CSC401/2511
TUTORIAL 3

Feb 4, 2022

OVERVIEW

- In-depth look at Part 3 (Classification)
  - (last week you saw in-depth look at Parts 1 and 2)
- Breakdown of sample data
- Piazza recap
- Q&A

CLASSIFICATION

- Four parts:
  - Compare classifiers
  - Experiment with the amount of training data used
  - Select the best features for classification
  - Do cross-fold validation

CLASSIFICATION 1: COMPARE CLASSIFIERS

- *Randomly* split data into 80% training, 20% testing.

- We have 5 classification methods, which you can consider to be 'black boxes' (input goes in, classes come out).
  1. Support vector machine with linear kernel
  2. Gaussian naïve Bayes classifier.
  3. Random forest classifier
  4. Neural network
  5. Adaboost (with decision tree)

CLASSIFICATION I: COMPARE CLASSIFIERS

```
Results for SGDClassifier:
        Accuracy: 0.XXXX
        Recall: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
        Precision: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
        Confusion Matrix:
[[ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]]

Results for GaussianNB:
... results for the rest of classifiers

my optional written analysis goes here :)
```

CLASSIFICATION I: COMPARE CLASSIFIERS

```
Results for SGDClassifier:
        Accuracy: 0.XXXX
        Recall: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
        Precision: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
        Confusion Matrix:
[[ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]]

Results for GaussianNB:
... results for the rest of classifiers

my optional written analysis goes here :)
```

Accuracy over all classes

Precision and recall per class (e.g. index 0 corresponds to class Left)

CLASSIFICATION I: COMPARE CLASSIFIERS

```
Results for SGDClassifier:
        Accuracy: 0.XXXX
        Recall: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
        Precision: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
        Confusion Matrix:
[[ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]
 [ XXX  XXX XXX XXX]]

Results for GaussianNB:
... results for the rest of classifiers

my optional written analysis goes here :)
```

Examples the model classified as Left

Examples labeled Left in the training data

CLASSIFICATION 2: AMOUNT OF DATA

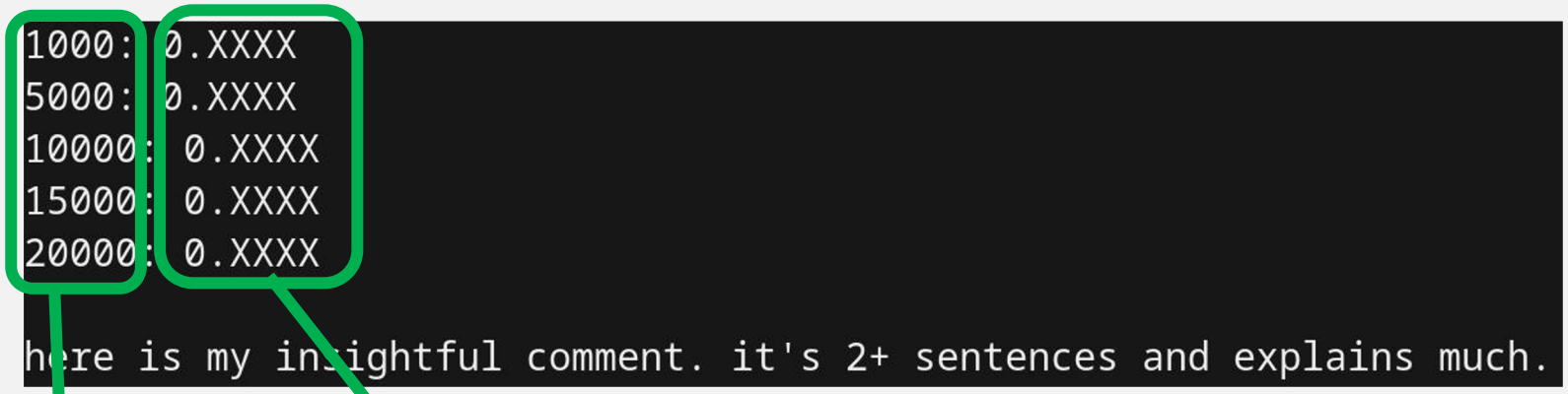
- You previously used a random $0.8 \cdot 40K = 32K$ comments to train.
- Using the classifier with the highest accuracy from Sec3.1, retrain the system using an arbitrary $1K, 5K, 10K, 15K, 20K$ samples from the original $32K$.

