CS410 – Course Final Project

Project Progress Report

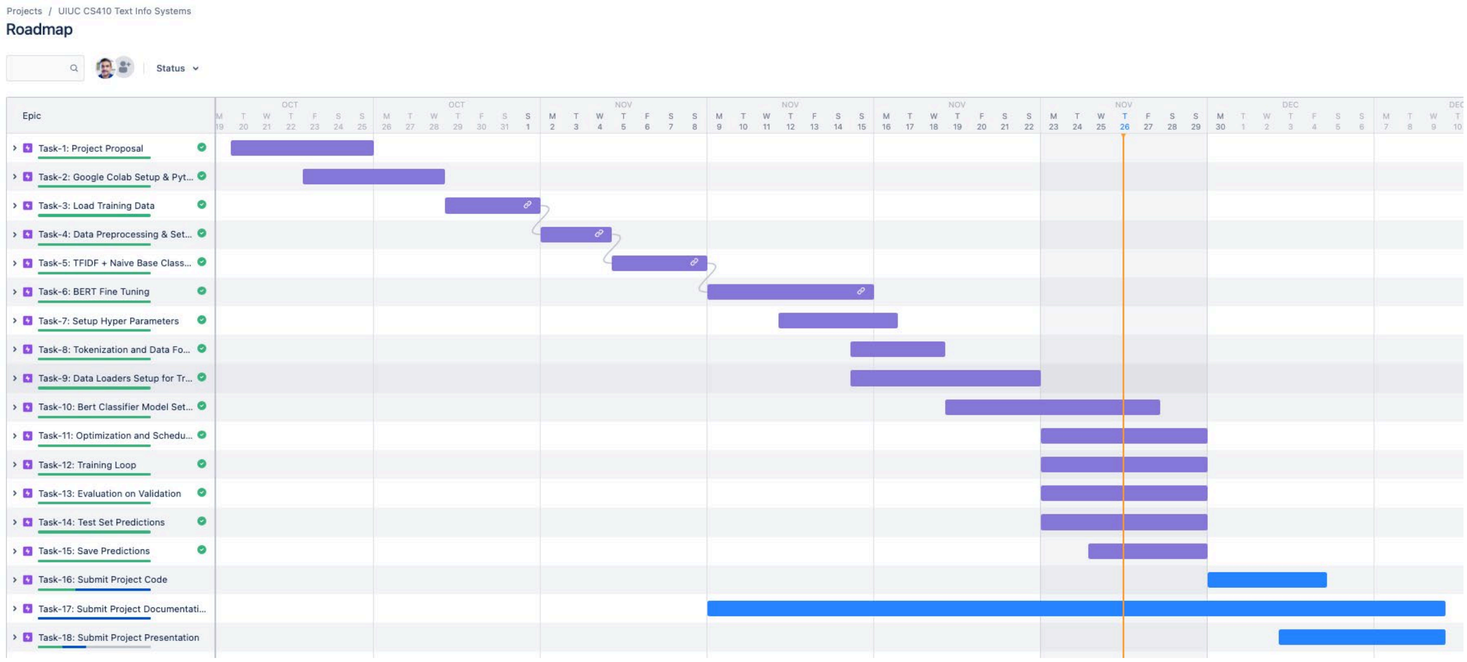
Sembian2@illinois.edu

Project Option 4: Competitions – Text Classification Competition

Team Name: Sembian2 (Individual)

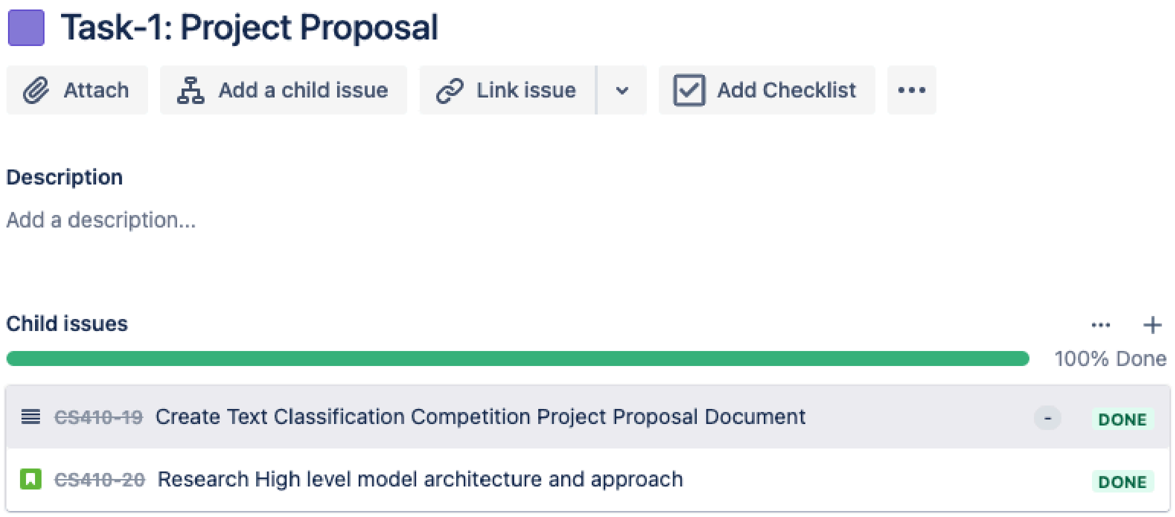
Which tasks have been completed?

The Final project was planned and split into 18 Tasks and over six 2 week sprints, below are the tasks and child issues completed with Sprint reports



Epic

- Task-1: Project Proposal
- Task-2: Google Colab Setup & Pyt...
- Task-3: Load Training Data
- Task-4: Data Preprocessing & Set...
- Task-5: TFIDF + Naive Base Class...
- Task-6: BERT Fine Tuning
- Task-7: Setup Hyper Parameters
- Task-8: Tokenization and Data Fo...
- Task-9: Data Loaders Setup for Tr...
- Task-10: Bert Classifier Model Set...
- Task-11: Optimization and Schedu...
- Task-12: Training Loop
- Task-13: Evaluation on Validation
- Task-14: Test Set Predictions
- Task-15: Save Predictions
- Task-16: Submit Project Code
- Task-17: Submit Project Documentati...
- Task-18: Submit Project Presentation



Task-1: Project Proposal

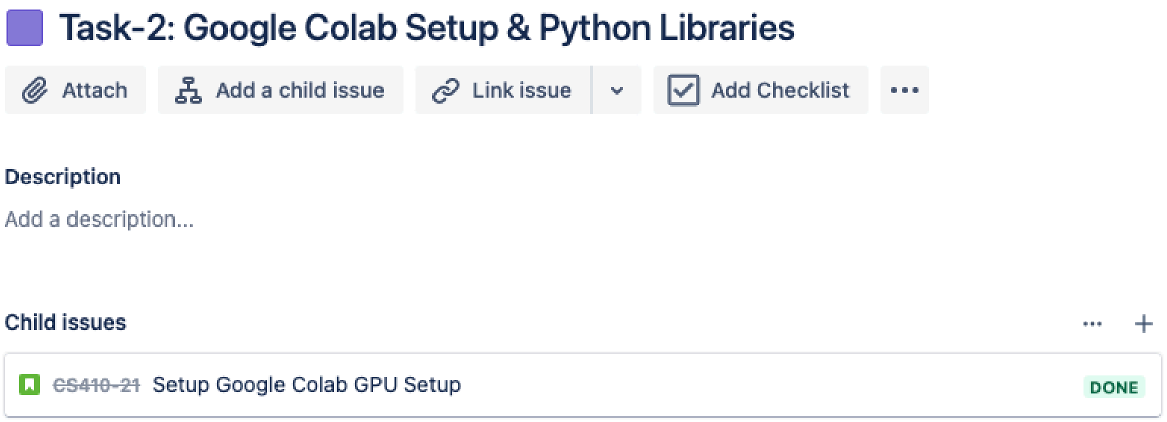
Description

Add a description...

Child issues

100% Done

~~CS410-19~~	Create Text Classification Competition Project Proposal Document	DONE
~~CS410-20~~	Research High level model architecture and approach	DONE



