



LeetCode difficulty estimator

Neural networks semestral project

Mihal Filip; Trappl Juraj; 2024



- popular platform for developers
- rich community
- content
 - Contests
 - Mock interviews for jobs
 - Problems/Tasks
 - Forum discussions
 - Courses



Our task

- Given a description of a problem, predict its difficulty
 - Easy / Medium / Hard

1. Two Sum

Easy  54151  1803  Add to List  Share

Difficulty (target)



Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have **exactly one solution**, and you may not use the *same* element twice.

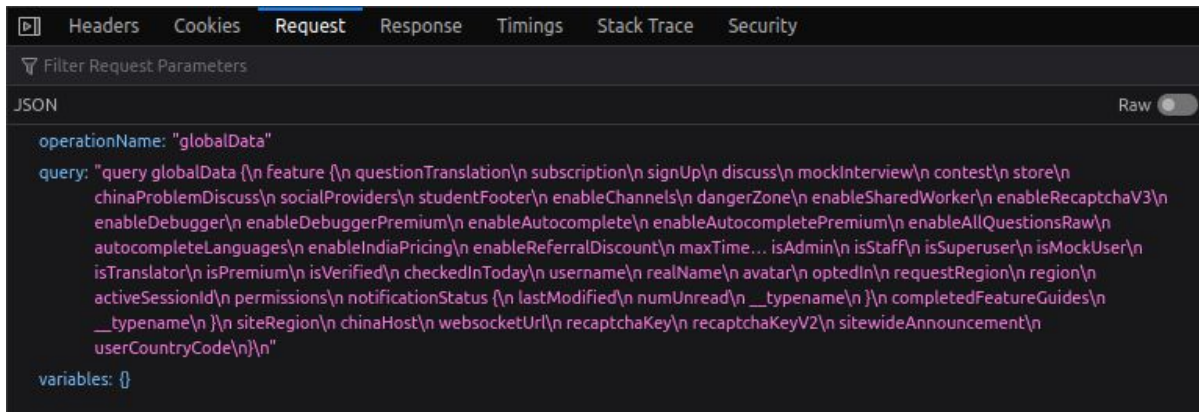
You can return the answer in any order.



Textual description (data)

How to obtain the data?

- Query the LeetCode internal API



The screenshot shows the 'Request' tab in a web browser's developer tools. The 'JSON' tab is selected, displaying a GraphQL query. The query is a single-line string with escaped characters. The variables are an empty object. The 'Raw' toggle is turned off.

```
operationName: "globalData"
query: "query globalData {\n  Feature {\n    questionTranslation\n    subscription\n    signUp\n    discuss\n    mockInterview\n    contest\n    store\n    chinaProblemDiscuss\n    socialProviders\n    studentFooter\n    enableChannels\n    dangerZone\n    enableSharedWorker\n    enableRecaptchaV3\n    enableDebugger\n    enableDebuggerPremium\n    enableAutocomplete\n    enableAutocompletePremium\n    enableAllQuestionsRaw\n    autocompleteLanguages\n    enableIndiaPricing\n    enableReferralDiscount\n    maxTime... isAdmin\n    isStaff\n    isSuperuser\n    isMockUser\n    isTranslator\n    isPremium\n    isVerified\n    checkedInToday\n    username\n    realName\n    avatar\n    optedIn\n    requestRegion\n    region\n    activeSessionId\n    permissions\n    notificationStatus {\n      lastModified\n      numUnread\n      __typename\n    }\n    completedFeatureGuides\n    __typename\n  }\n  siteRegion\n  chinaHost\n  websocketUrl\n  recaptchaKey\n  recaptchaKeyV2\n  sitewideAnnouncement\n  userCountryCode\n}\n"
variables: {}
```





