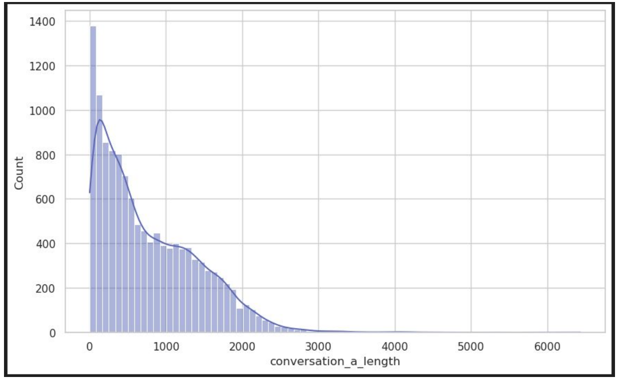
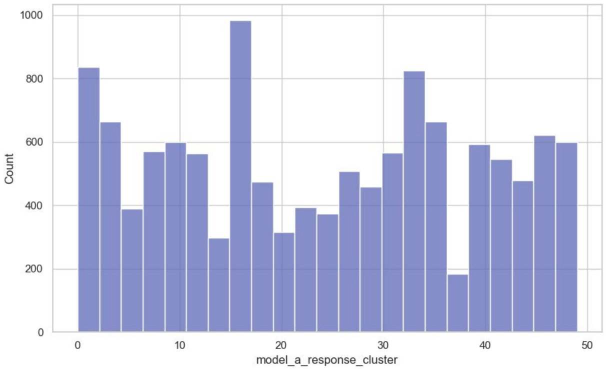
Natural Language Processing Final Project

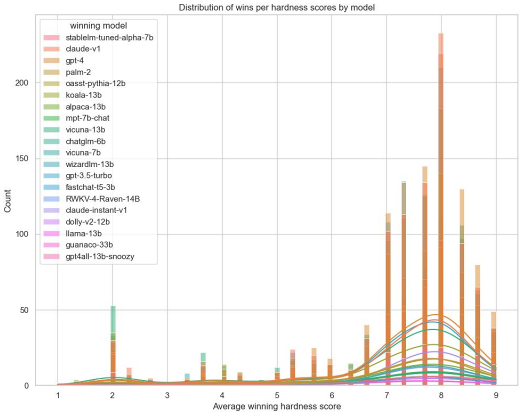
Austin Ly, Joyce Yu, Sam McCarthy-Potter