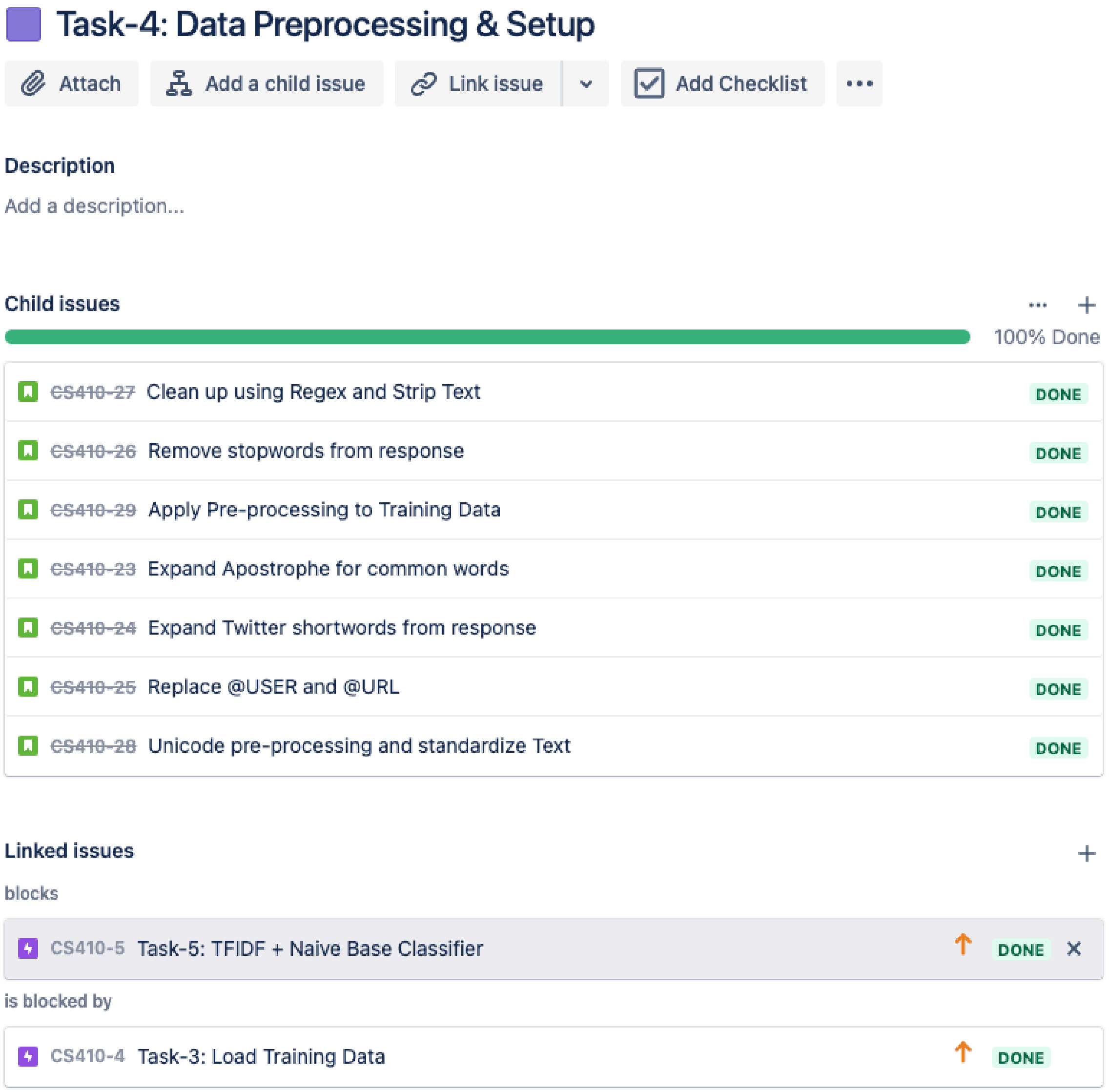
Task-2: Google Colab Setup & Python Libraries

Description

Add a description...

Child issues

- ~~CS410-21~~ Setup Google Colab GPU Setup — DONE



Task-4: Data Preprocessing & Setup

Description

Add a description...

Child issues

100% Done

- ~~CS410-27~~ Clean up using Regex and Strip Text — DONE
- ~~CS410-26~~ Remove stopwords from response — DONE
- ~~CS410-29~~ Apply Pre-processing to Training Data — DONE
- ~~CS410-23~~ Expand Apostrophe for common words — DONE
- ~~CS410-24~~ Expand Twitter shortwords from response — DONE
- ~~CS410-25~~ Replace @USER and @URL — DONE
- ~~CS410-28~~ Unicode pre-processing and standardize Text — DONE

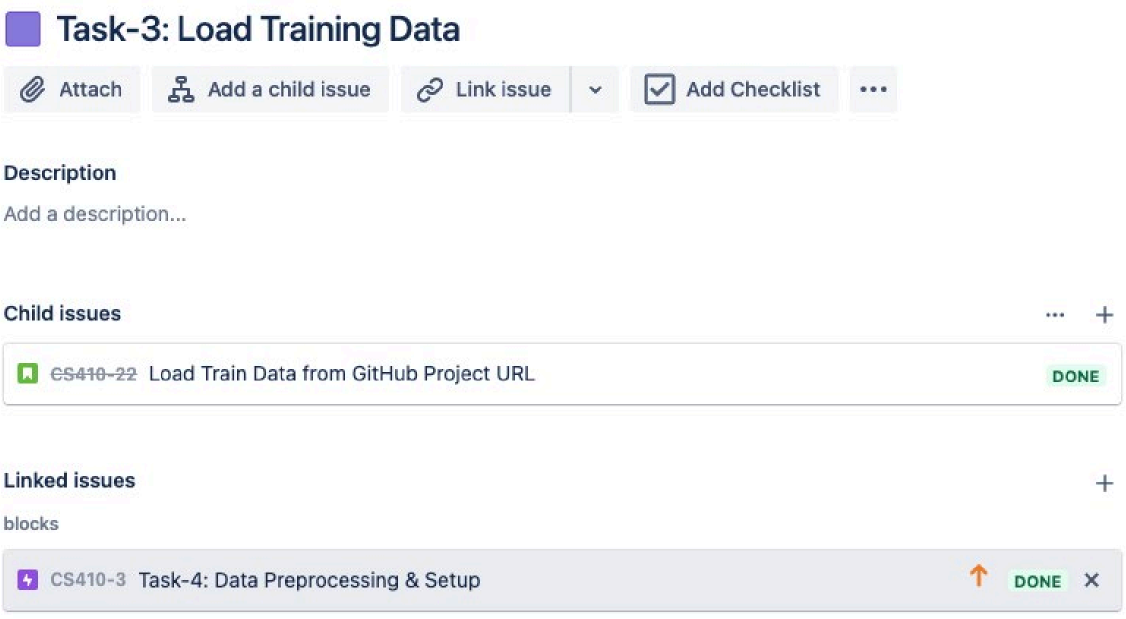
Linked issues

blocks

- CS410-5 Task-5: TFIDF + Naive Base Classifier — DONE

is blocked by

- CS410-4 Task-3: Load Training Data — DONE



Task-3: Load Training Data

Description

Add a description...

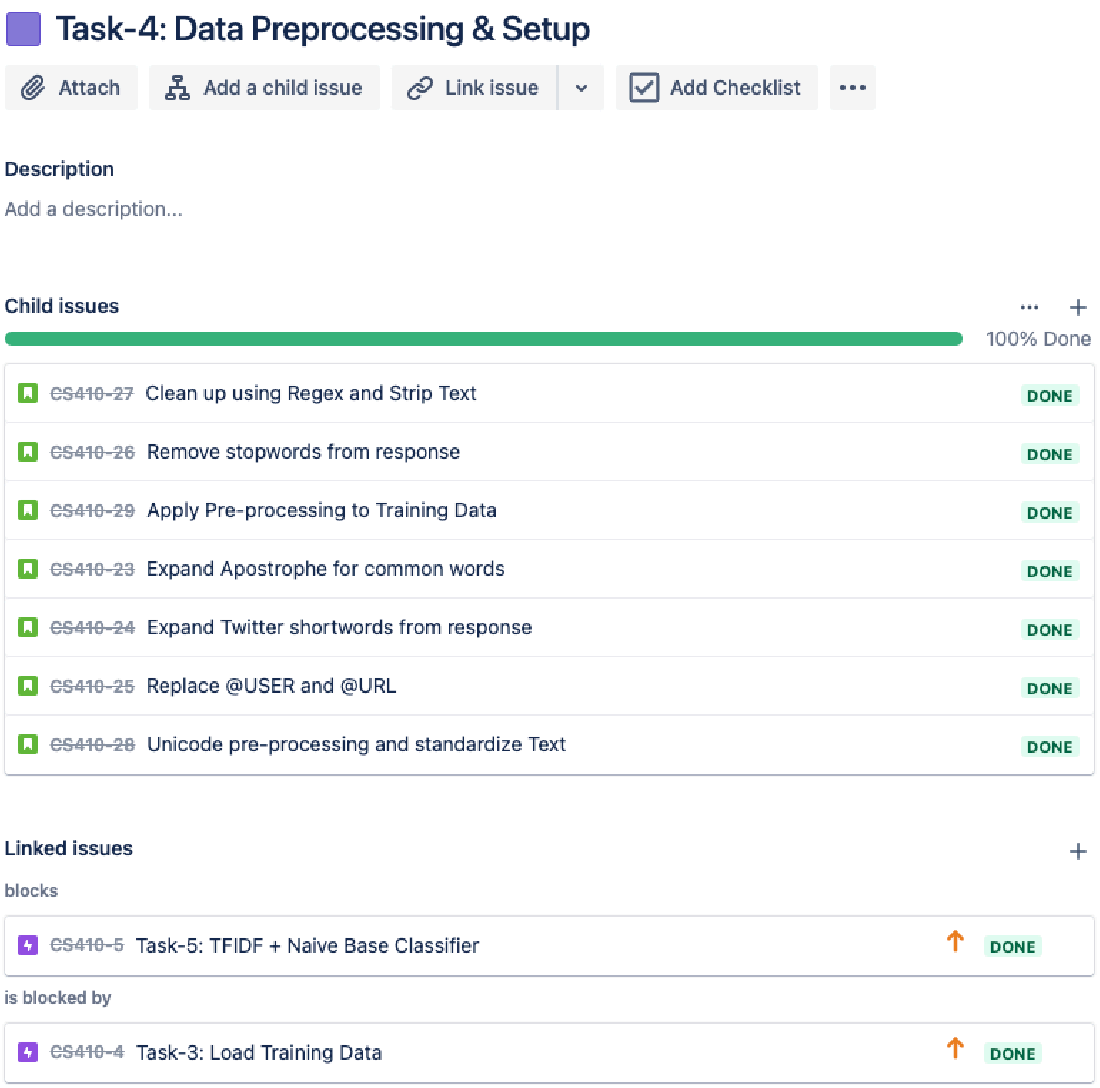
Child issues

- ~~CS410-22~~ Load Train Data from GitHub Project URL DONE

Linked issues

blocks

- CS410-3 Task-4: Data Preprocessing & Setup ↑ DONE ✕



Task-4: Data Preprocessing & Setup

Description

Add a description...