Querying the API

```
graphql_request_problems_list = leetcode.GraphqlQuery(  
    query="""  
        query problemsetQuestionList($categorySlug: String, $limit: Int, $skip: Int, $filters: QuestionListFilterInput) {  
            problemsetQuestionList: questionList(categorySlug: $categorySlug, limit: $limit, skip: $skip, filters: $filters) {  
                total: totalNum  
                questions: data {  
                    acRate  
                    difficulty  
                    title  
                    titleSlug  
                    topicTags {  
                        name  
                        id  
                        slug  
                    }  
                    hasSolution  
                }  
            }  
        }  
    """,  
    variables=variables,  
    operation_name="problemsetQuestionList"  
)
```

1. Query the list of problems names

```
# For each problem, query the difficulty and description.  
result = {}  
for title_slug in problems:  
    variables = {  
        "titleSlug": title_slug  
    }  
  
    graphql_request = leetcode.GraphqlQuery(  
        query="""  
            query questionData($titleSlug: String!) {  
                question(titleSlug: $titleSlug) {  
                    content  
                    difficulty  
                }  
            }  
        """,  
        variables=variables,  
        operation_name="questionData",  
    )  
  
    api_response = api_instance.graphql_post(body=graphql_request).to_dict()["data"]["question"]  
    result[title_slug] = {  
        "difficulty": api_response["difficulty"],  
        "content": api_response["content"]  
    }  
}
```

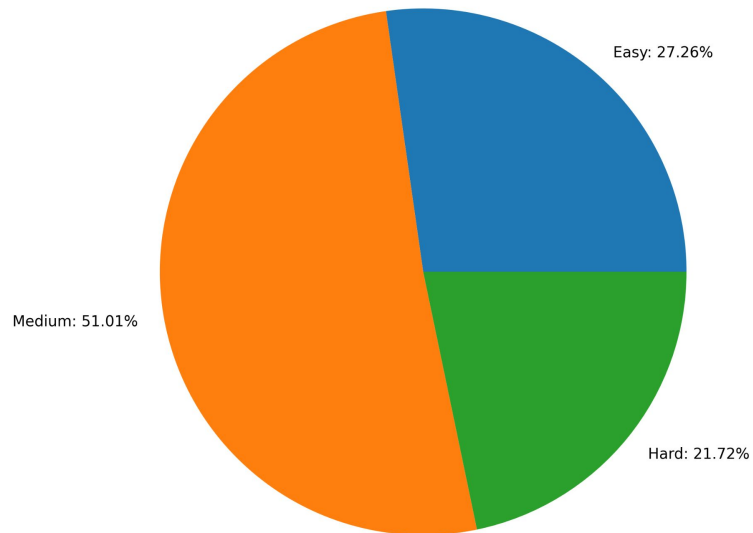
2. For each problem query the data



Dataset

- 2950 problems
 - ~584 problems for premium users
- **2366** examples to train

Distribution of Problem Difficulties



Understanding training examples

"content": "<p>Given an array of integers <code>nums</code> and an integer <code>target</code>, return indices of the two numbers such that they add up to <code>target</code>.</p>\n\n<p>You may assume that each input would have exactly one solution, and you may not use the same element twice.</p>\n\n<p>You can return the answer in any order.</p>\n\n<p> </p>\n\n<strong class=\"example\">Example 1:</p>\n\n<pre>\nInput: nums = [2,7,11,15], target = 9\nOutput: [0,1]\nExplanation: Because nums[0] + nums[1] == 9, we return [0, 1].\n</pre>\n\n<strong class=\"example\">Example 2:</p>\n\n<pre>\nInput: nums = [3,2,4], target = 6\nOutput: [1,2]\n</pre>\n\n<strong class=\"example\">Example 3:</p>\n\n<pre>\nInput: nums = [3,3], target = 6\nOutput: [0,1]\n</pre>\n\n<p> </p>\n\nConstraints:</p>\n\n\n<code>2 \leq nums.length \leq 10</code>\n<code>-10⁹ \leq nums[i] \leq 10⁹</code>\n<code>-10⁹ \leq target \leq 10⁹</code>\n\n\n<p>Only one valid answer exists.</p>\n\n<p> </p>\n\nFollow-up: Can you come up with an algorithm that is less than <code>O(n^2)</code> complexity?"

