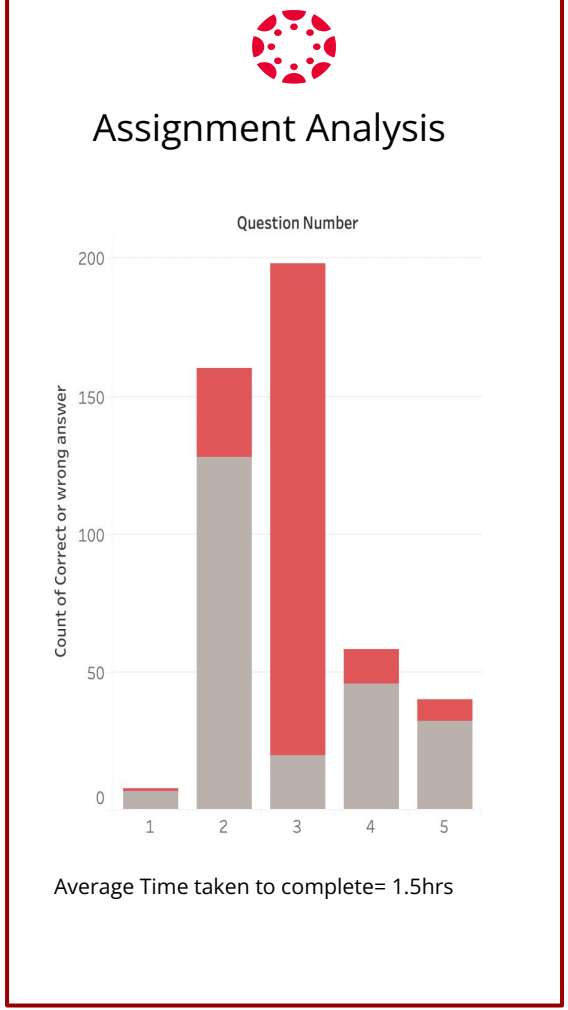
Learning Analytics Hackathon

Team Members:
Amanda Zheng
Hongyi Hu
Jie Chen
Luke Yang
Sonal Bimbra

Assignment Analysis for Instructors

- Highlights concepts to concentrate more on.
- Identify concepts that might be hard for students to understand.



TA appointments for Students

- Feature to 10-15 mins book time slots with TA.
- Save time.
- Enables TAs to plan their time.
- Lesser waiting time for students.
- Walk-ins also allowed

Appointments with TA:

TA Name:
Course name:
Instructor Name:

- Slot1
- Slot1
- Slot3
- Slot4

Confirm Booking

Know your Peers!

- No need to open Roster pdf everytime
- Easily accessible
- Easy search option

Click here to view MBAN Roster

Search here by name:

	Name	Section	Role
	Tolu Adegbehingbe	Learning Analytics Hackathon (Fall 2022)	Student
	Akash Adhikary	Learning Analytics Hackathon (Fall 2022)	Student
	Shivam Aggarwal	Learning Analytics Hackathon (Fall 2022)	Student
	Omair Ahmed	Learning Analytics Hackathon (Fall 2022)	Student
	Ty Andrews	Learning Analytics Hackathon (Fall 2022)	Student
	Gagan Bhatia	Learning Analytics Hackathon (Fall 2022)	Student
	Sonal BIMBRA	Learning Analytics Hackathon (Fall 2022)	Student
	Aneesh Bulusu	Learning Analytics Hackathon (Fall 2022)	Student

Selection of name opens the students details from the Roster!

Target: What contributes to a high score?