Child issues

100% Done

- ~~CS410-27~~ Clean up using Regex and Strip Text DONE
- ~~CS410-26~~ Remove stopwords from response DONE
- ~~CS410-29~~ Apply Pre-processing to Training Data DONE
- ~~CS410-23~~ Expand Apostrophe for common words DONE
- ~~CS410-24~~ Expand Twitter shortwords from response DONE
- ~~CS410-25~~ Replace @USER and @URL DONE
- ~~CS410-28~~ Unicode pre-processing and standardize Text DONE

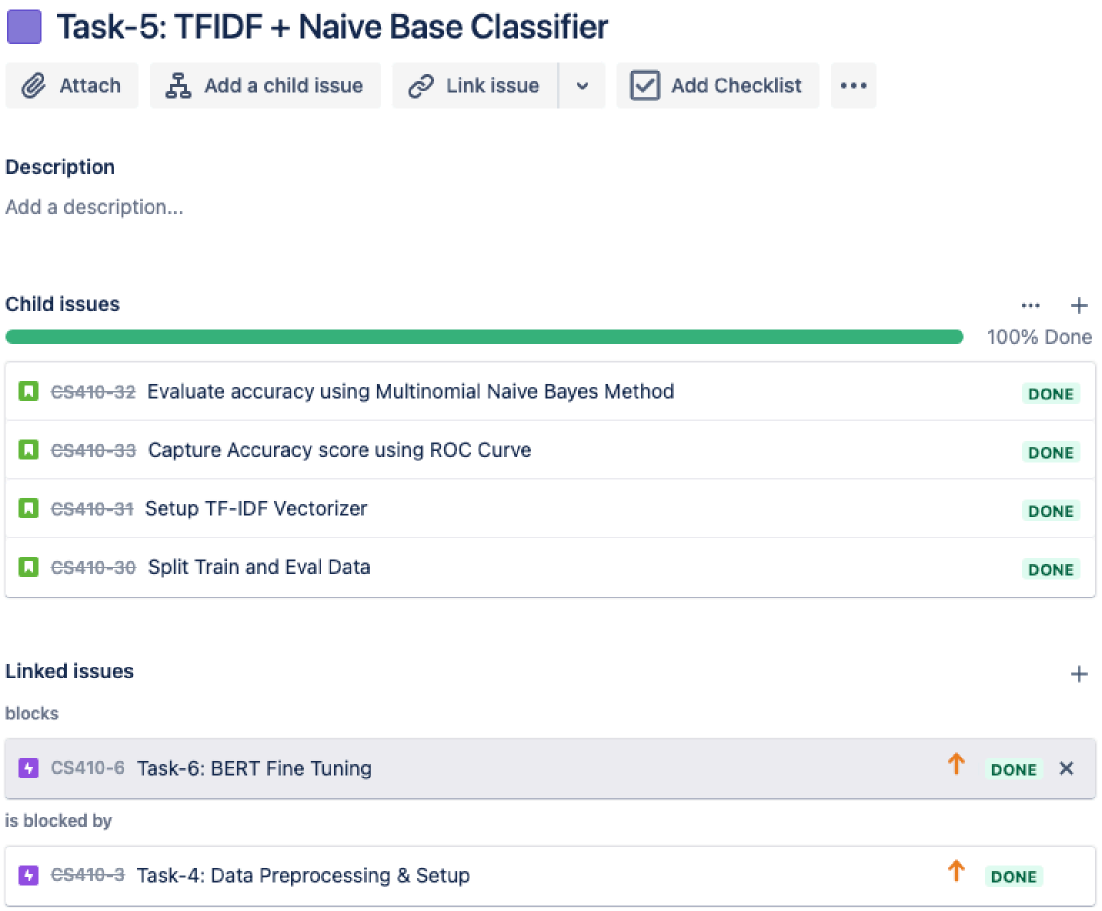
Linked issues

blocks

- ~~CS410-5~~ Task-5: TFIDF + Naive Base Classifier ↑ DONE

is blocked by

- ~~CS410-4~~ Task-3: Load Training Data ↑ DONE



Task-5: TFIDF + Naive Base Classifier

Description

Add a description...

Child issues

100% Done

- ~~CS410-32~~ Evaluate accuracy using Multinomial Naive Bayes Method DONE
- ~~CS410-33~~ Capture Accuracy score using ROC Curve DONE
- ~~CS410-31~~ Setup TF-IDF Vectorizer DONE
- ~~CS410-30~~ Split Train and Eval Data DONE

Linked issues

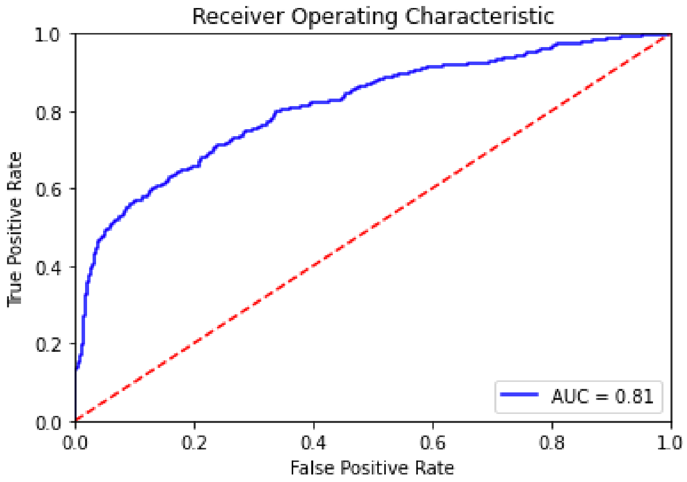
blocks

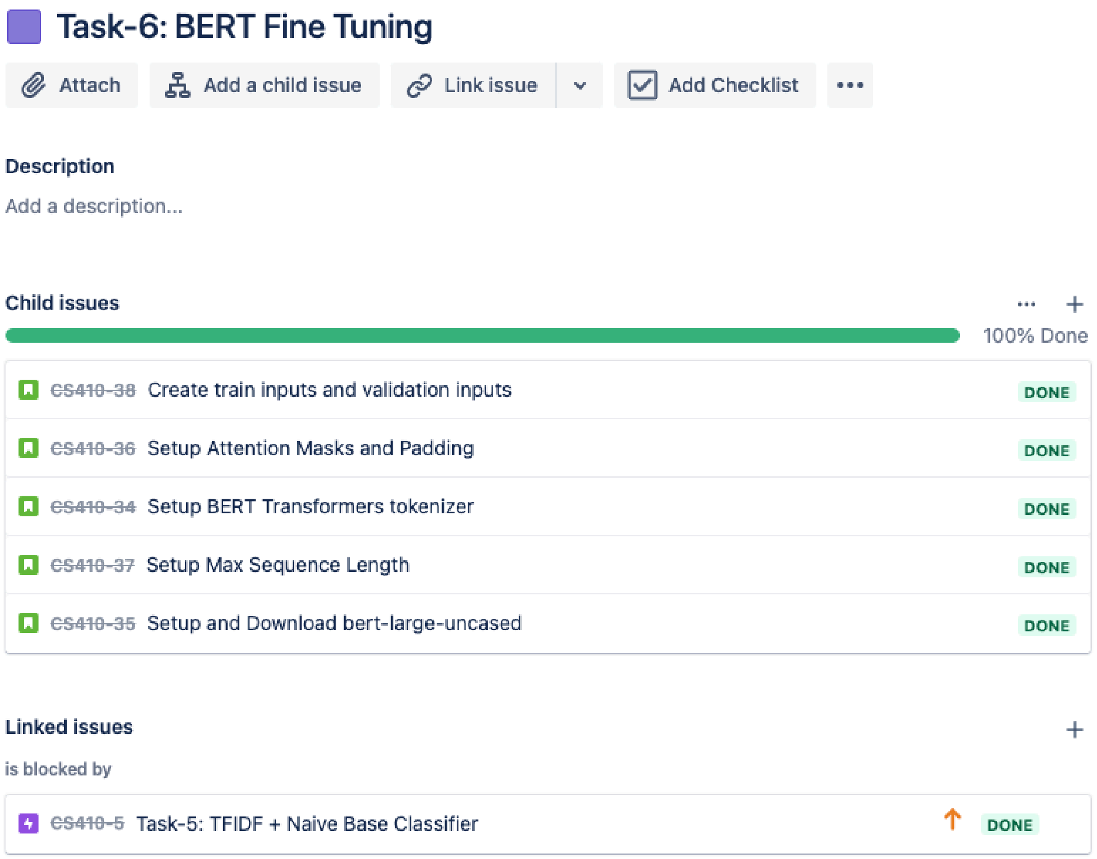
- CS410-6 Task-6: BERT Fine Tuning ↑ DONE ✕

is blocked by

- ~~CS410-3~~ Task-4: Data Preprocessing & Setup ↑ DONE

```
AUC: 0.8118
Accuracy: 72.24%
```





Task-6: BERT Fine Tuning

Description

Add a description...