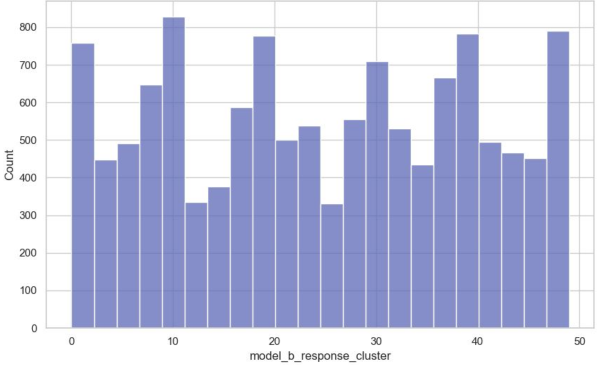
EDA

- Inspecting the Data
- Kmeans clustering
- Analysis of Hardness Score



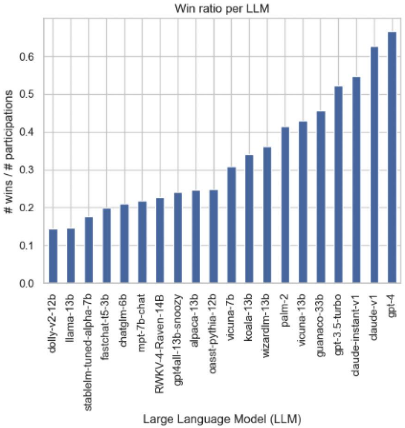


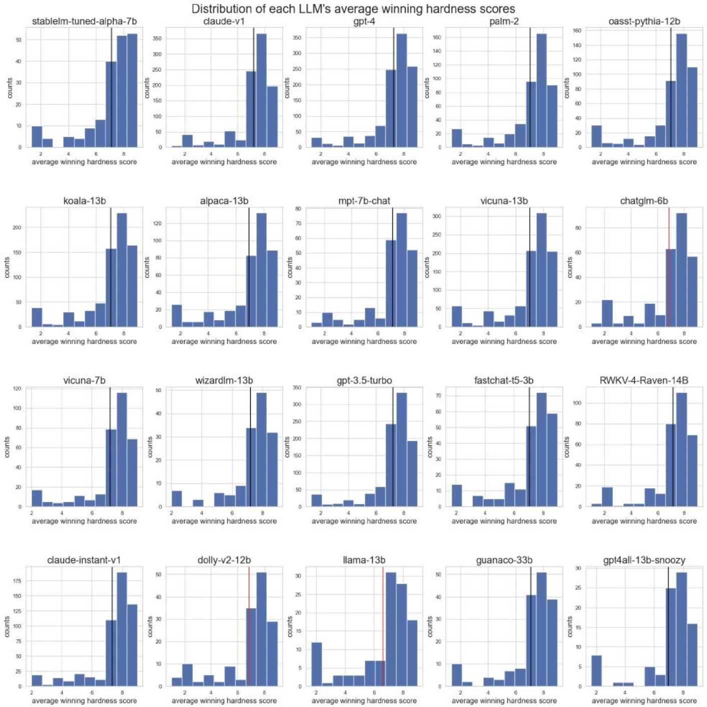




EDA (cont.)

- Feature Analysis
  - One-Hot Encoding
  - Win Frequency
- Model Visualization
- Hardness score distribution
  - Dolly-v2-12b
  - llama-13b





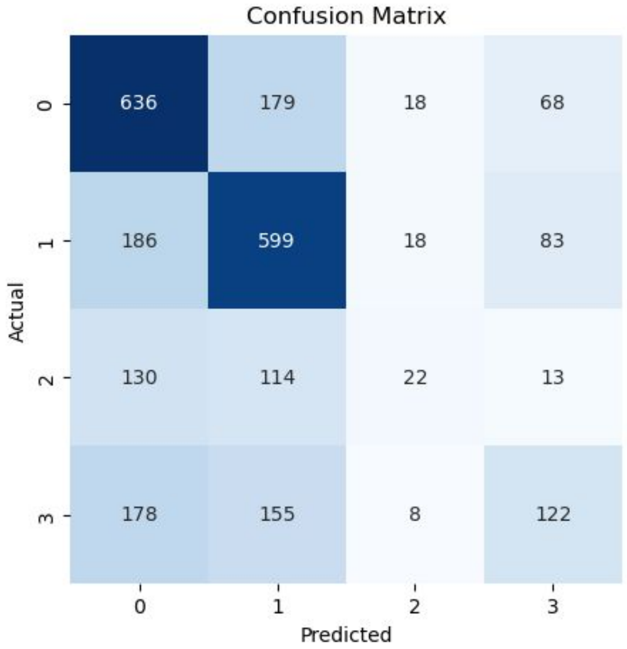
Task A - Predicting Winning Model

Features engineered:

- Elo Rating
- Prompt embeddings
- Model A response embeddings
- Model B response embeddings
- Winner encoded
- Models encoded and matched with Elo

Target: Winner

Task A - Predicting Winning Model



Actual \ Predicted	0	1	2	3
0	636	179	18	68
1	186	599	18	83
2	130	114	22	13
3	178	155	8	122