CLASSIFICATION 2:
AMOUNT OF DATA

```
1000: 0.XXXX
5000: 0.XXXX
10000: 0.XXXX
15000: 0.XXXX
20000: 0.XXXX

here is my insightful comment. it's 2+ sentences and explains much.
```

CLASSIFICATION 2:
AMOUNT OF DATA



Number of training **examples**

Accuracy on test set when training on the corresponding number of training examples.

CLASSIFICATION 3: FEATURE ANALYSIS

- Certain features may be more or less useful for classification, and too many can lead to various problems.
- Here, you will select the best k features for classification for $k = \{5,50\}$.
- Train the best classifier from Sec3.1 on just $k = 5$ features on both $1K$ and $32K$ training samples.
- Are some features always useful? Are they useful to the same degree (*p*-value)? Why are certain features chosen and not others?

CLASSIFICATION 3: FEATURE ANALYSIS

```
5 p-values: [0.XXXX, 0.XXXX, ... p-values [      5      feats]
50 p-values: [0.XXXX, 0.XXXX, ... p-values [      50      feats]
Accuracy for 1k: 0.XXXX
Accuracy for full dataset: 0.XXXX
Chosen feature intersection: {XX, XXX, XX, XXX} # should be 5 or fewer
Top-5 at higher: {XXX, XX, XXX, XXX, XX}        # should be 5

My answers to questions go here:
(a) answer
(b) goes
(c) here :)
```

CLASSIFICATION 3: FEATURE ANALYSIS

```
5 p-values:  [0.XXXX, 0.XXXX, ... p-values [5 feats]
50 p-values: [0.XXXX, 0.XXXX, ... p-values [50 feats]
Accuracy for 1k: 0.XXXX
Accuracy for full dataset: 0.XXXX
Chosen feature intersection: {XX, XXX, XX, XXX} # should be 5 or fewer
Top-5 at higher: {XXX, XX, XXX, XXX, XX}        # should be 5

My answers to questions go here:
(a) answer
(b) goes
(c) here :)
```

p-values for the {5,50} features when we set k=5 and k=50 for SelectKBest, using the full dataset

CLASSIFICATION 3: FEATURE ANALYSIS

```
5 p-values: [0.XXXX, 0.XXXX, ... p-values [redacted] feats]
50 p-values: [0.XXXX, 0.XXXX, ... p-values [redacted] feats]
Accuracy for 1k: 0.XXXX
Accuracy for full dataset: 0.XXXX
Chosen feature intersection: {XX, XXX, XX, XXX} # should be 5 or fewer
Top-5 at higher: {XXX, XX, XXX, XXX, XX}        # should be 5

My answers to questions go here:
(a) answer
(b) goes
(c) here :)
```

Accuracy for the best model from 3.1, trained on the 5 best features from the 1K dataset and the full dataset

CLASSIFICATION 3: FEATURE ANALYSIS

```
5 p-values: [0.XXXX, 0.XXXX, ... p-values      5      feats]
50 p-values: [0.XXXX, 0.XXXX, ... p-values     50     feats]
Accuracy for 1k: 0.XXXX
Accuracy for full dataset: 0.XXXX
Chosen feature intersection: {XX, XXX, XX, XXX} # should be 5 or fewer
Top-5 at higher: {XXX, XX, XXX, XXX, XX}        # should be 5

My answers to questions go here:
(a) answer
(b) goes
(c) here :)
```

Indices of the best features (in range 0-172)

- "**Chosen feature intersection**" means intersection of the top k=5 features selected for 1K and the full dataset
- "**Top-5 at higher**" means the top k=5 features for the full datset.

CLASSIFICATION 4: CROSS-FOLD VALIDATION

- What if the 'best' classifier from Sec3.1 only appeared to be the best because of a random accident of sampling?
- Test your claims more rigorously.

	Part 1	Part 2	Part 3	Part 4	Part 5	
Iteration 1						: Err1 %
Iteration 2						: Err2 %
Iteration 3						: Err3 %
Iteration 4						: Err4 %
Iteration 5						: Err5 %