Child issues

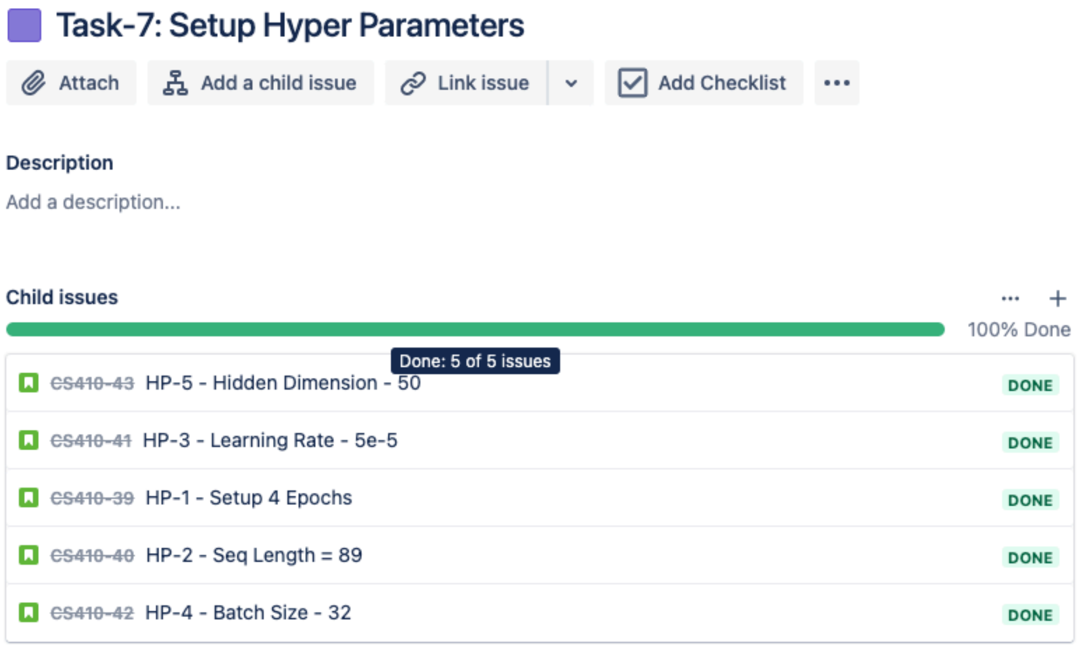
100% Done

- ~~CS410-38~~ Create train inputs and validation inputs DONE
- ~~CS410-36~~ Setup Attention Masks and Padding DONE
- ~~CS410-34~~ Setup BERT Transformers tokenizer DONE
- ~~CS410-37~~ Setup Max Sequence Length DONE
- ~~CS410-35~~ Setup and Download bert-large-uncased DONE

Linked issues

is blocked by

- ~~CS410-5~~ Task-5: TFIDF + Naive Base Classifier ↑ DONE



Task-7: Setup Hyper Parameters

Description

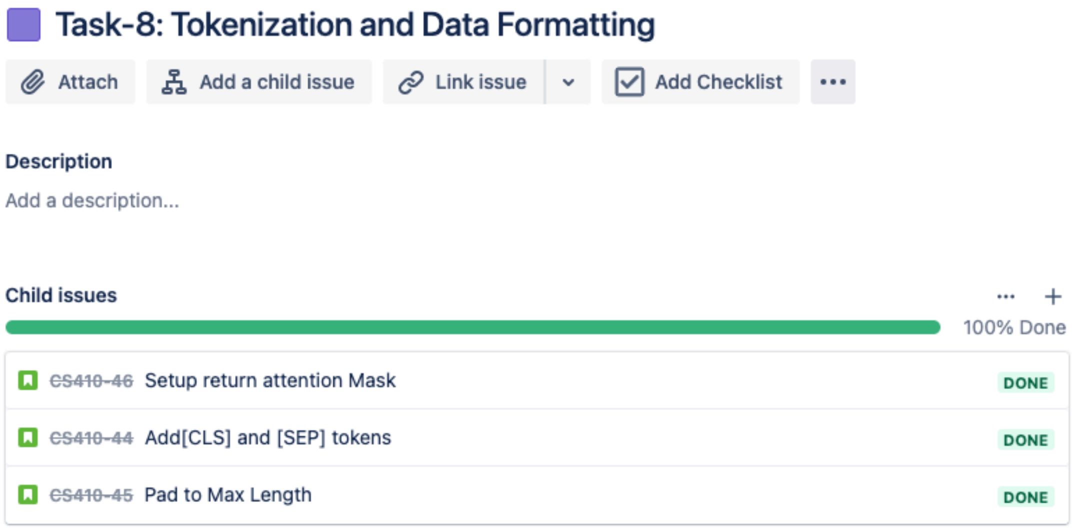
Add a description...

Child issues

100% Done

Done: 5 of 5 issues

- ~~CS410-43~~ HP-5 - Hidden Dimension - 50 DONE
- ~~CS410-41~~ HP-3 - Learning Rate - 5e-5 DONE
- ~~CS410-39~~ HP-1 - Setup 4 Epochs DONE
- ~~CS410-40~~ HP-2 - Seq Length = 89 DONE
- ~~CS410-42~~ HP-4 - Batch Size - 32 DONE



Task-8: Tokenization and Data Formatting

Description

Add a description...

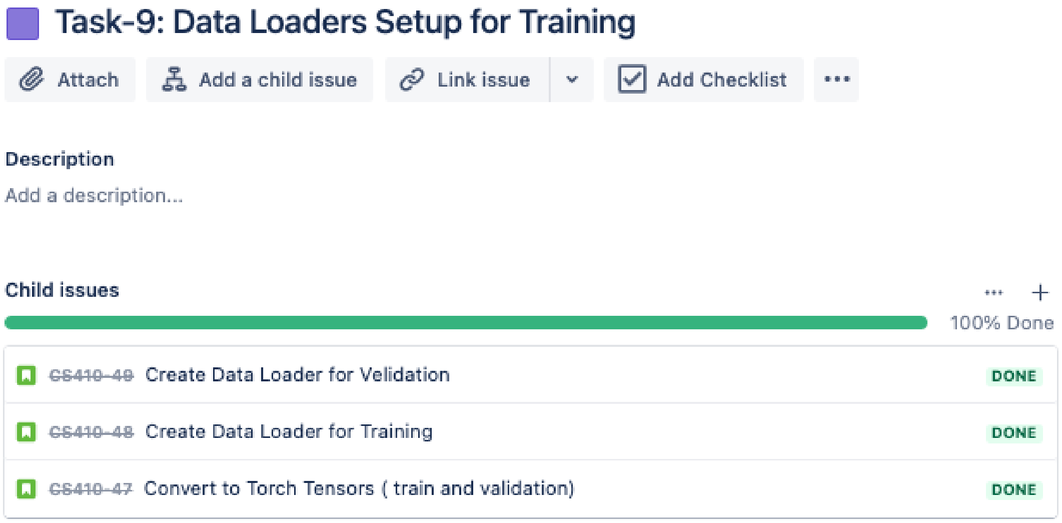
Child issues

100% Done

~~CS410-46~~ Setup return attention Mask DONE

~~CS410-44~~ Add[CLS] and [SEP] tokens DONE

~~CS410-45~~ Pad to Max Length DONE



Task-9: Data Loaders Setup for Training

Description

Add a description...

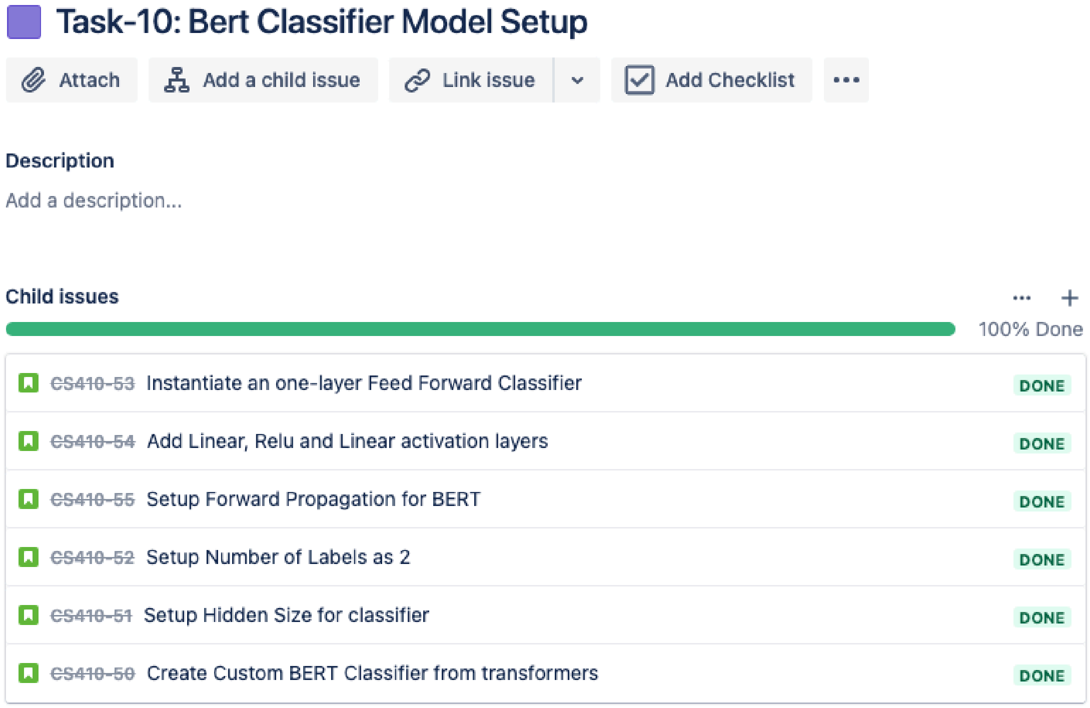
Child issues

100% Done

~~CS410-49~~ Create Data Loader for Velidation DONE

~~CS410-48~~ Create Data Loader for Training DONE

~~CS410-47~~ Convert to Torch Tensors (train and validation) DONE



Task-10: Bert Classifier Model Setup

Description

Add a description...

