**Section 2:**

With the course’s focus on classification and the encouragement in the instructions to do a project involving classification, we decided to create a classification model that would predict a CEO (Chief Executive Officer) versus a CFO (Chief Financial Officer). One of our biggest challenges involved how to come to our question. This is not a dataset that we found on Kaggle or a site that specifically curated it for the purposes of using and discovering classification techniques. Rather, it is a completely unedited data scrub from the SEC API. The process to discover our question was iterative, and we relied on exploratory data analysis and combining our intuitions in order to carve our path. Our overall goal was simple: demonstrate knowledge gained in ADS502 by using key learnings from the course to apply to “wild data.”

With this in mind, we started with two datasets, one for executive pay, and the other for company financial data. We extracted only the companies listed in the S and P 500 over the last 10 years. For each company’s financial records, we turned excel workbook files into CSV files in order to more easily manipulate the data. Thereafter, we combined the CSV files of the S and P 500 companies into a single Pandas data frame. We then grouped the data into average values for every year based on its ticker. For example, Apple will have average values specific to its financials for each year over the last 10 years.

Next, we moved to the executive pay dataset, and extracted the executive information for those individuals within the S and P 500 ticker list over the last 10 years. After this, we combined the two separate data frames into one by joining them via their tickers. Now that we had our main data frame ready, we performed cleaning and pre-processing before sub-setting the data frames by CEO and CFO. We did this both so that we could calculate a sum of the earnings based on the ticker grouping over the last 10 years separately for CEOs and CFOs, and also to remove the other executives included who were not either a CEO or CFO. Once this was done, we concatenated them together, essentially pasting one on top of the other. Once this data frame was ready, we partitioned it to be 40% testing and 60% training. This was from the guidance of the textbook, where it states, “for smaller or less complex data sets, one should retain sufficient records for accurate assessment, so that the training sets would contain only 50–67% of the original data" (Tan et al, 2019). We felt that this was a small enough dataset to keep 40% of the records in the test set.

At this point, we were almost ready to run the classification models and select the best one. Before this, however, we first scaled the data because of the large numbers involved when summing over ten years. Then we ran our first classification model as a CART model. We ran the complete cell with the setup, then we decided to create a function so that the other models could use the function to create a confusion matrix based on the model’s train/test datasets. This made running the other models more space-efficient, as only one line of code was required for each one. We proceeded to run CART, C5.0, random forest, logistic regression, naïve bayes, K Neighbors, and support vector machine models.

**Section 4**

The first step was to get the data we needed from the SEC API for executive pay, and from Stockrow for the company financials. After gathering the data from both sources, we decided we wanted to use only S&P 500 data and so we also loaded a data source containing the 503 tickers. We matched the data from the company financials to only return data for each ticker in the list of 503. For each ticker in the list, we created CSV files and then combined all the CSV files into a Pandas data frame to perform manipulations on. Now that our data frame was ready, we narrowed the fields into just ones we wanted: Ticker, Date, Revenue, Gross Profit, Operating Income, Income after Tax, and Net Income Common. Additionally, we created a datetime object from the Date feature and extracted the year out of it. This let us ultimately break down the information by yearly basis so that we could establish yearly trends.

Next, we grouped the data by ticker (representative of company) and year and got the mean values for each ticker per year. To make sure this was done correctly, we sanity checked “AAPL,” Apple, and found that the numbers matched when we did an independent search online. From here, we were ready to join in the executive pay data set and did so based on the ticker; only executive pay information for the S&P 500 companies were included. Note that not all companies were in the executive pay dataset, and after merging them together only 485 companies remained with 18 not present in the executive pay from the SEC API.

For data cleaning, we noted some duplicate records and performed a standard drop\_duplicates() method was used to eliminate these duplicates. The company Agilent had duplicate records that were duplicates in reality, but had only one slight difference that caused them to get past the drop\_duplicates method. We eliminated these duplicates by deleting the column that was supposed to contain full names but instead contained “human resources” and “group.” We noted about 200 outliers in the executive pay data and dropped these as well. One such outlier was a salary of 18 billion US dollars in one year. Additionally, there were also some missing values in the company financials. For instance, 73 of the 485 companies did not have revenue data in 2012 and 14 did not have revenue data in 2021. We decided not to impute values or drop these companies because our classification model was classifying executives as either CEOs or CFOs. The more important metrics, in our eyes, were individual pay statistics.