- Data Wrangling

```
canvas_df.head()
```

✓ 0.7s Python

Student	Participation & engagement Current Score	Current Score	count_of_likes	total_activity_time	discussion_topic_message_length	post_message_length	answer_count
LEARNER_1	90.0	80.9	0.214286	985351.0	12207	33617	14
LEARNER_10	85.0	78.4	0.298246	421763.0	48509	60378	57
LEARNER_11	90.0	84.3	0.156250	285214.0	31952	40861	32
LEARNER_12	80.0	81.2	0.200000	176035.0	19072	22838	20
LEARNER_13	80.0	85.5	0.219512	774430.0	36887	44122	41

Baseline Model: DummyRegressor()

```
dummy = DummyRegressor()
dummy.fit(X_train, y_train)
dummy.score(X_test, y_test)
```

✓ 0.8s Python

```
-0.04886244503632664
```

```
pd.DataFrame(dummy.predict(X_test), y_test, columns = ['prediction'])
```

✓ 0.4s Python

Current Score	prediction
65.7	81.626923
70.9	81.626923
84.4	81.626923
88.0	81.626923
78.3	81.626923
92.2	81.626923
78.5	81.626923

Test Model: Support Vector Machines (SVM RBF)

```
svr = SVR()
scaler = StandardScaler()
pipe = make_pipeline(scaler, svr)
pipe.fit(X_train, y_train)
pipe.score(X_test,y_test)
```

✓ 0.6s

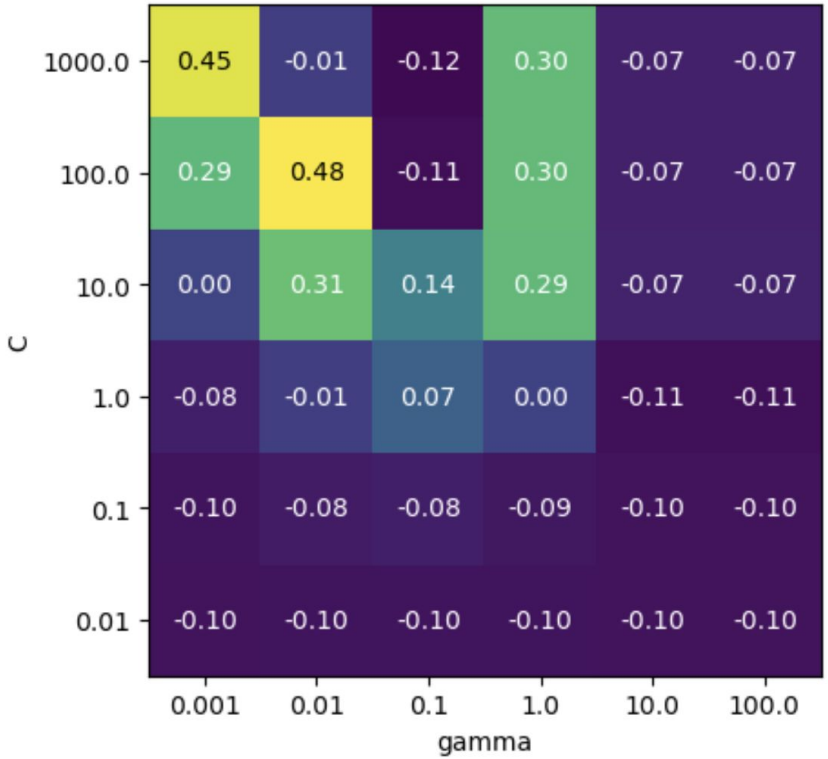
```
0.3435526800484907
```

```
pd.DataFrame(np.array([pipe.predict(X_test),
                       y_test]).T,
                       columns = ['prediction', 'target'])
```