Child issues

100% Done

~~CS410-53~~ Instantiate an one-layer Feed Forward Classifier DONE

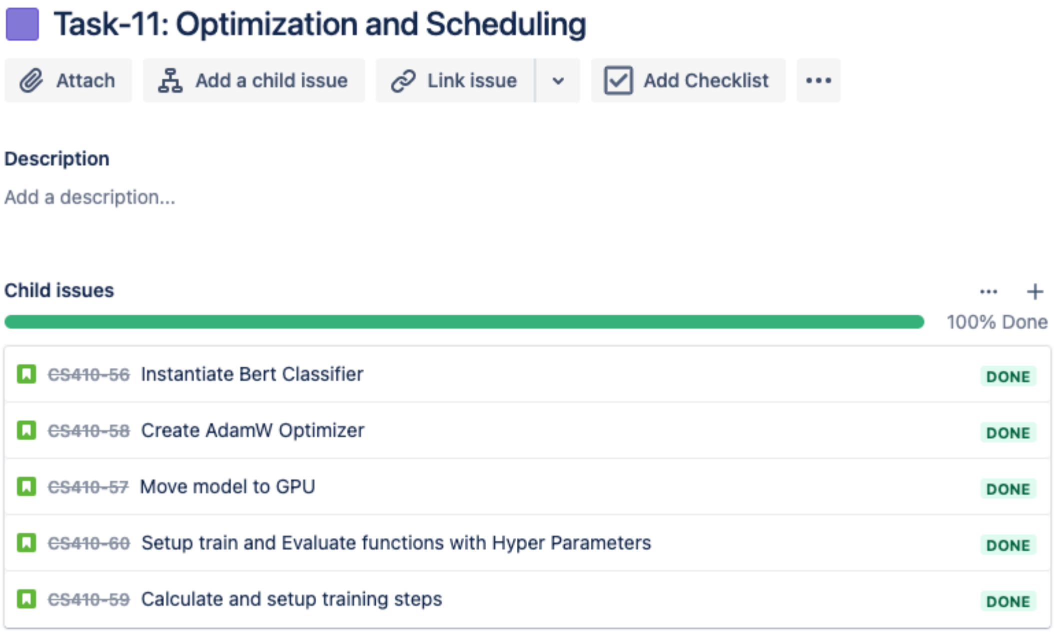
~~CS410-54~~ Add Linear, Relu and Linear activation layers DONE

~~CS410-55~~ Setup Forward Propagation for BERT DONE

~~CS410-52~~ Setup Number of Labels as 2 DONE

~~CS410-51~~ Setup Hidden Size for classifier DONE

~~CS410-50~~ Create Custom BERT Classifier from transformers DONE



Task-11: Optimization and Scheduling

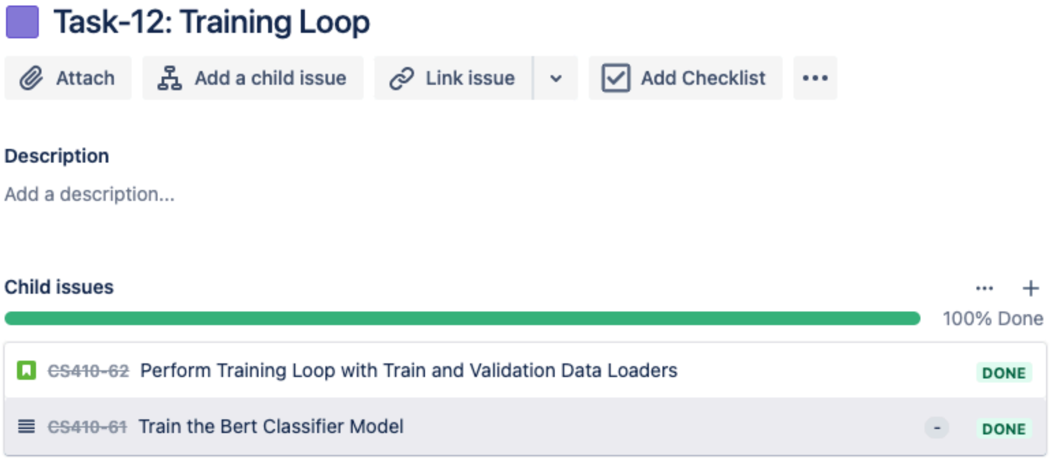
Description

Add a description...

Child issues

100% Done

~~CS410-56~~	Instantiate Bert Classifier	DONE
~~CS410-58~~	Create AdamW Optimizer	DONE
~~CS410-57~~	Move model to GPU	DONE
~~CS410-60~~	Setup train and Evaluate functions with Hyper Parameters	DONE
~~CS410-59~~	Calculate and setup training steps	DONE



Task-12: Training Loop

Description

Add a description...

Child issues

100% Done

~~CS410-62~~	Perform Training Loop with Train and Validation Data Loaders	DONE
~~CS410-61~~	Train the Bert Classifier Model	DONE

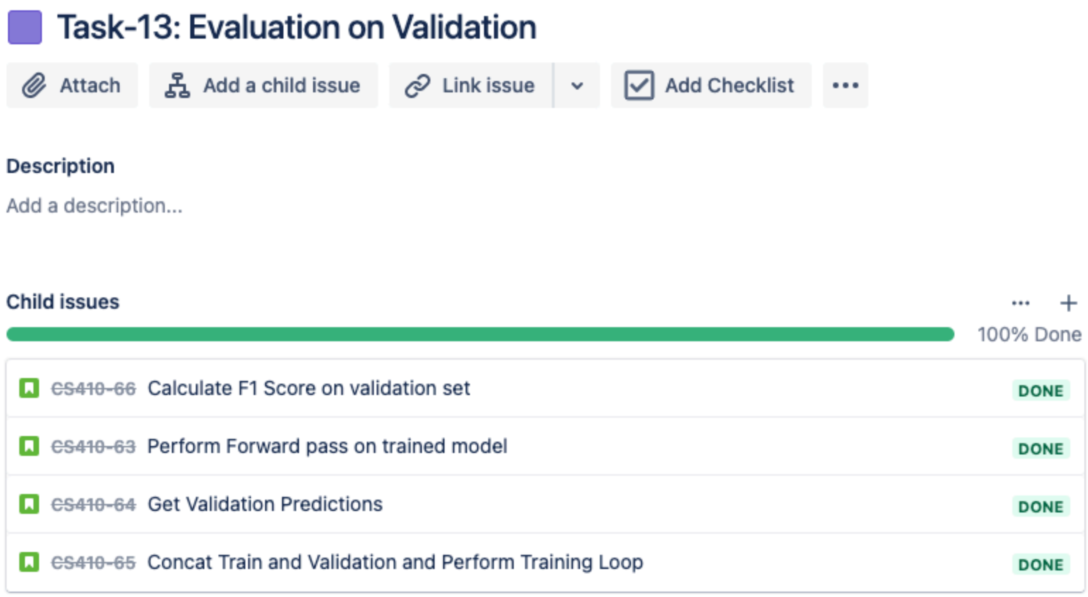
```
Start training...

 Epoch  |  Batch  |  Train Loss  |  Val Loss  |  Val Acc  |  Elapsed
----------------------------------------------------------------------
   1    |   20    |   0.664404   |     -      |     -     |   31.78
   1    |   40    |   0.534449   |     -      |     -     |   30.99
   1    |   60    |   0.533544   |     -      |     -     |   31.70
   1    |   80    |   0.500559   |     -      |     -     |   32.19
   1    |   100   |   0.479444   |     -      |     -     |   32.82
   1    |   104   |   0.415712   |     -      |     -     |   6.14
----------------------------------------------------------------------
   1    |    -    |   0.538812   |  0.444033  |   79.88   |  196.45
----------------------------------------------------------------------


 Epoch  |  Batch  |  Train Loss  |  Val Loss  |  Val Acc  |  Elapsed
----------------------------------------------------------------------
   2    |   20    |   0.291500   |     -      |     -     |   35.13
   2    |   40    |   0.280435   |     -      |     -     |   33.84
   2    |   60    |   0.300033   |     -      |     -     |   34.14
   2    |   80    |   0.294220   |     -      |     -     |   34.19
   2    |   100   |   0.264788   |     -      |     -     |   34.18
   2    |   104   |   0.178849   |     -      |     -     |   6.32
----------------------------------------------------------------------
   2    |    -    |   0.282156   |  0.582648  |   78.42   |  209.20
----------------------------------------------------------------------


 Epoch  |  Batch  |  Train Loss  |  Val Loss  |  Val Acc  |  Elapsed
----------------------------------------------------------------------
   3    |   20    |   0.178021   |     -      |     -     |   35.83
   3    |   40    |   0.084245   |     -      |     -     |   34.24
   3    |   60    |   0.126948   |     -      |     -     |   34.21
   3    |   80    |   0.113072   |     -      |     -     |   34.07
   3    |   100   |   0.097395   |     -      |     -     |   34.11
   3    |   104   |   0.030649   |     -      |     -     |   6.31
----------------------------------------------------------------------
   3    |    -    |   0.117088   |  0.745395  |   79.05   |  210.33
----------------------------------------------------------------------


 Epoch  |  Batch  |  Train Loss  |  Val Loss  |  Val Acc  |  Elapsed
----------------------------------------------------------------------
   4    |   20    |   0.032811   |     -      |     -     |   35.79
   4    |   40    |   0.025885   |     -      |     -     |   34.05
   4    |   60    |   0.056671   |     -      |     -     |   34.21
   4    |   80    |   0.051280   |     -      |     -     |   34.09
   4    |   100   |   0.024765   |     -      |     -     |   34.04
   4    |   104   |   0.079427   |     -      |     -     |   6.33
----------------------------------------------------------------------
   4    |    -    |   0.039798   |  0.846048  |   81.17   |  210.02
----------------------------------------------------------------------
```