For exploratory data analysis (EDA), we aimed to find patterns in the pay of CEOs over the last 10 years. “CEO Salary Percentiles 2012-2021” measured the percentiles of salaries the CEOs on the x-axis with the income number on the y-axis. This means if a salary is the highest in the dataset, it will be in the 99th percentile, and if a salary is the lowest in the dataset, it will be in the 1st percentile. From this visualization, we can see that there is a consistent slope until roughly the 80th percentile mark is reached. This means that the increase in income from say the 40th percentile to the 60th percentile is reliable and predictable. This is not the case after about the 80th percentile however, where we see a steepening of the slope of the line for all years. This indicates that those CEOs above the 80th percentile earn much than those below them. A comparable increase in earner percentile, say 20% increase, from the 40th to 60th is much less in real dollar value than a 20% increase from the 70th percentile earner to the 90th percentile earner. Additionally, we highlight the lines for 2020 and 2021 to show the difference between these two years and note that 2021 is much higher for salary than 2020 for any given percentile chosen.

We have also created boxplots, violin plots, kernel density estimate (KDE) plots, a color mapped kernel density estimate (KDE) plot, and bar plots to investigate a multitude of factors. Our focus was on the changing CEO pay landscape, and we used visualizations to investigate changes in salary, stock awards, non-equity incentive compensation, and total compensation. We found a trend of CEO salaries rising in the last 16 years (since 2006), and the most visually telling representation of this trend was the KDE plot, which displayed a much higher density of incomes with years being represented by colors.

When analyzing stock awards, we see a surprising widening of box spread in 2021, with very few outliers relative to prior years. This is part of an overall trend of widening interquartile ranges between 2006 and 2021 and may be due to stock awards being a more common form of compensation now compared with years past. Such a short box with extreme positive skewness in 2006 indicates that a high proportion of CEOs had low stock options as part of their compensation package. In fact, for all years the data is very positively skewed, which indicates that the mean is larger than the median and is heavily impacted by outlying very large values. This is a logical conclusion based on the observation that the top 20% of CEO earners are making much more than the lower 80%. The large stock awards going to these top earners drags up the mean and results in a high positive skewness.

Finally, it is worth noting that companies vary wildly in how they pay their top earners. For instance, Microsoft has a very exaggerated change the year that they elected to have Satya Nadella become the new CEO in 2014. In years prior, the CEO position had collecting hardly any compensation from the company. This is the way that Bill Gates, the prior CEO, had decided to run his business that he had founded. But when it was time to make a change, the chief executive position started to make thousands of times the amount as before. Microsoft is only one example, and it is conceivable that other CEOs that founded their companies likewise have decided to have very low compensation as well.

**Section 5**

Our basic goal for model selection was to follow the textbook’s advice to select the “to select the model that shows lowest generalization error rate” (Tan et al., 2019). We considered other possibilities for optimal selection, such as minimizing the amounts of false positives or false negatives. Ultimately, we felt that a generalized accuracy measure would work well for this type of data, as there is not a need to be extra conservative like there would be for, say, diagnosing whether a patient’s health indicators classify them as having a disease or not having it. In this case, it would be better to have false positives (patient does not really have the disease but tests say they do) rather than false negatives (patient really has the disease but tests say they do not).

To evaluate our models, we partitioned the data to “assess the performance of a learned model on a labeled test set has not been used at any stage of model selection” (Tan et al., 2019). We felt that the data set was sufficiently small to follow the books advice for smaller or less complex datasets, where “one should retain sufficient records for accurate assessment, so that the training sets would contain only 50–67% of the original data” (Tan et al., 2019). Our main concern was balancing the proportion so that the training model could effectively learn the behavior but not be so large that the testing model’s error metric was unreliable because it was not able to compute over enough instances. Ultimately, we decided on a 70% partition into the training set and 30% into the test set.

We tried different models to see which would produce the best results for reducing our error rate, producing the best accuracy for true positives. The best model for this purpose and which we felt was the best was the logistic regression model.

**Section 6**

Classifying CEOs and CFOs is only scratching the surface of the type of information this data source is capable of. Our initial goal was to solve the more politically hard-hitting question of, “do executives actually get paid based on their companies do or do they reward themselves no matter what the company does?” We wanted to create a classification model that would predict if a CEOs or CFOs compensation trends would correctly classify if a company was a rapid grower, flatliner, or decliner. Even though the 2010s was a very prosperous time in the American economy, we found that about 70 of the companies in the S&P 500 had declining revenues during the period. One of the beautiful elements of the economy though is that growth is limitless, while the maximum amount of decline possible is just 100%. Thus, due to the fantastic success of companies like Tesla and Amazon, the growth of the overall economy far outweighed the companies bringing down the overall index.

Beyond classification, the data set also has potential for linear regression projects, as well as unsupervised algorithms like clustering. For example, a linear regression project could investigate which factors are most important to overall compensation for executives. Variables such as company revenue, company net income, position title, age, location, and many more could provide indicators as to which factors affect executive pay statistics the most. Clustering algorithms could be used to find trends that are worth exploring for supervised algorithms that are not known from an initial view of the data.