Lots of HTML clutter

Sometimes there are examples mentioned

Sometimes required complexity is mentioned

- What is relevant? What should be considered as features?



Primary text feature extraction

- Tf-Idf
 - Word-level & character-level ngrams
- Word embeddings
 - Static
 - GloVe
 - fastText
 - Contextualized
 - BERT

$$w_{x,y} = \text{tf}_{x,y} \times \log \left(\frac{N}{\text{df}_x} \right)$$

TF-IDF

Term x within document y

$\text{tf}_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents



Shallow learning models

- 80:20 train:test
- Tf-idf
 - ~400k features (1-3 word-level, 1-5 character level)
- class imbalance problem
 - Oversampling (SMOTE)
 - balanced batches
 - **downsampling**

Classification

(1 out of 3 classes)

- Perceptron
- SVM linear kernel
- MLP Classifier

Regression

(prediction on a scale from 0 to 2)

- MLP Regressor

Perceptron

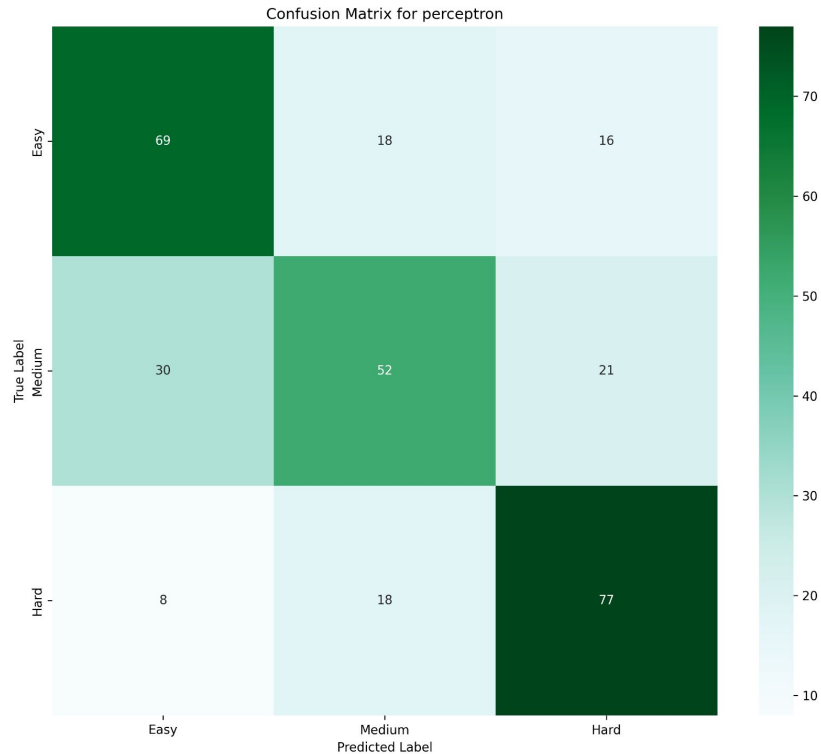
Best model:

- Tf-idf
 - 1 to 5 character level n-grams
 - 1 to 3 word level n-grams
- SelectPercentile(percentile=50)
- L2 penalty
- alpha 0.0001