Model: logistic regression model

Train-Test 90/10 split Stratified due infrequency of ties

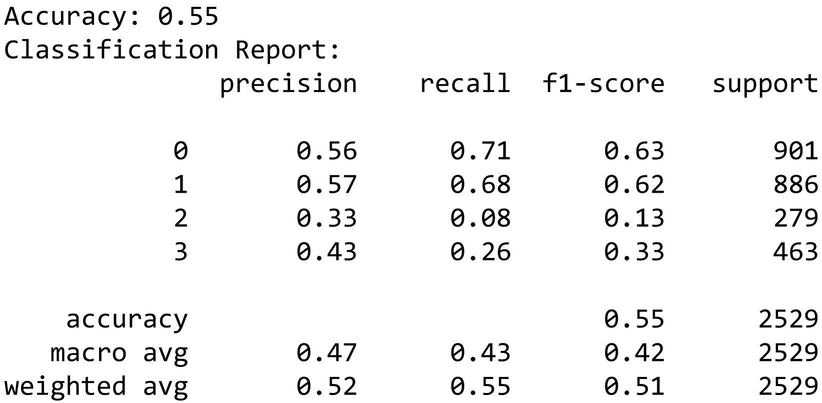
```
X_train, X_test, y_train, y_test = train_test_split(
```

```
X, y, test_size=0.1, random_state=40, stratify=y)
```

Evaluated

Accuracy and F1-scores

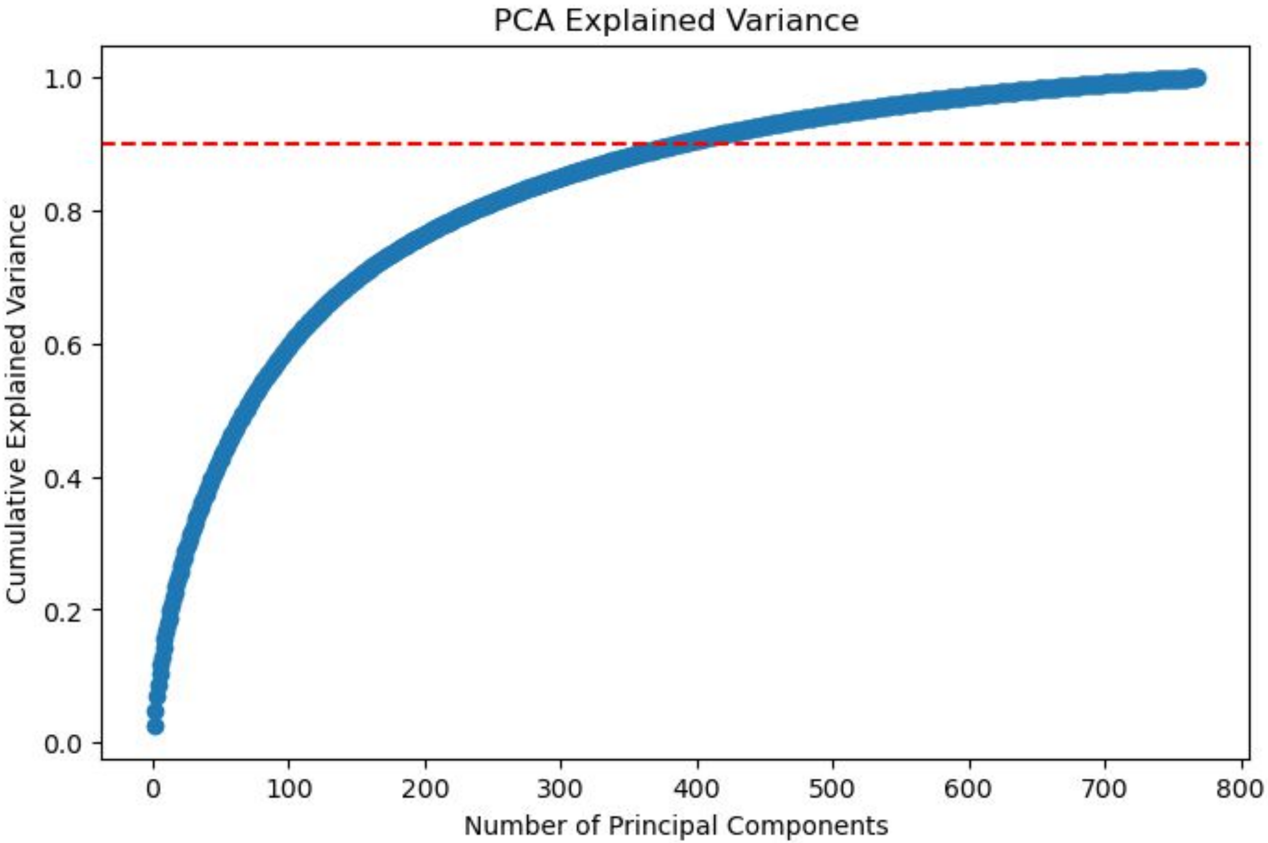
Precision Recall



```
Accuracy: 0.55
Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.71      0.63       901
           1       0.57      0.68      0.62       886
           2       0.33      0.08      0.13       279
           3       0.43      0.26      0.33       463

    accuracy                           0.55      2529
   macro avg       0.47      0.43      0.42      2529
weighted avg       0.52      0.55      0.51      2529
```