Task-13: Evaluation on Validation

Description

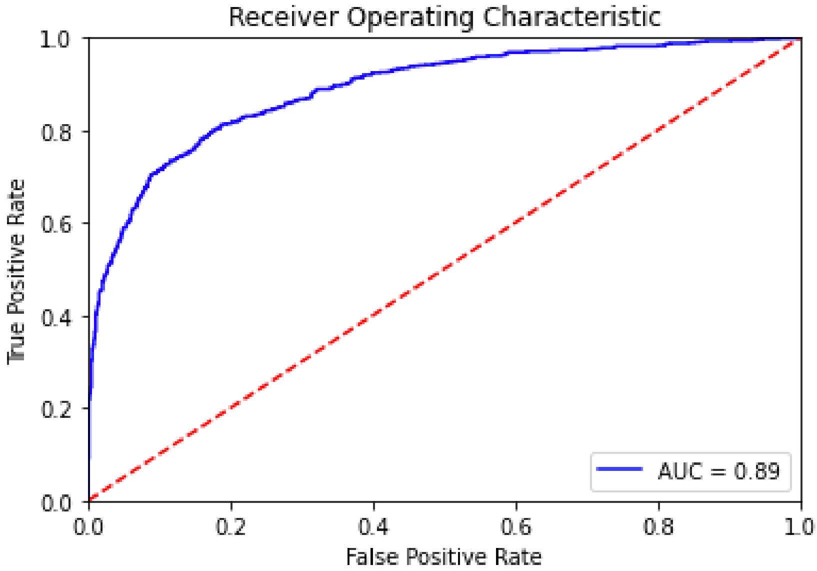
Add a description...

Child issues

100% Done

- ~~CS410-66~~ Calculate F1 Score on validation set DONE
- ~~CS410-63~~ Perform Forward pass on trained model DONE
- ~~CS410-64~~ Get Validation Predictions DONE
- ~~CS410-65~~ Concat Train and Validation and Perform Training Loop DONE

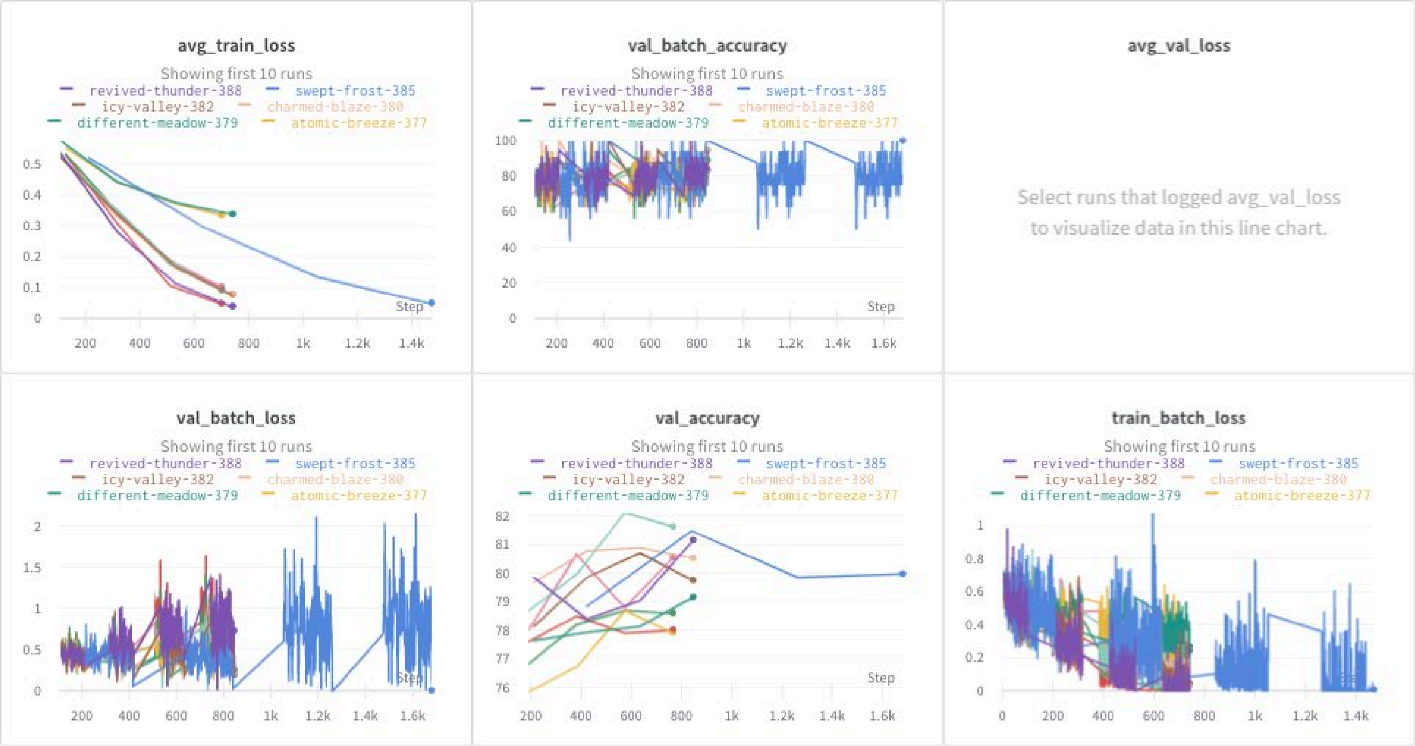
```
AUC: 0.8887
Accuracy: 81.15%
```

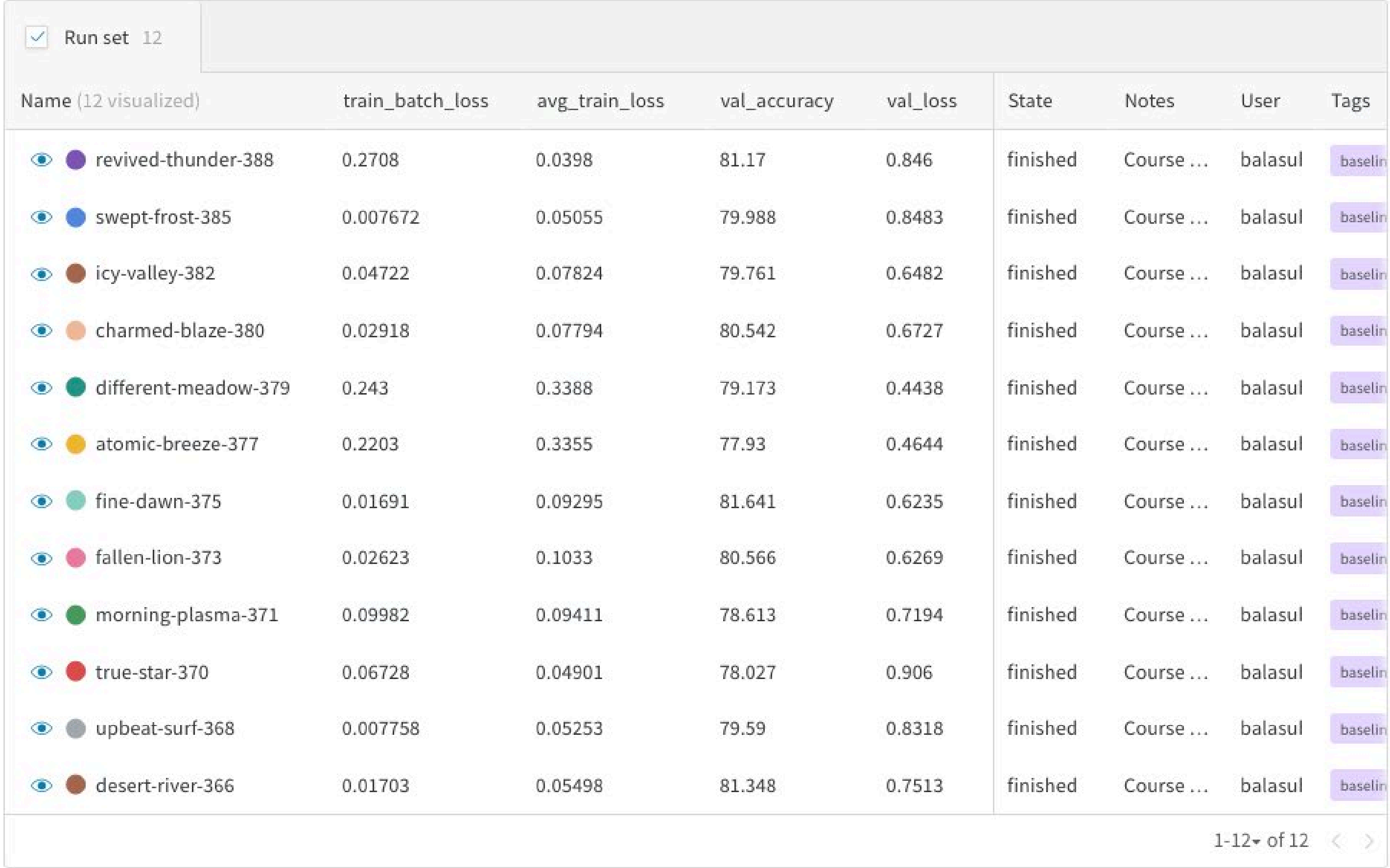


For HyperParameter tuning I used wandb.com (weights and Biases) to report out the various runs and compared the best score and the run named revivedpthunder-388 scored the highest and achieved a 81.17 validation accuracy and 75.19 Test Accuracy in the leaderboard

