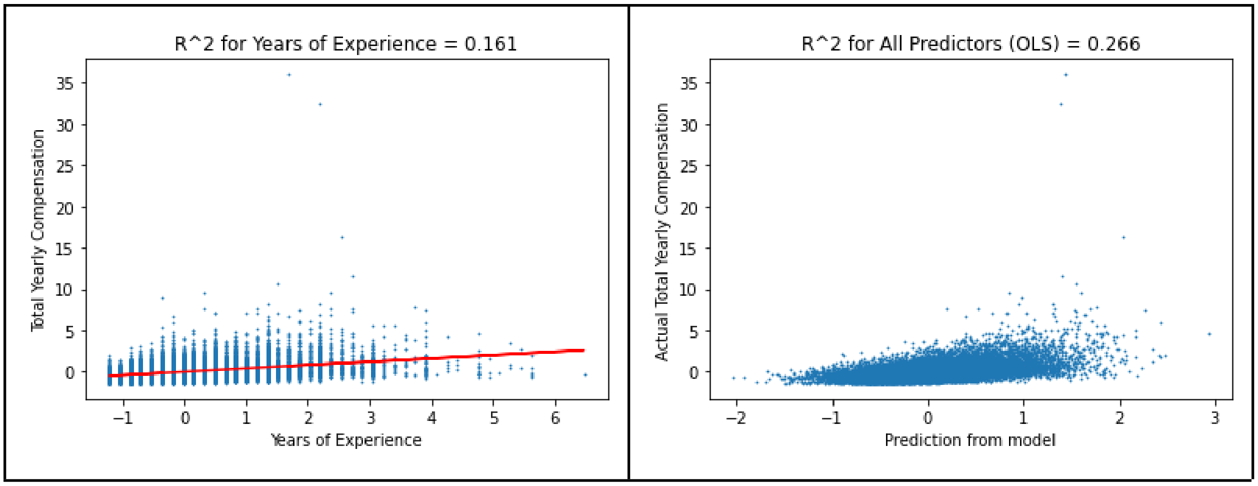
Simone Rittenhouse
Prof. Pascal Wallisch
Into to Machine Learning
9 March 2022

Intro to Machine Learning: Homework 2

1. **Using multiple linear regression: What is the best predictor of total annual compensation, how much variance is explained by this predictor vs. the full multiple regression model?**
    a. To find the best predictor of total annual compensation, I first re-coded gender into 0 = 'Male', 1='Female' and 'Zodiac' into twelve binary variables. I subsetted my dataset to include only quantitative variables. I then dropped all rows in which there were missing values for 'Education', 'Race', or 'gender.' I also dropped the dummy variables 'Highschool,' 'Race_Two_Or_More,' and the fourth zodiac variable to prevent the model from being overdetermined. I then normalized the dataset (using z-scoring) and ran a multiple linear regression to see the coefficients of each predictor and the R-squared. Lastly, I ran simple linear regressions, one for each predictor, to find the R-squared of each normalized predictor individually.
    b. I re-coded gender and zodiac because they were both categorical variables and weren't meaningful in their unprocessed form. I also dropped the gender values 'Other' because there were only 400 rows (out of 20,000+ observations) and binary gender was more interpretable than re-coding the categories into three distinct values. I chose to drop 'Highschool,' 'Race_Two_Or_More,' and the fourth zodiac variable to prevent an overdetermined model because, out of the dummy variables for race, education, and zodiac, they were the least correlated with total annual compensation and therefore the least predictive in linear regression. I normalized the data to ensure that the betas between predictors were comparable and not affected by scaling differences. I also ran simple linear regression to compare how much variance was explained by each predictor individually versus with the full model.
    c. The simple linear regression model with the highest R-squared used the predictor 'yearsofexperience' (R-squared = 0.161). Additionally, in the full multiple regression model, 'yearsofexperience' had the largest beta coefficient of 0.387. The multiple regression model had a larger R-squared of 0.265552.