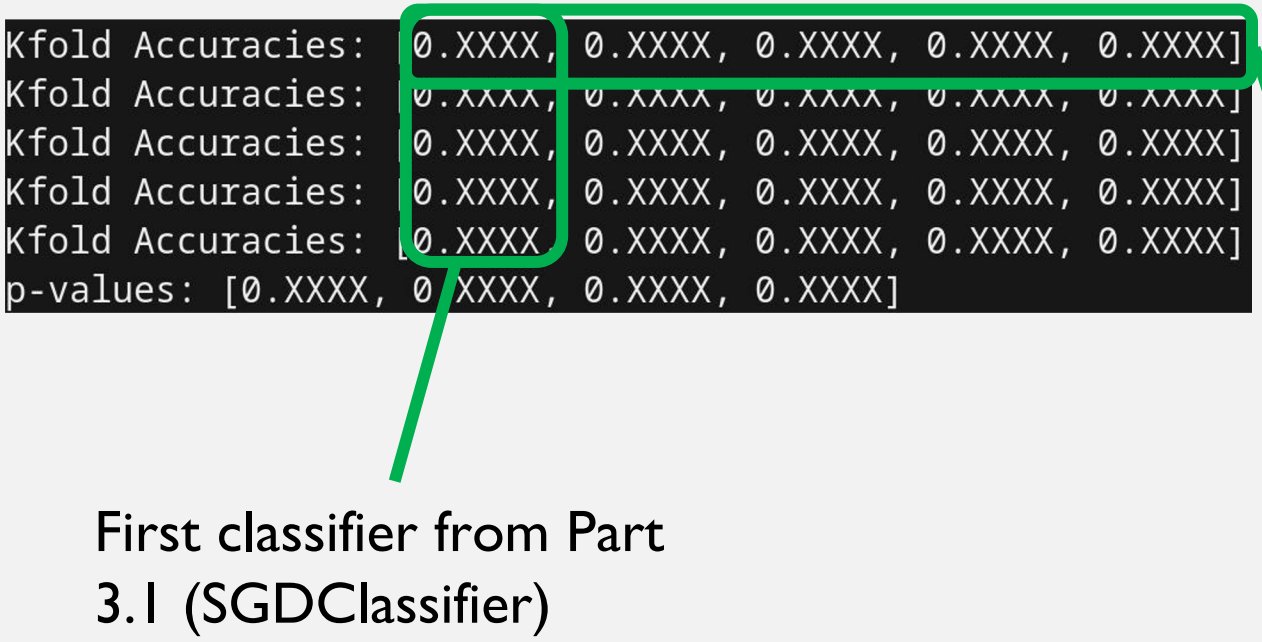
Testing Set

Training Set

CLASSIFICATION 4:
CROSS-FOLD VALIDATION

```
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
p-values: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
```

CLASSIFICATION 4:
CROSS-FOLD VALIDATION



First classifier from Part 3.1 (SGDClassifier)

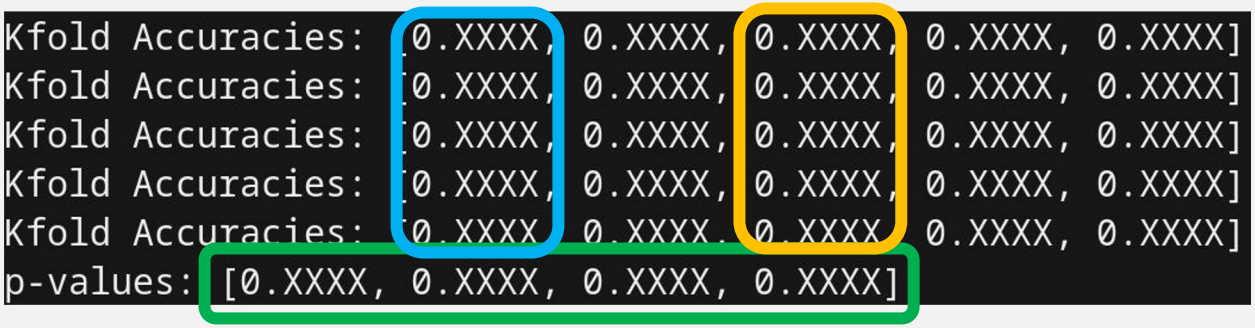
First fold of the data

CLASSIFICATION 4:
CROSS-FOLD VALIDATION

```
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
Kfold Accuracies: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
p-values: [0.XXXX, 0.XXXX, 0.XXXX, 0.XXXX]
```

p-values from t-tests comparing the accuracies across folds between the best classifier from Part 3.1 and the other classifiers

CLASSIFICATION 4:
CROSS-FOLD VALIDATION



p-values from t-tests comparing the accuracies across folds between the best classifier from Part 3.1 and the other classifiers

Ex: If the best classifier from 3.1 was the RandomForestClassifier (the 3rd classifier), then the p-values should be reported in the order: [1 vs. 3, 2 vs. 3, 4 vs. 3, 5 vs. 3].

What do these p-values tell us?