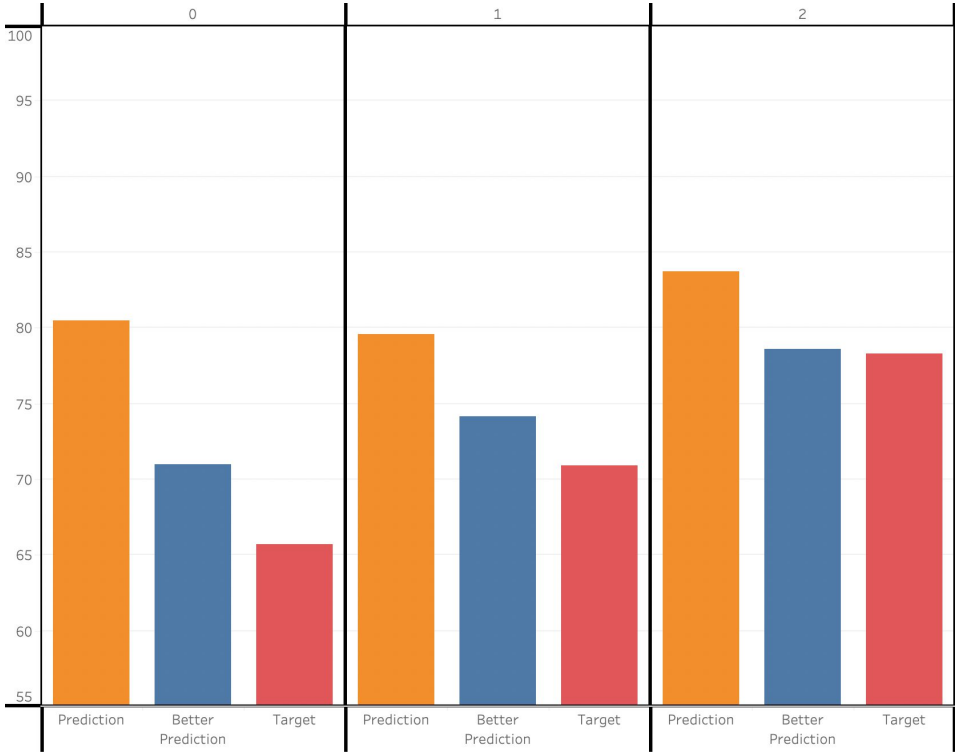
✓ 0.2s

	prediction	target
0	80.442970	65.7
1	79.547524	70.9
2	81.707317	84.4
3	83.249288	88.0
4	83.732376	78.3
5	82.895178	92.2
6	80.184186	78.5

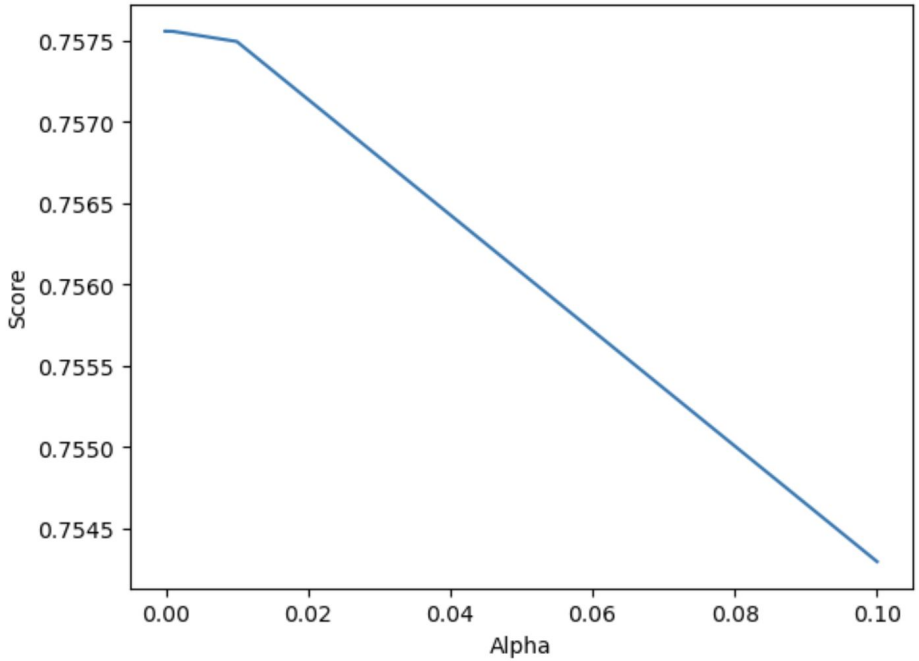
Hyperparameter Tuning (SVM RBF)



Hyperparameter Tuning (SVM RBF)



Test Model: Ridge (Linear Regression)

