goal has to be measurable (Eg: reduce the spam detection by 80%)

2. Data Understanding

- Identify data collection and replication method
- Need to add more data or not by **going back to the previous step**

3. Data Preparation

- Clean data to be ready to apply ML models
- Tabular formatting of data with features and target variable labelling

4. Modelling

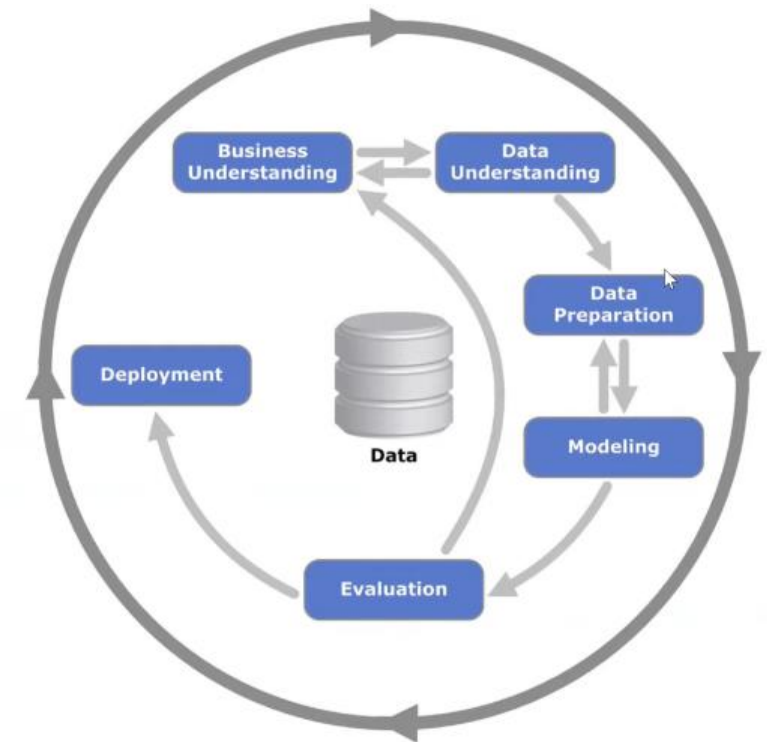
- Machine Learning Step
- If new features or problems with data detected, **go back to the previous step** and reformat data.

5. Evaluation

- **Compare result** with goals
- Decide whether it is acceptable

5. Deployment

- Deploy to users
- Test maintainability through monitoring
- Go back to business goals for further improvement



Model Selection

1. Model Evaluation process:

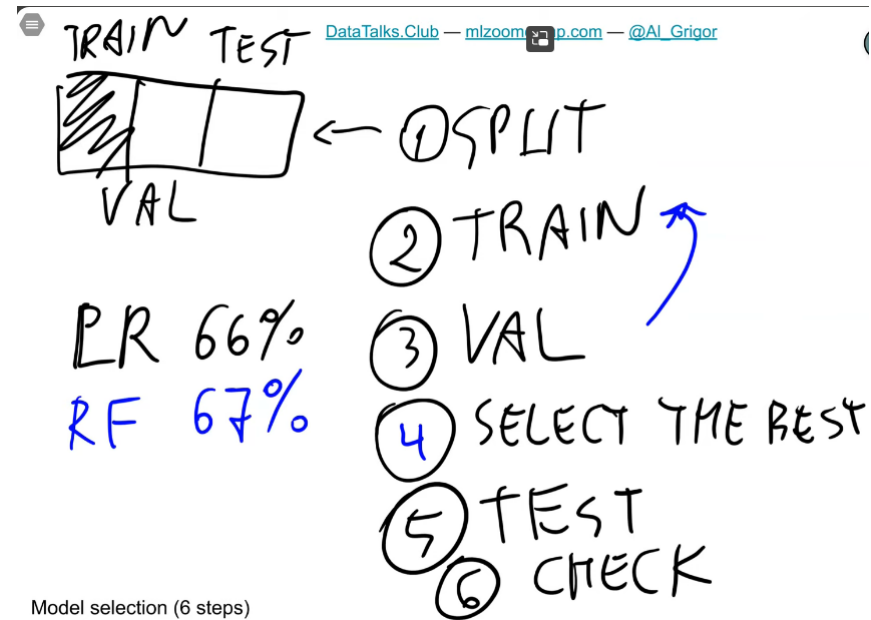
- Divide dataset into 2 parts. Training data to train the models and test data to validate the results of the models.
- In practice it might not work
- A model may luckily give good result on the validation dataset.

2. Multiple Comparison Problem

- When testing many models against a single validation data, one of the models may turn out lucky out of no reason.
- We divide our dataset into 3 parts instead of 2. Train data, Validate data, and finally a test data which will be used after validation to avoid the model from being lucky twice.

3. Solution

- After selecting a good model, retrain with 80 % of the data by combining training data with validation data to get a good final model.
- Finally use the test data on improved model to validate the results



Dataset(100%)			
Train 80%	Test 20%		X
Train 60%	Validate 20%	Test 20%	✓

Homework

Set up the environment

You need to install Python, NumPy, Pandas, Matplotlib and Seaborn. For that, you can the instructions from [06-environment.md](#).

Question 1

What's the version of Pandas that you installed?

You can get the version information using the `__version__` field:

`pd.__version__`

A 1.3.4

Getting the data

For this homework, we'll use the California Housing Prices dataset. Download it from [here](#).

You can do it with `wget`:

`wget https://raw.githubusercontent.com/alexeygrigorev/datasets/master/housing.csv`

Or just open it with your browser and click "Save as...".

Now read it with Pandas.

Question 2

How many columns are in the dataset?

•10

•6560

•10989

•20640

Question 3

Which columns in the dataset have missing values?

•total_rooms

•total_bedrooms

•both of the above

•no empty columns in the dataset

Question 4

How many unique values does the `ocean_proximity` column have?

•3

•5

•7

•9

Question 5

What's the average value of the `median_house_value` for the houses located near the bay?

•49433

•124805

•259212

•380440

Question 6

1.Calculate the average of `total_bedrooms` column in the dataset.

2.Use the `fillna` method to fill the missing values in `total_bedrooms` with the mean value from the previous step.

3.Now, calculate the average of `total_bedrooms` again.

4.Has it changed?

Has it changed?

Hint: take into account only 3 digits after the decimal point.

•Yes

•No

Question 7

1.Select all the options located on islands.

2.Select only columns `housing_median_age`, `total_rooms`, `total_bedrooms`.

3.Get the underlying NumPy array. Let's call it `X`.

4.Compute matrix-matrix multiplication between the transpose of `X` and `X`. To get the transpose, use `X.T`. Let's call the result `XTX`.

5.Compute the inverse of `XTX`.

6.Create an array `y` with values `[950, 1300, 800, 1000, 1300]`.

7.Multiply the inverse of `XTX` with the transpose of `X`, and then multiply the result by `y`. Call the result `w`.

8.What's the value of the last element of `w`?

Note: You just implemented linear regression. We'll talk about it in the next lesson.

•-1.4812

•0.001

•5.6992

•23.1233

