The NHL playoffs are just around the corner, so the perfect opportunity to combine our passion of data analytics and hockey into one project was formed. The Hockey Abstract, a website that shares a wide variety of player statistics shares seasonal data on their website. With over 200 different stats reported, it serves as a great data set to explore hockey and predictive capabilities on continuous variables. In this project we set out to predict a player's salary based on the statistics found from the Hockey Abstract (http://www.hockeyabstract.com/testimonials/nhl2017-18).

We broke the bulk of the project into two Jupyter Notebooks. They are split between data cleaning and exploratory data analysis (EDA), and model building. The data comes from the 2017-2018 season, and contains fan sources as well as stats directly from the NHL. The raw file is an Excel Pivot Table. It has several different pages of statistics, as well as a Legend and About page. This analysis makes use of the first page of data, where the Legend is used for flattening the pivot table into a Pandas dataframe in python. We explore the data, and eventually automate the processing into Pandas by creating our own abbreviations. On top of the importing process, some of the data also needed cleaning, as it was missing values in some of the columns. A background in hockey helps fill in missing values for players such as shoot out-shots, since if you don't play in a shoot-out you simply have zero. While others required some thought such as missing draft information. We share our flattened and cleaned data in the files: nhl_2017_18_condensed.csv, nhl_2017_18_numeric_data.csv, and nhl_2017_18_text_data.csv.

Our analysis took advantage of the Python Bokeh library, performing the majority of the EDA by creating three interactive bar plots that a user can explore. These can be found in the 3 notebooks having Interactive Graph in their title. They create bar graphs comparing a variety of stats for 1 to 10 top players/teams, as well as scatter plots to visualize the salary and correlation of two recorded statistics. The interactive route made analysis easier, so as not to have to make many calls to Matplotlib to create all the required plots. Also, users who may not know as much Python can make quick use of the interactive functionality.

The second half of this project worked towards the main goal of predicting player salary from the given dataset. We made use of the numeric columns, as well as creating dummy variables from categorical text columns. After a first pass it became clear that Linear Regression, with no extra penalization, would be best. The Ridge method, which uses L2 normalization to keep model coefficients low, did not outperform Linear Regression. This was shown using parameter tuning of Ridge and comparing graphically to Linear Regression. Instead we used a forward wrapping method combined with k-fold cross validation to find the most influential features, and keep them for fitting a model. The model was successful with an R^2 score of 0.78 on the training data, and 0.75 on unseen test data. The most influential statistic was found to be Point Sharing, a catch all that follows how well a player contributes to the point's standings of their team. It was closely followed by a given team's expected goals based on when a player is on the ice.

Given more time we could improve our method with a more adhoc approach to explore what caused the model to predict a small number of unphysical salaries, shown by negative values. As we were using multidimensional regression that uses a hyperplane we could not quickly visualize an issue and track it down. Other avenues for future improvement would be including text based analysis on a player's choice of equipment.