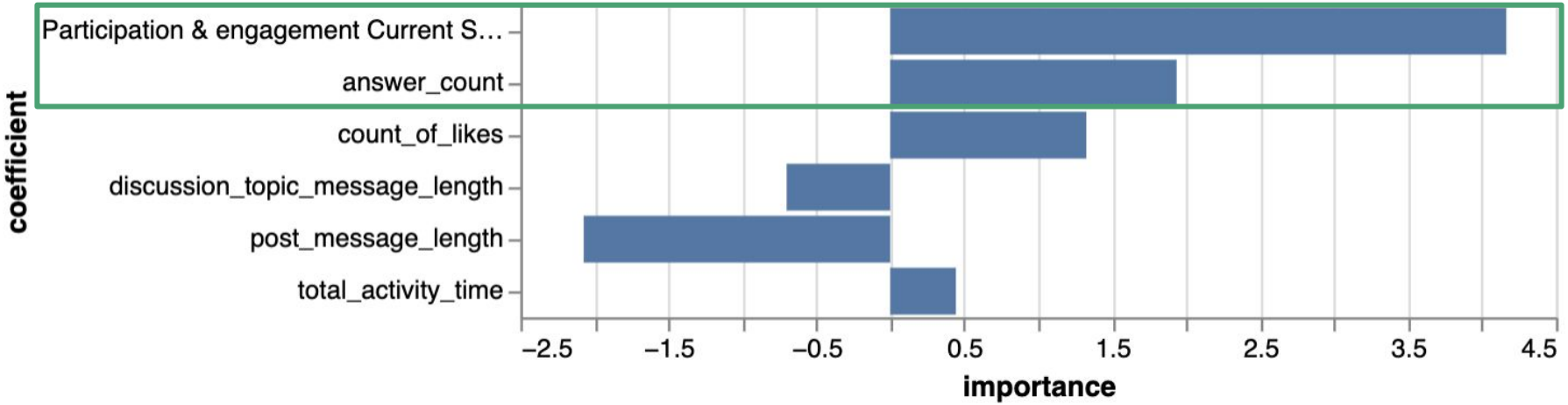



```
ridge = Ridge(alpha=1e-50)

pipe = make_pipeline(StandardScaler(), ridge)
pipe.fit(X_train, y_train)

pd.DataFrame(np.array([pipe.predict(X_test),
                       y_test]).T,
                       columns = ['prediction', 'target'])

pipe.score(X_train, y_train)
```