Principal Component Analysis Task A

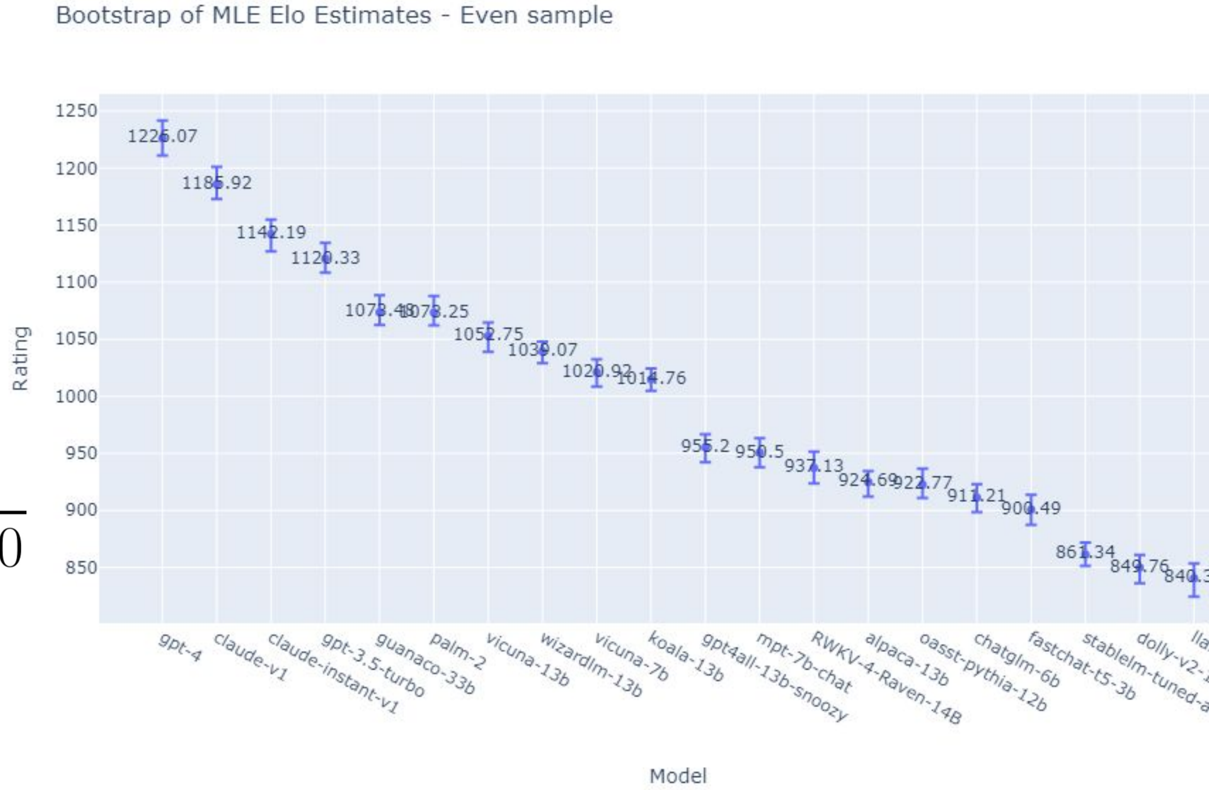
```
plt.plot(range(1, 769),
np.cumsum(pca.explained_varianc
e_ratio_), marker='o',
linestyle='--')
plt.xlabel('Number of Principal
Components')
plt.ylabel('Cumulative
Explained Variance')
plt.title('PCA Explained
Variance')

plt.axhline(y=0.9, color='r',
linestyle='--', label="90%
Variance")
```



Elo calculation

Elo rating is assigned to measure how well a model has done of the course of all the matches it has been in.