Stratified k-fold, k = 5

Average test macro f1: **58.26**

Average test micro f1: **58.56**



Linear SVM

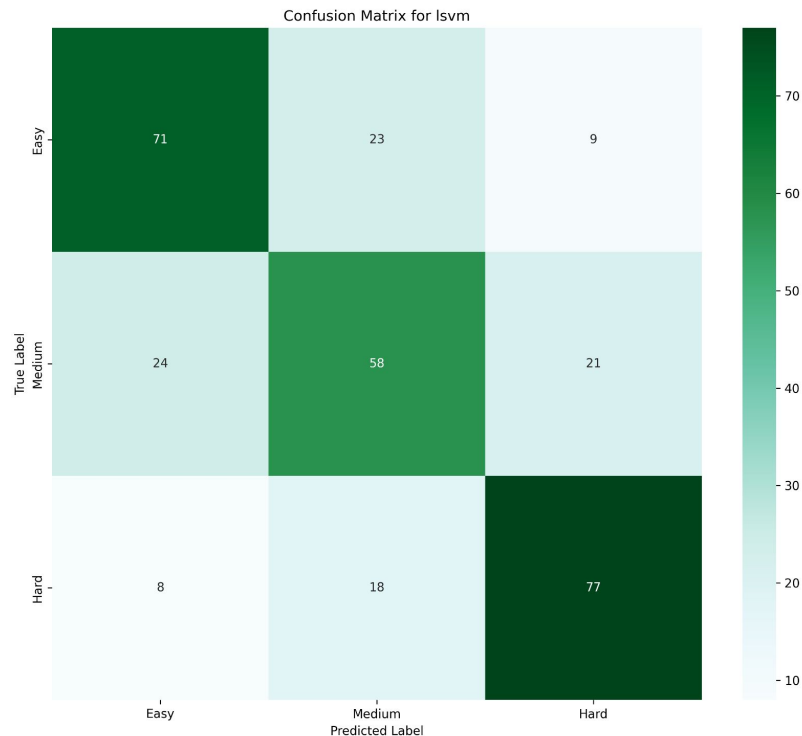
Best model:

- Tf-idf
 - 1 to 5 character level n-grams
 - 1 to 3 word level n-grams
- SelectPercentile(percentile=5)
- Square hinge loss
- L2 penalty
- alpha 0.0001

Stratified k-fold, k = 5

Average test macro f1: **60.94**

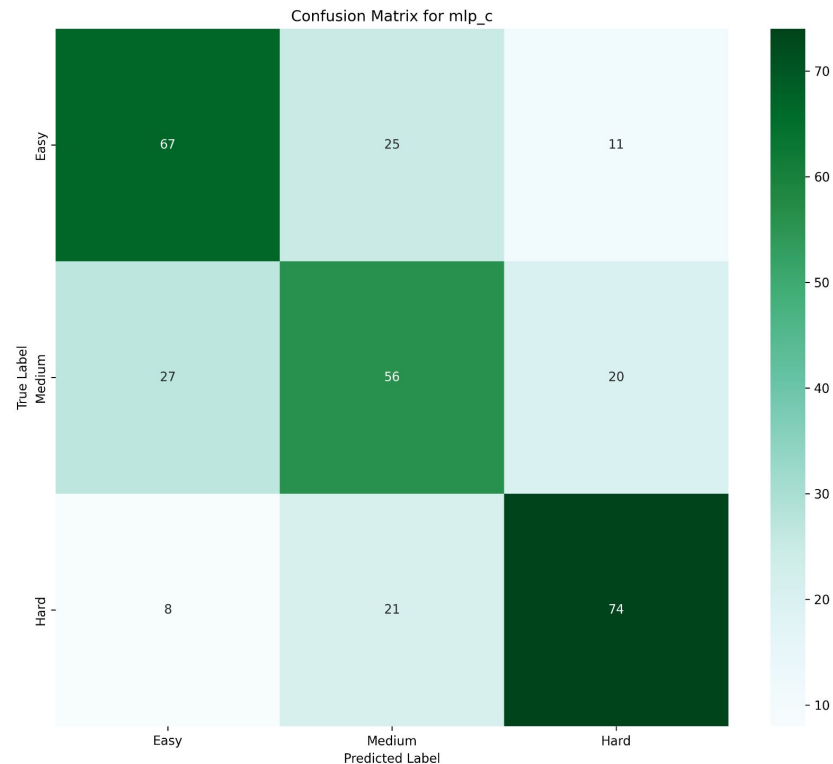
Average test micro f1: **61.15**



MLP Classifier

- Adam optimizer, momentum 0.9 with Nesterov, ReLU, Tf-Idf word & character 3-grams, 5-grams
- Stratified k-fold, k = 5
- SelectPercentile(percentile=5)