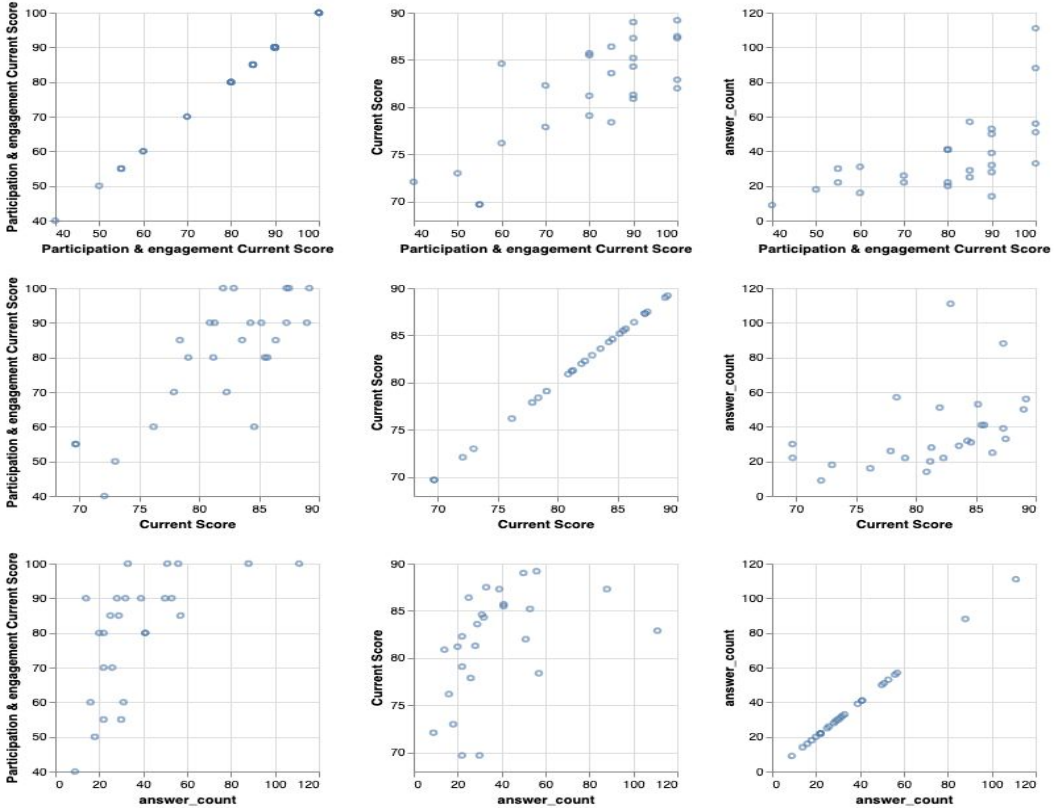
[37] ✓ 0.1s

```
0.7575583513308782
```

Feature Importance

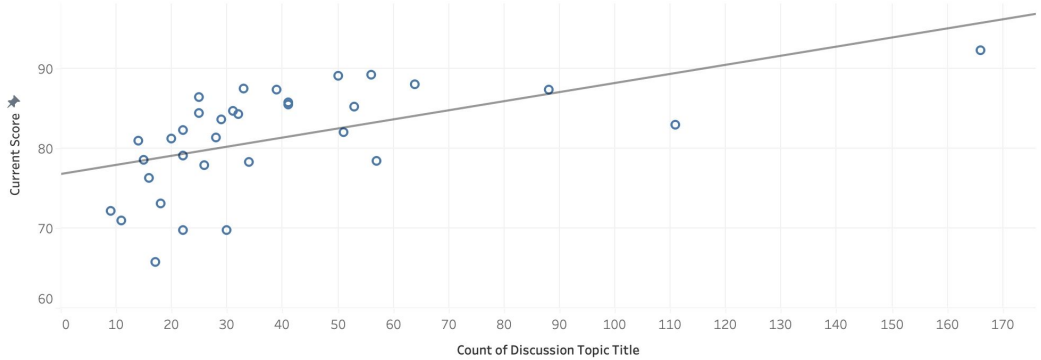


Factors might Affect Your Scores

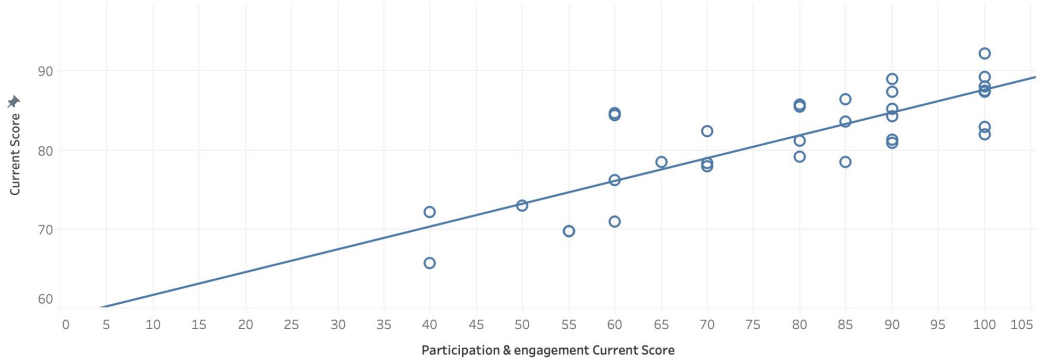


Factors might Affect Your Scores

Total Replies & Scores

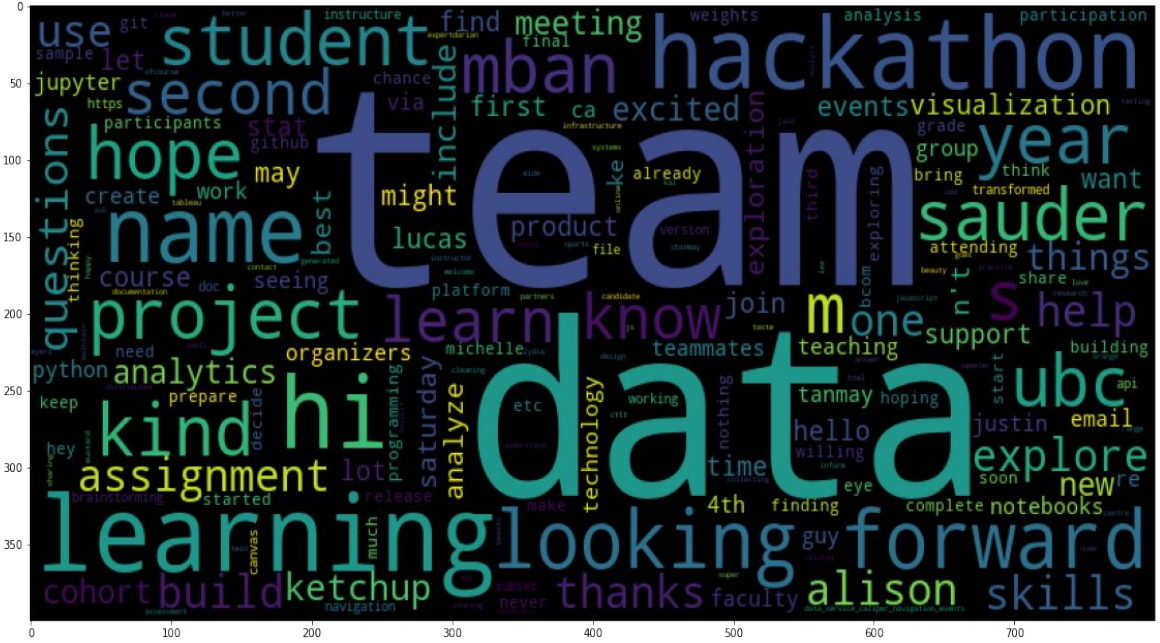