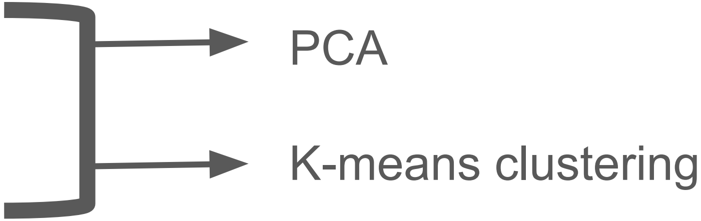
$$\underbrace{E_A}_{\text{Expected score of player A}} = \frac{1}{1 + 10^{\underbrace{(R_B - R_A)}_{\text{Difference between Elo score of B and A}}/400}}$$



Task B - Interpreting difficulty level (i.e. hardness score) of a given prompt / question.

Features engineered:

- Prompt embeddings
- Model A response embeddings
- Model B response embeddings
- One-hot encoded the top most frequent topics
- Average winning hardness score rounded to the nearest integer



PCA

K-means clustering

Target: Average winning hardness score rounded to the nearest integer

Task B - Interpreting hardness score

Features engineered:

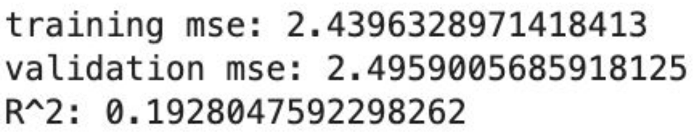
- One-hot-encoded top 500 categorical topics from ‘topic_modeling_2’ column
- **PCA of design matrix (one-hot-encoded top 500 topics)**
- Average hardness score rounded to the nearest integer

Features used for training: One-hot-encodings

Target: Average hardness score rounded to the nearest integer

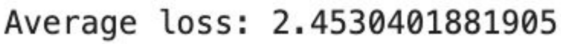
Model: Linear Regression

Evaluation Metric: MSE

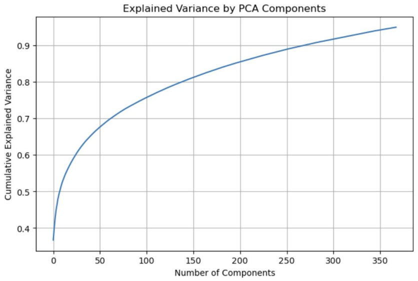


```
training mse: 2.4396328971418413
validation mse: 2.4959005685918125
R^2: 0.1928047592298262
```

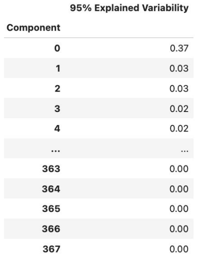
K-Fold Cross Validation (k = 10):



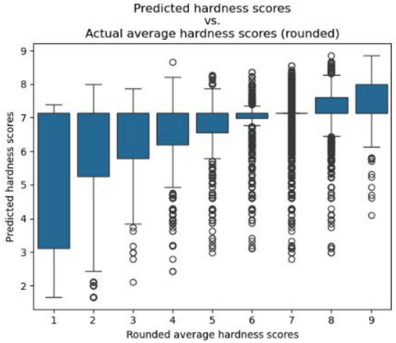
```
Average loss: 2.4530401881905
```



Explained Variance by PCA Components

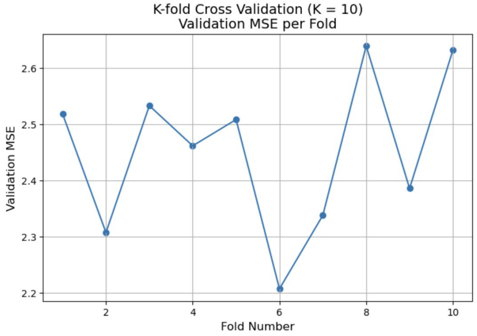


	95% Explained Variability
Component	
0	0.37
1	0.03
2	0.03
3	0.02
4	0.02
...	...
363	0.00
364	0.00
365	0.00
366	0.00
367	0.00



