Summary

**Files**

* Jupyter Notebook Models: model\_svm.ipynb, model\_deep\_learning.ipynb
* Saved Models: Nader\_DeepLearing.h5, Nader\_SVM.sav

**Problems / Unresolved Issues**

Deep Learning (Neural) Model

* Grid Search & Saving Model as “Sav” file: Unable to surpass error message “can't pickle \_thread.RLock objects” after multiple tries as trying to adjust various parameters

**Findings**

*Deep Learning (Neural) Model*

* Unable to utilize Grid Search therefore cannot identify the most optimal parameters
* Accuracy rate is 88%, which is decent however we don’t have a breakdown of each unique classes’ score (i.e. separate scores of False Positive, Confirmed, & Candidate)

*SVM Model*

* Weighted Average F1 score is 84%, which is close to the accuracy score of the Deep Learning Model.
* Best F1 score via the grid search implementation is 88%, which is equivalent to the Deep Learning accuracy rate.
* SVM Model outputs the F1 score for each individual class, we notice False Positives are significantly higher the remaining other 2.

**Conclusion**

The SVM model is a better model to adopt due to:

1. Ability to identify unique classes F1 score individually – This makes it easier to point at which classes we are struggling to predict accurately instead of only dealing with an average accuracy rate.
2. Grid Fitting Model: The ability to search for the best hyper parameters is an edge to identify the best parameters, unlike the issue with the Deep Learning as it is hard to implement a grid search model.