Participation score & Scores



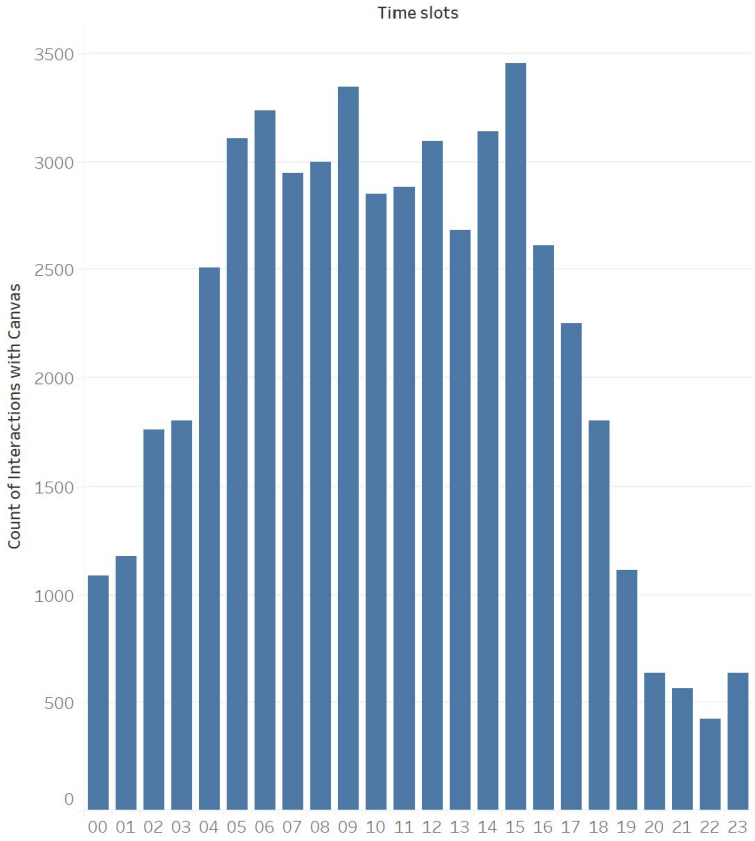
Discussion Board Word Cloud

- Identify the most talked about topics (related to # likes)
- Highlight any challenges faced by students



Maintenance Suggestion

- Maintain Canvas between 20 to 23



Thank you! Any Question?