[Report Link to wandb]

Training and Validation Reports HyperParameter Tuning

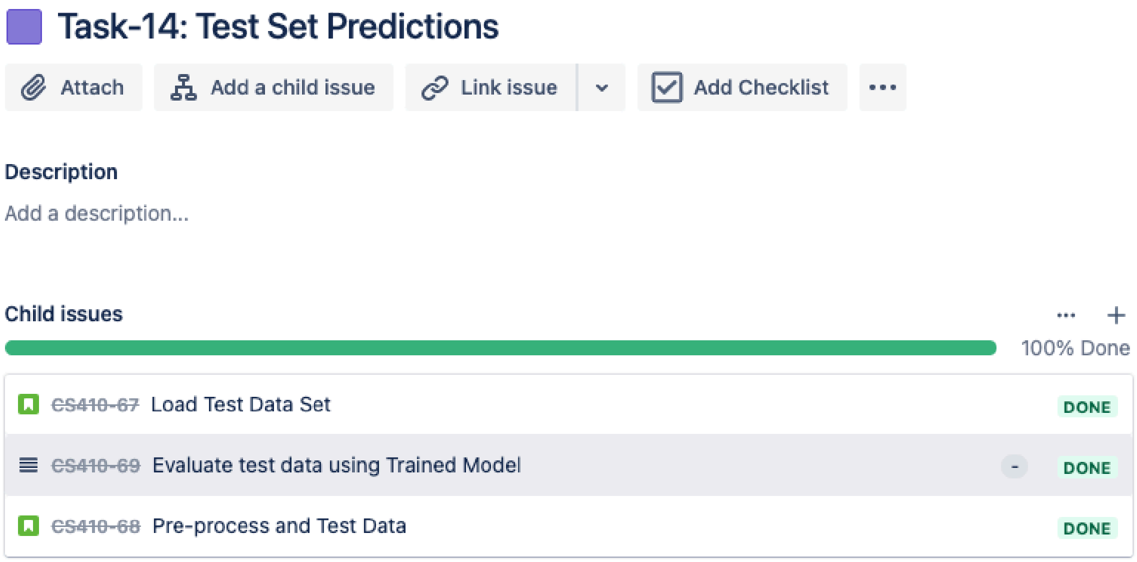




☑ Run set 12

Name (12 visualized)	train_batch_loss	avg_train_loss	val_accuracy	val_loss	State	Notes	User	Tags
● revived-thunder-388	0.2708	0.0398	81.17	0.846	finished	Course ...	balasul	baselin
● swept-frost-385	0.007672	0.05055	79.988	0.8483	finished	Course ...	balasul	baselin
● icy-valley-382	0.04722	0.07824	79.761	0.6482	finished	Course ...	balasul	baselin
● charmed-blaze-380	0.02918	0.07794	80.542	0.6727	finished	Course ...	balasul	baselin
● different-meadow-379	0.243	0.3388	79.173	0.4438	finished	Course ...	balasul	baselin
● atomic-breeze-377	0.2203	0.3355	77.93	0.4644	finished	Course ...	balasul	baselin
● fine-dawn-375	0.01691	0.09295	81.641	0.6235	finished	Course ...	balasul	baselin
● fallen-lion-373	0.02623	0.1033	80.566	0.6269	finished	Course ...	balasul	baselin
● morning-plasma-371	0.09982	0.09411	78.613	0.7194	finished	Course ...	balasul	baselin
● true-star-370	0.06728	0.04901	78.027	0.906	finished	Course ...	balasul	baselin
● upbeat-surf-368	0.007758	0.05253	79.59	0.8318	finished	Course ...	balasul	baselin
● desert-river-366	0.01703	0.05498	81.348	0.7513	finished	Course ...	balasul	baselin

1-12▾ of 12



Task-14: Test Set Predictions

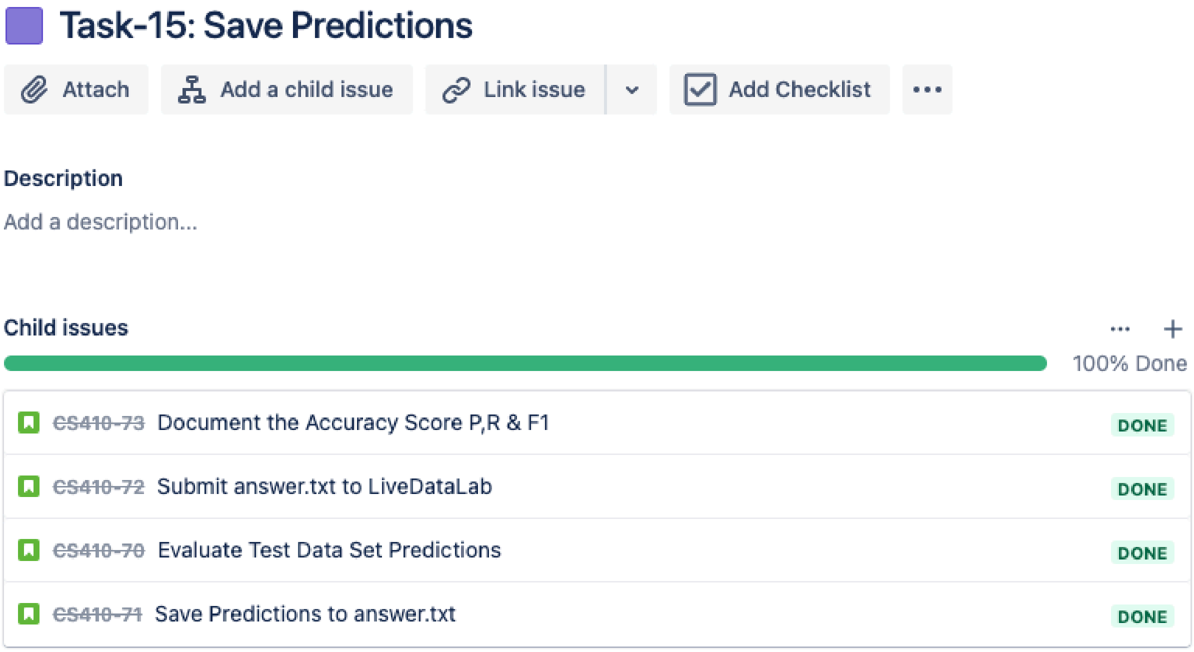
Description

Add a description...

Child issues

100% Done

~~CS410-67~~	Load Test Data Set	DONE
~~CS410-69~~	Evaluate test data using Trained Model	DONE
~~CS410-68~~	Pre-process and Test Data	DONE



Task-15: Save Predictions

Description

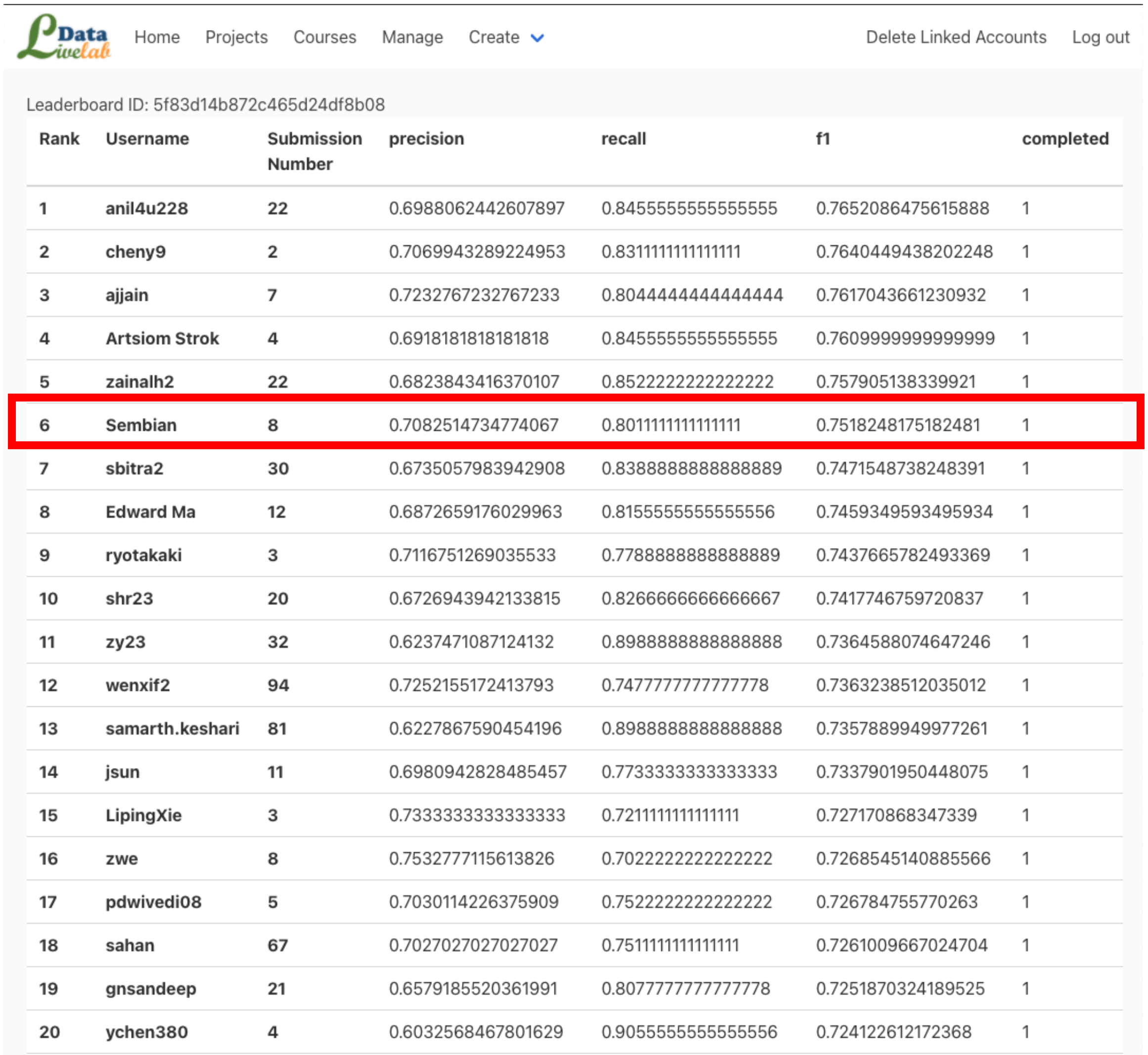
Add a description...

