CSE 512 Final Project Plan and Milestones
Brian Dolhansky and Marco Túlio Ribeiro

Building the scikit-learn backend layer

For a general application of this tool, we would like to make the choice of model fairly generic. Right now we're still limited to linear models, but we would like to handle different types of datasets besides just raw text.

- We would like to feed a general dataset to scikit-learn.
  - *Status:* complete
  - This was completed by **Marco**.
- ...and have the website frontend retrieve the linear feature weights from some server
  - *Status:* complete
  - This was completed by **Marco**.

Transitioning to Multiclass

Our initial visualization was tied to a binary classification problem, but most real-world classification tasks consist of multiple classes. Therefore we needed to rethink how we present our data to the user.

- The x-axis of the "databin" needs to correspond to classes now instead of prediction values. We lose some information by going to this format, but the prediction values will be displayed elsewhere. (See the relayout section).
  - *Status*: complete.
  - This was completed by **Brian**.
- We need to adjust our coloring to handle multiple classes.
  - *Status*: complete.
  - This was completed by **Marco**.
- Another way of thinking about the databin is by having the examples sorted by some class. The user could then choose which class is in focus. This interactivity would let the user explore different aspects of the data better
  - *Status*: complete.
  - This was completed by **Marco**.

Handling new types of data

We would like to be able to handle tabular data in addition to raw text. This is a maybe.

- Ensure the visualization can display the same visualizations for both tabular data and raw text.
  - *Status*: **incomplete**.
- When the data is too big, the databin becomes too crowded. We can cluster nearby points, and then click on them to expand. This would allow us to visualize much larger datasets
  - Status: **incomplete**

- Alternatively, we could provide a “zooming” mechanism where all of the points are shown, but if there are too many to reliable select with a mouse, a magnifying glass (similar to text selection in iOS) could be shown above the examples
    - Status: **incomplete**

New Visualizations

There are multiple new statistical visualizations we would like to add.

- We can visualize weight importance with a sorted horizontal bar chart.
    - *Status*: complete.
    - This was completed by **Marco**.
- In the “example” tab, we need to show feature co-occurence.
    - *Status*: **incomplete**
- In the “documents” tab, we need to show nearest-neighbor documents.
    - *Status*: **incomplete**
- It would be very interesting if we could allow two types of brushing (these are both maybes):
    - On features: brushing on some features would highlight the examples that contain those features in the databin
    - On examples: brushing on some examples in the databin could result in an explanation that is aggregate. We have no idea of how to do this though.

Relayout

In order to handle the more complicated visualiztions we’re developing, we need additional space to display them all.

- Create new storyboards
    - *Status*: complete.
    - This was completed by **Brian**.
- Develop a tabbed layout in which we can put the relevant visualizations.
    - *Status*: in progress, being worked on by **Brian**.
- Rearrange the interaction windows/info button.
    - *Status*: in progress, being worked on by **Brian**.

Final Touchups

After we have all of our visualizations and interaction modes developed, we need to clean up the website and develop a unifying theme.

- Ensure colors are similar
    - *Status*: in progress, being worked on by **Marco**
- Ensure that the interaction displays correctly for various browser window sizes
    - *Status*: incomplete