Predicted hardness scores
vs.
Actual average hardness scores (rounded)

avg y	avg y^	abs(avg y - avg y^)
1	5.62	4.62
2	6.12	4.12
3	6.40	3.40
4	6.60	2.60
5	6.75	1.75
6	6.86	0.86
7	7.07	0.07
8	7.31	0.69
9	7.46	1.54



K-fold Cross Validation (K = 10)
Validation MSE per Fold

Task B - Interpreting hardness score

Features engineered:

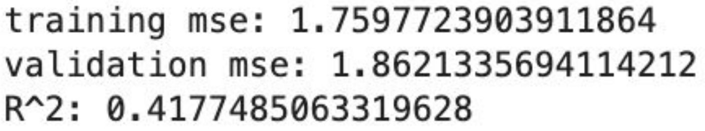
- One-hot-encoded top 500 categorical topics from ‘topic_modeling_2’ column
- **PCA of design matrix (One-hot-encoded top 500 most frequently occurring topics**
- **and prompt embeddings)**
- Average hardness score rounded to the nearest integer

Features used for training: One-hot-encodings and Prompt embeddings

Target: Average hardness score rounded to the nearest integer

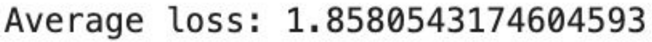
Model: Linear Regression

Evaluation Metric:

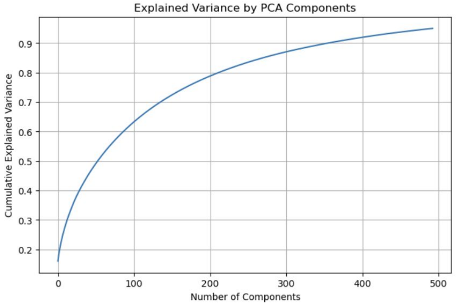


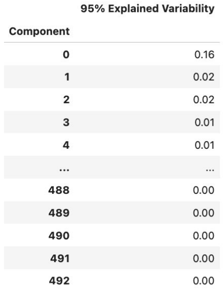
```
training mse: 1.7597723903911864
validation mse: 1.8621335694114212
R^2: 0.4177485063319628
```

K-Fold Cross Validation (k = 10):



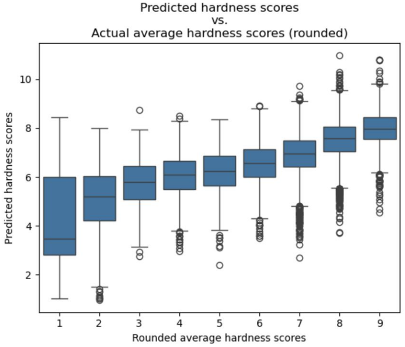
```
Average loss: 1.8580543174604593
```

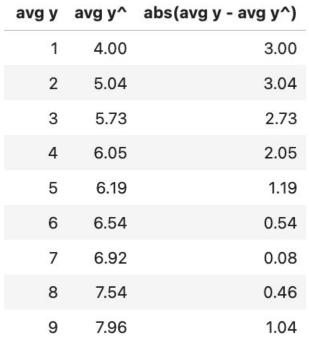




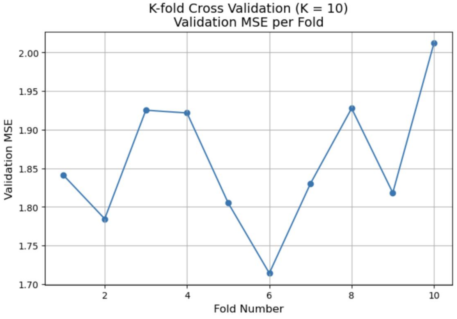
95% Explained Variability

Component	
0	0.16
1	0.02
2	0.02
3	0.01
4	0.01
...	...
488	0.00
489	0.00
490	0.00
491	0.00
492	0.00