Hidden layer(s) size	Average test macro f1	Average test micro f1
<u>(64.)</u>	<u>59.42</u>	<u>59.61</u>



MLP regressor

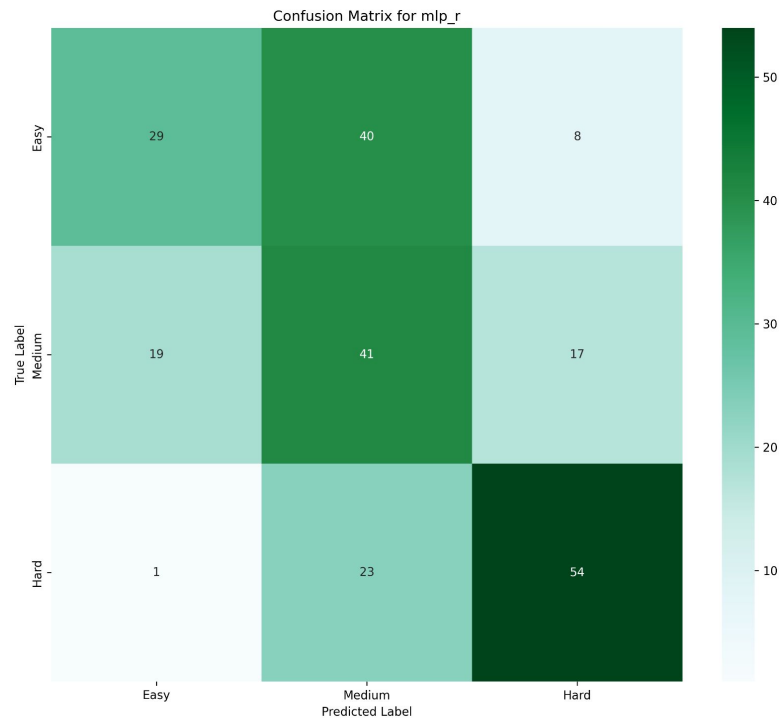
linear scale

0 - easy, 1 - medium, 2-hard

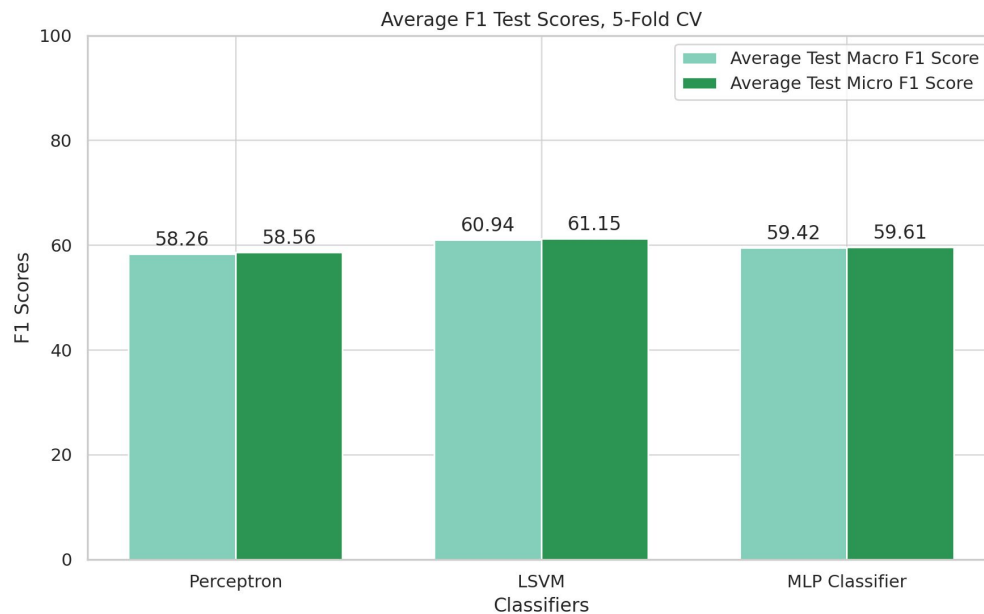
rounding the predictions 0.75, 1.25 thresholds