Child issues

100% Done

~~CS410-73~~	Document the Accuracy Score P,R & F1	DONE
~~CS410-72~~	Submit answer.txt to LiveDataLab	DONE
~~CS410-70~~	Evaluate Test Data Set Predictions	DONE
~~CS410-71~~	Save Predictions to answer.txt	DONE

Test Accuracy of 75.19% - Position 6 on Leaderboard as of 11.26.2020

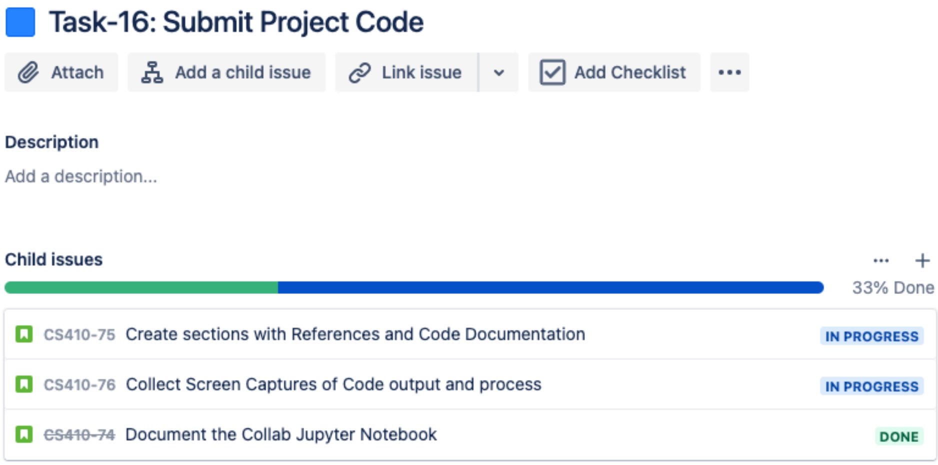


Leaderboard ID: 5f83d14b872c465d24df8b08

Rank	Username	Submission Number	precision	recall	f1	completed
1	anil4u228	22	0.6988062442607897	0.8455555555555555	0.7652086475615888	1
2	cheny9	2	0.7069943289224953	0.8311111111111111	0.7640449438202248	1
3	ajjain	7	0.7232767232767233	0.8044444444444444	0.7617043661230932	1
4	Artsiom Strok	4	0.6918181818181818	0.8455555555555555	0.7609999999999999	1
5	zainalh2	22	0.6823843416370107	0.8522222222222222	0.757905138339921	1
6	Sembian	8	0.7082514734774067	0.8011111111111111	0.7518248175182481	1
7	sbitra2	30	0.6735057983942908	0.8388888888888889	0.7471548738248391	1
8	Edward Ma	12	0.6872659176029963	0.8155555555555556	0.7459349593495934	1
9	ryotakaki	3	0.7116751269035533	0.7788888888888889	0.7437665782493369	1
10	shr23	20	0.6726943942133815	0.8266666666666667	0.7417746759720837	1
11	zy23	32	0.6237471087124132	0.8988888888888888	0.7364588074647246	1
12	wenxif2	94	0.7252155172413793	0.7477777777777778	0.7363238512035012	1
13	samarth.keshari	81	0.6227867590454196	0.8988888888888888	0.7357889949977261	1
14	jsun	11	0.6980942828485457	0.7733333333333333	0.7337901950448075	1
15	LipingXie	3	0.7333333333333333	0.7211111111111111	0.727170868347339	1
16	zwe	8	0.7532777115613826	0.7022222222222222	0.7268545140885566	1
17	pdwivedi08	5	0.7030114226375909	0.7522222222222222	0.726784755770263	1
18	sahan	67	0.7027027027027027	0.7511111111111111	0.7261009667024704	1
19	gnsandeep	21	0.6579185520361991	0.8077777777777778	0.7251870324189525	1
20	ychen380	4	0.6032568467801629	0.9055555555555556	0.724122612172368	1

Which tasks are pending?

The following tasks are in progress and I am on track for completing the project code documentation and presentation.



Task-16: Submit Project Code

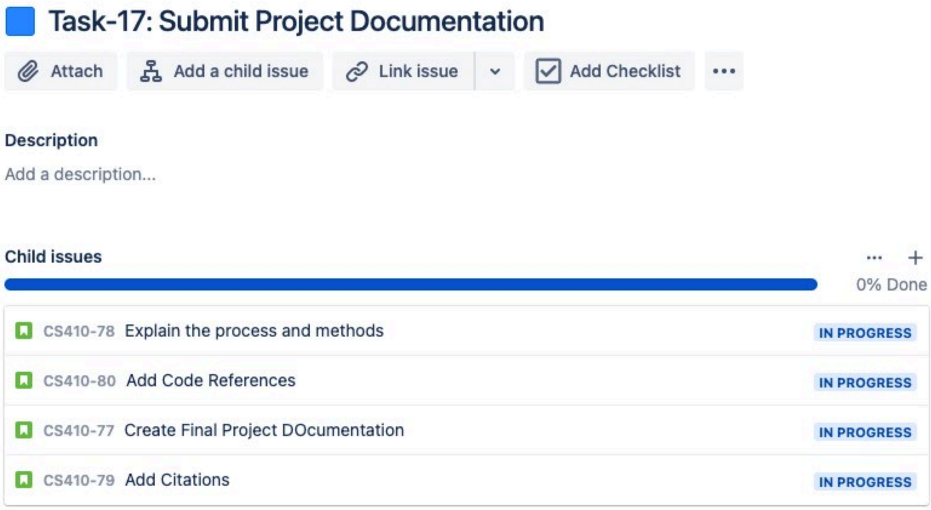
Description

Add a description...

Child issues

33% Done

CS410-75	Create sections with References and Code Documentation	IN PROGRESS
CS410-76	Collect Screen Captures of Code output and process	IN PROGRESS
~~CS410-74~~	Document the Collab Jupyter Notebook	DONE



Task-17: Submit Project Documentation

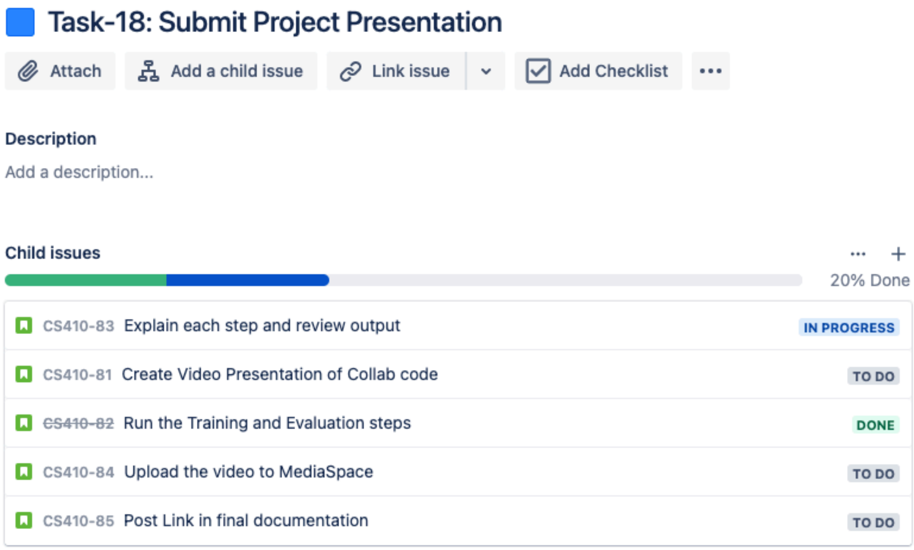
Description

Add a description...

Child issues

0% Done

CS410-78	Explain the process and methods	IN PROGRESS
CS410-80	Add Code References	IN PROGRESS
CS410-77	Create Final Project DOcumentation	IN PROGRESS
CS410-79	Add Citations	IN PROGRESS



Task-18: Submit Project Presentation

Description

Add a description...

Child issues

20% Done

CS410-83	Explain each step and review output	IN PROGRESS
CS410-81	Create Video Presentation of Collab code	TO DO
~~CS410-82~~	Run the Training and Evaluation steps	DONE
CS410-84	Upload the video to MediaSpace	TO DO
CS410-85	Post Link in final documentation	TO DO

Are you facing any challenges?

Using the Transformers, Pytorch and BERT Classification model I was able to beat the baseline score on the leaderboard and improved the score by repeating the training with Hyper Parameter tuning and text pre-processing techniques and achieved a score of 75.19% Test Accuracy, and have no challenges.