OVERVIEW

- In-depth look at Part 3
  - (last week you saw in-depth look at Parts 1 and 2)
- Breakdown of sample data
- Piazza recap
- Q&A

SAMPLE DATA

- 3 files:
  - **sample_in.json** -- input to a1_preproc.py
  - **sample_out.json** -- output of a1_preproc.py (to be fed to a1_extractFeatures.py)
  - **sample.npz** -- output of a1_extractFeatures.py

SAMPLE DATA

sample_in.json -- input to a1_preproc.py

```
>>> "{\"id\": \"c0b61z2\", \"subreddit_id\": \"t5_2r2jt\", \"body\": \"Hehe, I second this. I ADO
RE Clueless.\", \"downs\": 0, \"gilded\": 0, \"edited\": false, \"author_flair_text\": null, \"di
stinguished\": null, \"author_flair_css_class\": null, \"score\": 3, \"controversiality\": 0, \"u
ps\": 3, \"archived\": true, \"created_utc\": \"1247850061\", \"name\": \"t1_c0b61z2\", \"link_id
\": \"t3_9235m\", \"subreddit\": \"TwoXChromosomes\", \"retrieved_on\": 1426000662, \"author\": \
"[deleted]\", \"parent_id\": \"t1_c0b60eq\", \"score_hidden\": false}"
```

"body": "Hehe, I second this. I adore Clueless."

SAMPLE DATA

sample_out.json -- output of a1_preproc.py


```
{
  "id": "c0b61z2",
  "body": "hehe/UH ,/, I/PRP second/VBP this/DT ./.\nI/PRP ADORE/VBP clueless/NNP ./.\n",
  "cat": "Left"
},
```


hehe/UH ,/, I/PRP second/VBP this/DT ./.
I/PRP ADORE/VBP clueless/NNP ./.\n

SAMPLE DATA

sample_feats.npz -- output of a1_extractFeatures.py

```
[i>>> import numpy as np
[>>> sample_feats = np.load('./sample_feats.npz')['arr_0']
[>>> sample_feats[4]
 array([ 1.00000000e+00,  2.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  1.00000000e+00,
```

Tip: compare your output to the numbers in sample.npz using the method np.allclose()

OVERVIEW

- In-depth look at Part 3
  - (last week you saw in-depth look at Parts 1 and 2)
- Breakdown of sample data
- Piazza recap
- Q&A

REMINDER: TEMPLATE CHANGES

- a1_preproc.py
  - Updated regex for removing URLs
- a1_classify.py
  - Output format string for printing p-values

GENERAL QUESTIONS

- **Global variables**
  - You may define them outside of main, as we have done for wordlists.
  - IF you define them in the main() function, use the global keyword.
- **Versions** – Use Python 3.9 on cdf (where spaCy 3.2.1 is installed).
- **Runtime** – Parts 1 and 2 should each take around 10 minutes or less. Part 3 may take longer depending on CDF traffic. Be sure to use `alpha=0.05` for the MLPClassifier!
- Don't change **function headers** or string **output formats** for Part 3.

QUESTIONS ON PART 1

- **SpaCy version matters**
  - Tagging varies between spaCy versions. Use version 3.2.1.
  - There was some confusion about whether single-word comments raise an error. This is not an issue with version 3.2.1 (which you should use).

QUESTIONS ON PART 2

What counts as **future tense**?

- *'ll, will, gonna, going+to*+VB
- Note that "going" --> "go/VBG" after preprocessing.
- Don't need to worry about:
  - Cases with an elided verb (e.g. "I'm going to")
  - Non-standard contractions (e.g. "I'ma")

In general, we will consider whatever spaCy outputs to be the "correct" output, and we will not autotest unusual edge cases.

QUESTIONS ON PART 2

- What counts as **multiple punctuation**?
  - All characters in the token must be punctuation.
- What about **spaces in lemmas**?
  - (e.g. "N.Y." -> "New York/NNP")
  - Use your judgement, we will not test these cases.
- Why are my **uppercase counts** so low?
  - Problem: Lemmas are all lowercase, so replacing tokens with their lemma makes this feature less meaningful.
  - What to do: Continue using lemmas, since it's what's specified in the assignment handout.

QUESTIONS ON PART 3

- What are we supposed to do with train and test data in 3.4, when we're doing k-fold validation?
  - Re-combine these into one dataset, then use that.

OVERVIEW

- In-depth look at Part 3
  - (last week you saw in-depth look at Parts 1 and 2)
- Breakdown of sample data
- Piazza recap
- Q&A