Test RMSE: **0.675**

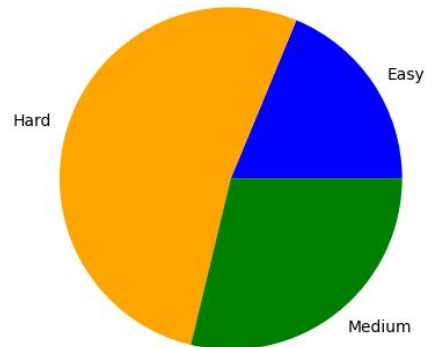
Test accuracy: **53.4%**



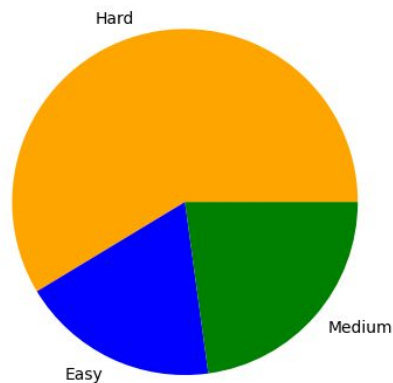
Shallow learning summary



Distribution of difficulties of problems that contain "maximum"



Distribution of difficulties of problems that contain "minimum"



Inference problems

- Difficulty may seem subjective
 - Easy vs. Medium / Medium vs. Hard?
- We want to minimize Easy vs. Hard problems
 - Easy problems predicted being Hard
 - Hard problems predicted being Easy
- An example of a common problem
 - Hard to classify "min-max" problems correctly

2078. Two Furthest Houses With Different Colors

Easy 865 24 Add to List Share

There are n houses evenly lined up on the street, and each house is beautifully painted. You are given a 0-indexed integer array `colors` of length n , where `colors[i]` represents the color of the i^{th} house.

Return the **maximum** distance between **two** houses with **different** colors.

The distance between the i^{th} and j^{th} houses is $\text{abs}(i - j)$, where $\text{abs}(x)$ is the **absolute value** of x .

Predicted as hard by each classifier.

Classification using contextualized embeddings

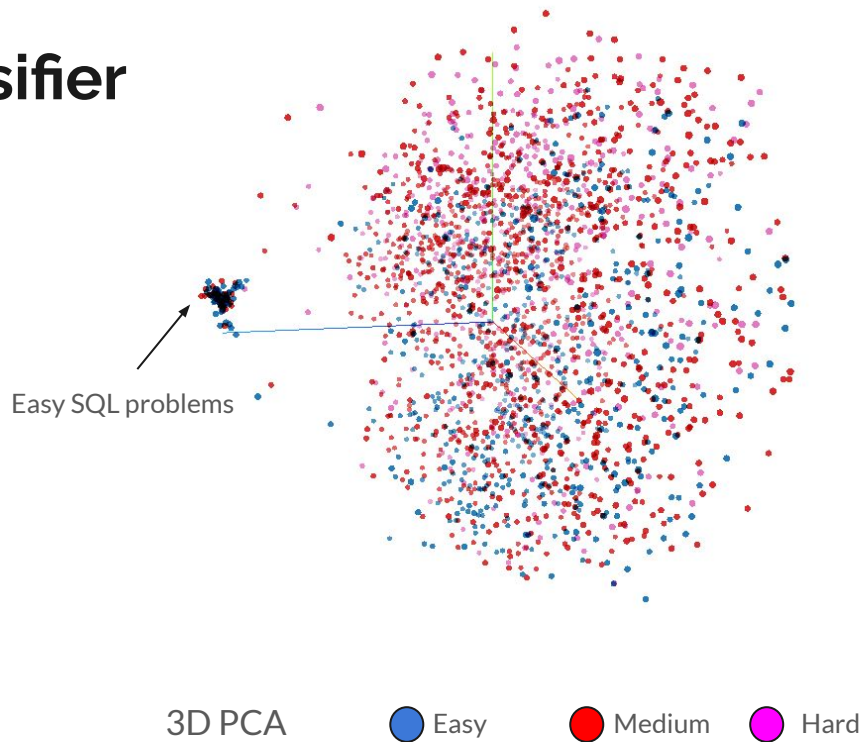
- Tensorflow 2.12, KerasTuner
 - [Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization](#)
- Pre-trained [bert-base-uncased](#)
 - No fine-tuning
- General training information
 - Batch size 32
 - Datasets sizes 1079:231:232 samples (70:15:15)
- On average, contextualized representations are more context-specific in higher layers
 - <https://ai.stanford.edu/blog/contextual/>