avg y	avg y^	abs(avg y - avg y^)
1	4.00	3.00
2	5.04	3.04
3	5.73	2.73
4	6.05	2.05
5	6.19	1.19
6	6.54	0.54
7	6.92	0.08
8	7.54	0.46
9	7.96	1.04



Task B - Interpreting hardness score

Features engineered:

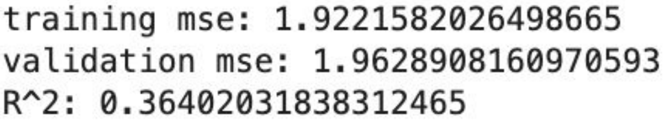
- One-hot-encoded top 10 categorical topics from ‘topic_modeling_1’ column
- **One-hot-encoded top 10 most frequently occurring topics**
- **K-means clustering (k=50)** of each embedded datasets
- Average hardness score rounded to the nearest integer

Features used for training: Prompt clusters, response A clusters, response B clusters, one-hot-encoded top topics)

Target: Average hardness score rounded to the nearest integer

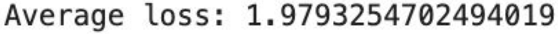
Model: Linear Regression

Evaluation Metric:

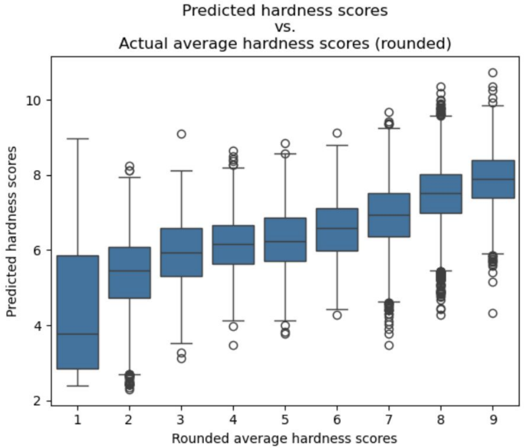


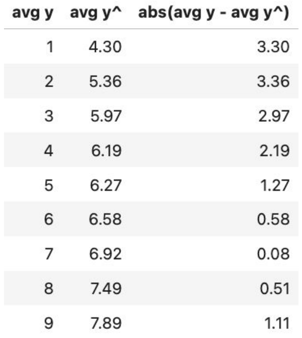
```
training mse: 1.9221582026498665
validation mse: 1.9628908160970593
R^2: 0.3640203183831 2465
```

K-Fold Cross Validation (k = 10):

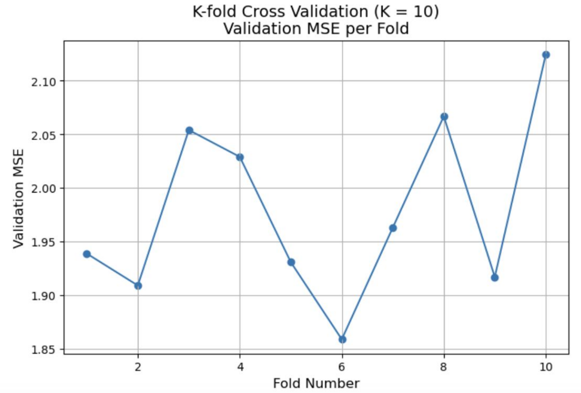


```
Average loss: 1.9793254702494019
```