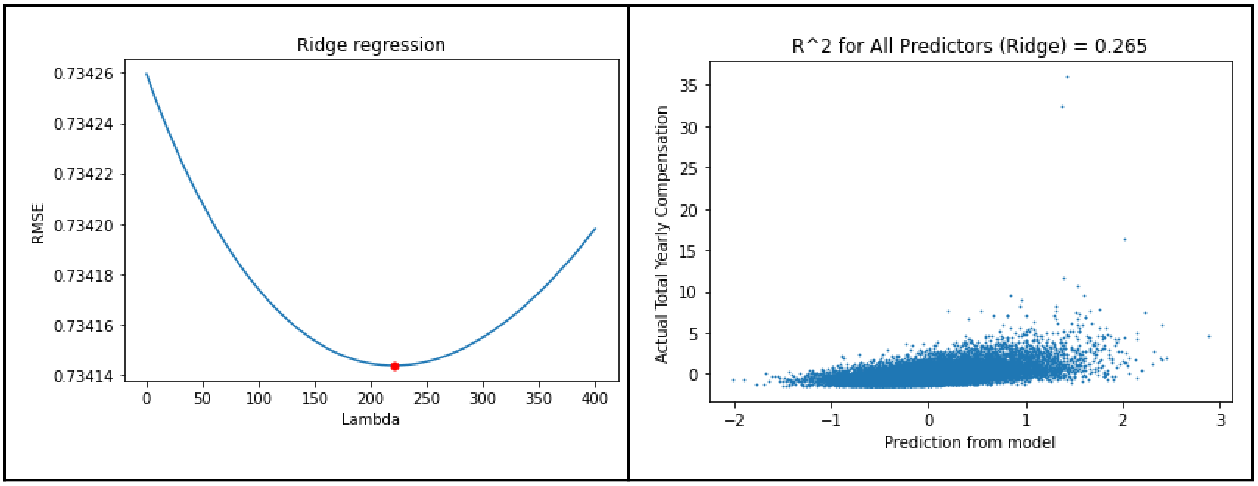
d. Based on my results, the best predictor of total annual compensation is years of experience. Years of experience alone in a simple linear regression model explains 16.1% of the variance in total annual compensation. This is smaller than the variance explained by the full multiple regression model, which explains 26.6% of the variance. This is to be expected, however, as the R-squared of a multiple regression model will increase as more predictors are added. However, in both cases, the R-squared is still relatively low, which is likely due to some outliers (people who make much more than average) in the data set. These outliers can be seen in the scatter plot for the full regression model.

2. **Using ridge regression to do the same as in 1): How does the model change or improve compared to OLS? What is the optimal lambda?**

a. Using the same normalized quantitative data as in question 1, I used train_test_split() to create a training and test data set, with a test set of 30% of the total data and a random state of 0. I then created 1001 lambda values between 0 and 400 to perform hyperparameter tuning. I iterated through the lambda values, creating ridge regression models fit to the training data and finding the RMSE on the test data. I used the lambda with the smallest RMSE across these 1001 values as my optimal lambda, and built a ridge regression model on the full normalized data set (as in question 1) using this lambda. I then found the R-squared and beta coefficients to compare with the previous question.

b. I used the same data as in question one because I wanted to compare how ridge regression changes the betas, so I again wanted unitless beta coefficients - as in question 1. Additionally, using normalized data meant that I wasn't regularizing the intercept, a problem we discussed with the sklearn implementation of ridge regression. This is because z-scoring my data made the intercept = 0. I then split the data in order to perform hyperparameter tuning because the RMSE on the data that is used to build the ridge regression model is not meaningful. Instead, we need to evaluate how different lambda values cause the models to perform on new data (i.e. test data). I used 1001 lambdas to iterate over as it allowed me to see how the RMSE changes with enough granularity to see a local minimum. I also chose an odd number of lambdas so that there was a minimal value. I chose a train_test_split() test level of 30% because this level gave me the best results when trying to find a lambda value that minimized the RMSE. However, I'm very suspicious of my results, as the optimal lambda value was highly dependent on the training/test split size as well as the bounds of the lambdas I used - and varied significantly when testing other test set sizes and lambda values. I suspect this could be because of the outliers in the dataset. With different test set sizes, the outliers are more or less likely to appear in the test data, which could significantly affect the RMSE of the models. Additionally, I built a separate ridge regression model using the full data set (as in question 1) with the most optimal lambda from hyperparameter tuning because I wanted to compare the performance of the model (using R-squared) on the same dataset as in question 1.