1. layer BERT dense classifier

Features

- 1. Layer BERT embeddings
- mean pooling
- (768,)

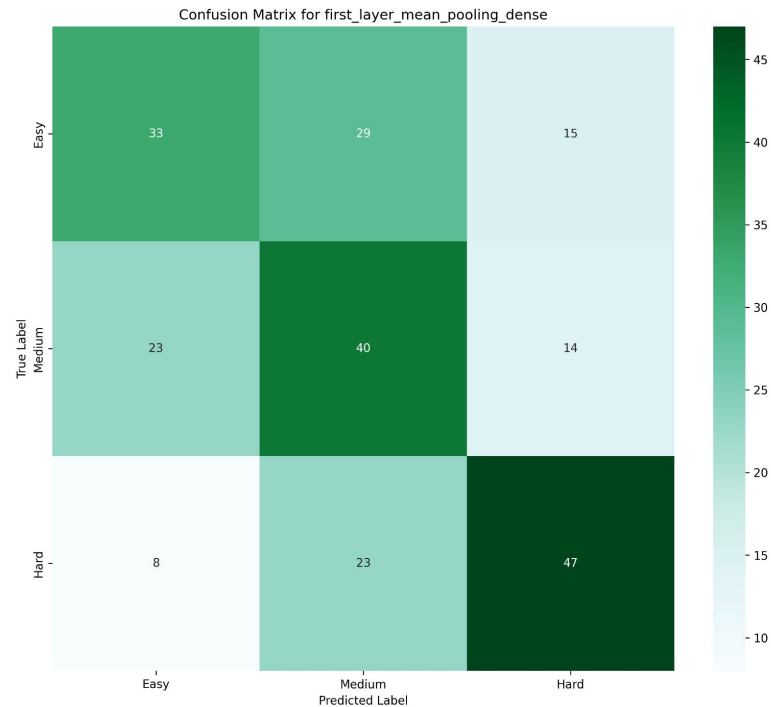


1. layer BERT dense classifier

Model

- Classification head
 - 1 x Dense
 - 576 (relu)
 - Adam optimizer
 - no Batch Norm
 - initial lr 0.01 with cosine decay
 - dropout 0.4
 - L1 ~0.00004
 - L2 ~0.0003

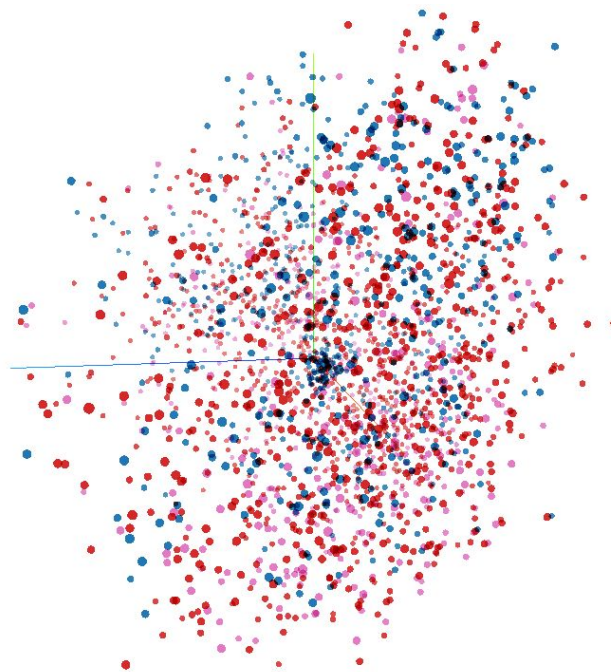
Test accuracy: **51.7%**



2. layer BERT dense classifier

Features

- 2. Layer BERT embeddings
- mean pooling
- (768,)