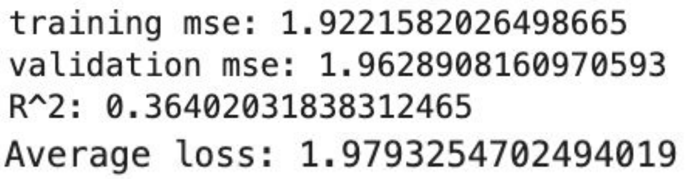


avg y	avg y^	abs(avg y - avg y^)
1	4.30	3.30
2	5.36	3.36
3	5.97	2.97
4	6.19	2.19
5	6.27	1.27
6	6.58	0.58
7	6.92	0.08
8	7.49	0.51
9	7.89	1.11

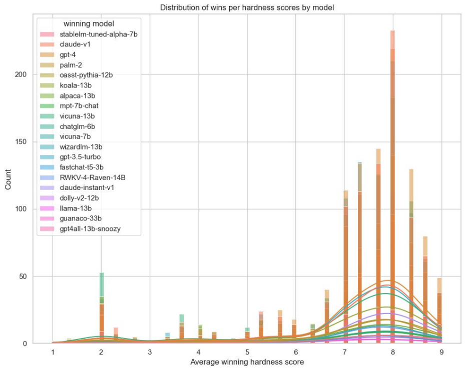


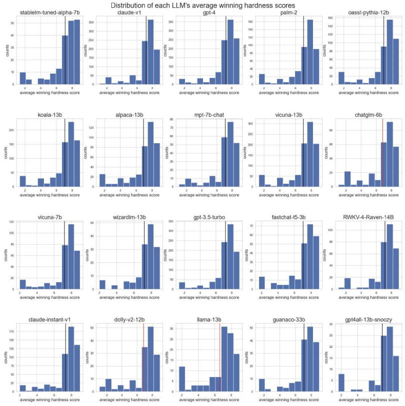
Task B - Interpreting hardness score + Ethical concerns

Evaluation Metric of third model:

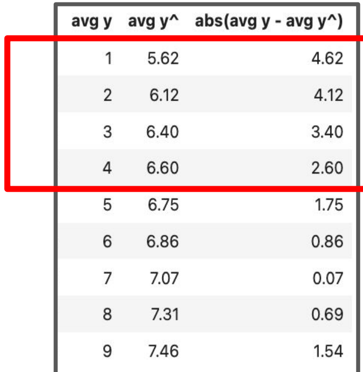


```
training mse: 1.9221582026498665
validation mse: 1.9628908160970593
R^2: 0.36402031838312465
Average loss: 1.9793254702494019
```



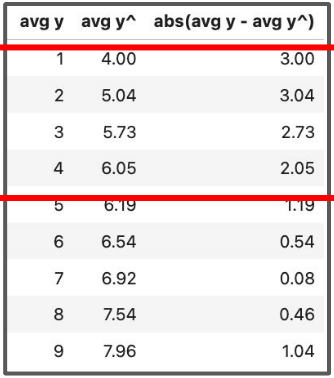


Model 1



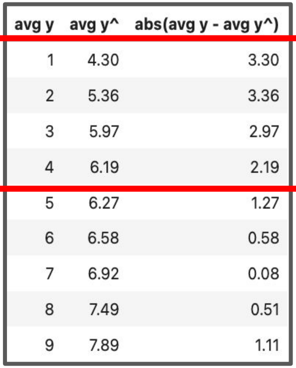
avg y	avg y^	abs(avg y - avg y^)
1	5.62	4.62
2	6.12	4.12
3	6.40	3.40
4	6.60	2.60
5	6.75	1.75
6	6.86	0.86
7	7.07	0.07
8	7.31	0.69
9	7.46	1.54

Model 2



avg y	avg y^	abs(avg y - avg y^)
1	4.00	3.00
2	5.04	3.04
3	5.73	2.73
4	6.05	2.05
5	6.19	1.19
6	6.54	0.54
7	6.92	0.08
8	7.54	0.46
9	7.96	1.04

Model 3



avg y	avg y^	abs(avg y - avg y^)
1	4.30	3.30
2	5.36	3.36
3	5.97	2.97
4	6.19	2.19
5	6.27	1.27
6	6.58	0.58
7	6.92	0.08
8	7.49	0.51
9	7.89	1.11