# Machine Learning

**Prompt**: Your task is to use your data science skills to identify a data story and build a data product from the New York City 311 Service Request data set. First, we'd like you to create a model to predict what the type of a given complaint will be. After that, we encourage you to add anything else you find exciting that showcases your skills: another predictive model, an interactive dashboard or visualization, an interesting statistical analysis, or anything else you dream up! Be creative, but the most important thing is to showcase your data skills.

## Predict Complaint Type

In a normal commercial use case you would have more details regarding the constraints of the task or at least the chance to infer or extract them from a product owner. There are no constraints regarding what fields can be used to predict the complaint type or any implication on what problem is trying to be solved.

The agency and description of the complaint type are likely to be extremely related conceptually to the complaint type and so should not be considered completely independent variables from each other. Its important to know this going into the model creation because this will make it more likely for my model to perform well, but it will be knowingly introducing bias. This bias would be acceptable if we were looking to predict complaint type in an effort to flag data that might be inaccurate.

Ignoring the dependence some variables have on each other would not be acceptable if we were creating an evaluation of different agencies based on how long they take to close a request since the agencies will naturally be dealing with different complaint types which will require different resources.

### Plan

1. To use the words in the descriptor field to train an NLP NN to predict the complaint type.
2. To use the categorical variables to predict the complaint type: agency, incident\_zip, hour of day
3. To combine the two models into a functional model so that the prediction of the NLP model can be passed in as a categorical input along with the values from the categorical model
4. [don’t expect to get this far] To run accuracy tests to hold categorical inputs fixed and vary the time of day for a variety of input combinations to measure the impact that the time of day variable has on the prediction
5. [don’t expect to get this far] To abstract the previous function so that it can be applied to zip code and time of day using the same function
6. [don’t expect to get this far] try to predict which borough the Unspecified records belong to. Maybe they had a zip filled in and this would be trivial, don’t know

### Hyper-parameters

listing out hyperparameters that aren’t obvious

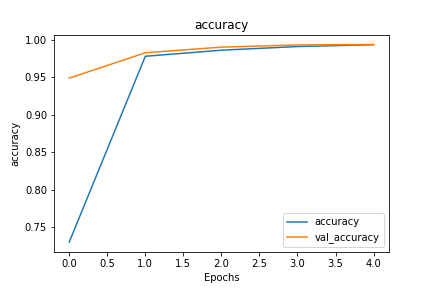
* limit = size of dataset sampled from entire database
* min\_freq = number of minimum occurrences of a complaint type before it is used in the training
* max number of words in the vocab is 256, any other words are set to OOV
* max length of the padding done on the input sequences
* obviously all the hyper-parameters in the model definition

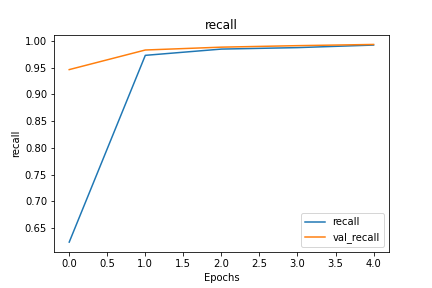
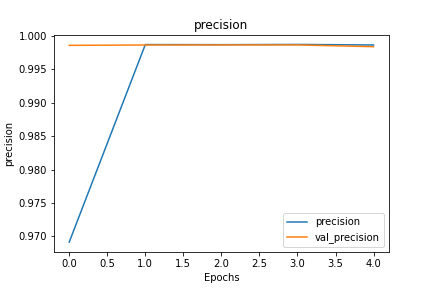
### Preprocessing

* ignore records where descriptor is empty because that is what is going to be used as the input
* ignore the complaint types that don’t happen often - good place to start to get a model working before expanding the complaint types.
* a vocabulary is created from the words it sees in the training records for descriptor
* a full list of categories is taken from the database that occur frequently enough
* after tokenizing the text, it is then padded or truncated to uniform length

See Predict\_ComplaintType.ipynb for details

Unfortunately this complaint type prediction wasn’t interesting. I checked accuracy, precision, and recall. They all performed well on train, dev, and test sets. I increased the number of categories I was considering to make the prediction harder on the NN, and decreased the parameters inside the NN. With more time I could try predicting it with the categorical variables, and I could expand the categories more as well as measuring the accuracy of the prediction on the infrequent categories compared to the frequent categories.





# Questions

1. I couldn’t get the [DATE\_PART\_YEAR function](https://docs.aws.amazon.com/redshift/latest/dg/r_DATE_PART_YEAR.html) to work with “SELECT COUNT(unique\_key) , DATE\_PART\_YEAR(created\_date) FROM public.three\_one\_one GROUP BY DATE\_PART\_YEAR(created\_date)”