3D PCA

● Easy

● Medium

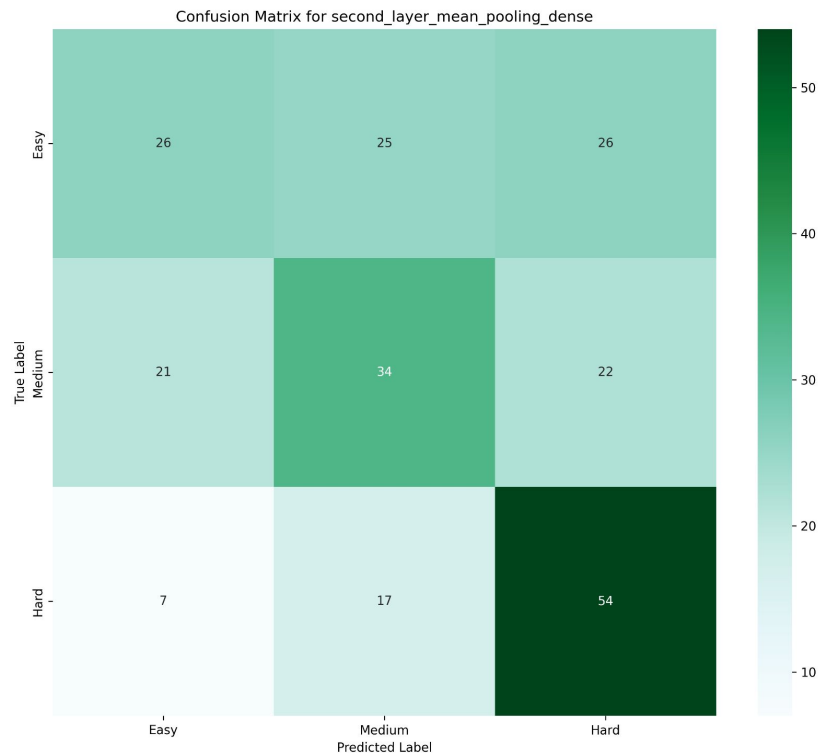
● Hard

2. layer BERT dense classifier

Model

- Classification head
 - 1 x Dense
 - 960 (selu)
 - SGD optimizer, momentum 0.9 with Nesterov
 - no Batch Norm
 - initial lr 0.01 with cosine decay
 - dropout 0.1
 - L1 ~2.25
 - L2 ~1.27

Test accuracy: **49.1%**






Improvements

- Custom tokenization
 - E.g. handle mathematical expressions
- Fine-tune LLM first, then extract embeddings
- Try other models like ELMo, GPT-2 for comparison
- In-context learning classification



Showcase




You are given an integer n . Return the sum of first n fibonacci numbers.

Constraints:


$1 \leq n \leq 100$

Best model prediction: **Easy**



You are given a hard task. Compute average of 2 numbers a and b .

Best model prediction: **Easy**



I am losing my interest in human beings; in the significance of their lives and their actions. Someone has said it is better to study one man than ten books. I want neither books nor men; they make me suffer. Can one of them talk to me like the night – the Summer night? Like the stars or the caressing wind? The night came slowly, softly, as I lay out there under the maple tree. It came creeping, creeping stealthily out of the valley, thinking I did not notice. And the outlines of trees and foliage nearby blended in one black mass and the night came stealing out from them, too, and from the east and west, until the only light was in the sky, filtering through the maple leaves and a star looking down through every cranny.

You are given an integer n . Return the sum of first n fibonacci numbers.

Constraints:

$1 \leq n \leq 100$

Best model prediction: **Medium**



Thank you for your attention!