c. Despite my concerns regarding the validity of my results, I found an optimal lambda value of 220.4. Using the full dataset, the model's performance did not improve, and stayed almost the same (R-squared = 0.265427). The R-squared decreased slightly from question 1; however, the decrease was very small (~ 0.0001) and is possibly due to numerical error. Additionally, the beta values across all 26 predictors decreased in terms of their magnitude, which is to be expected using ridge regression. Years of experience still had the largest associated beta coefficient, but instead of 0.387 it became 0.379.



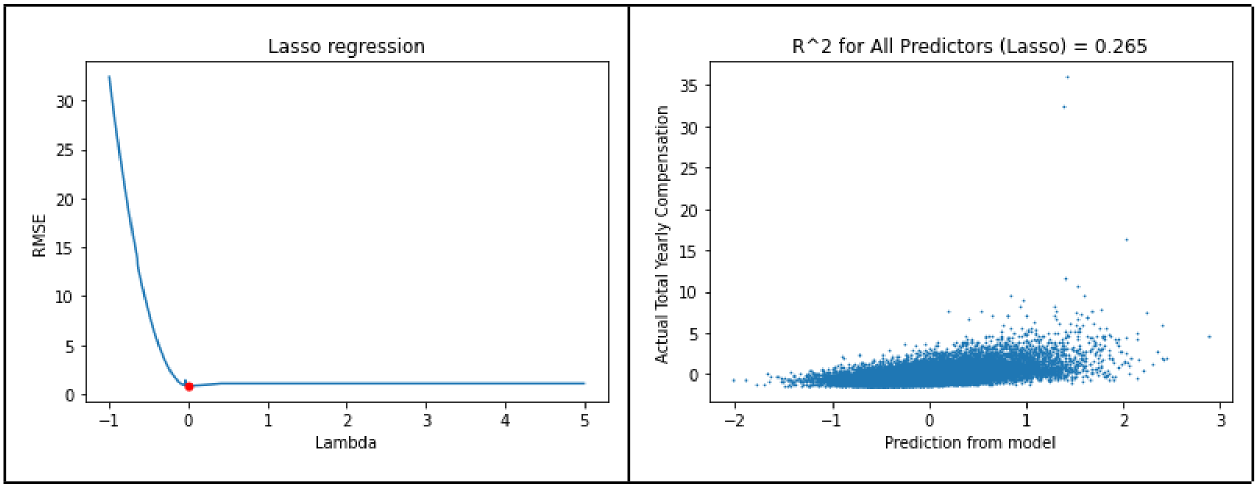
d. Using hyperparameter tuning and cross-validation through a train_test_split(), I found an optimal lambda of 220.4. This is an extremely large lambda value, which is surprising and makes me a bit suspicious of my results. Additionally, the overall performance of a ridge regression model using this lambda and the same data set as question one had virtually the same performance as that obtained using multiple linear regression, with an R-squared of 0.265. One key difference was that, as expected, ridge regression shrank the beta coefficients of the model towards 0. However, the decrease in beta coefficients was also very slight (e.g. 0.387 to 0.379).

3. **Using Lasso regression to do the same as in 1): How does the model change now? How many of the predictor betas are shrunk to exactly 0? What is the optimal lambda now?**

    a. I again used the normalized data, and split it into a training and test set, with a test set size of 20% and a random state of 0. Similar to my hyperparameter tuning for ridge regression, I created 1001 lambdas between the values of -1 and 5. I then iterated through these lambda values, creating Lasso regression models using the training data and finding the RMSE on the test set. I used the lambda value that minimized the RMSE within my tested lambda range, and I then used this optimal lambda value to create a Lasso regression model for my entire dataset. I again found the R-squared to evaluate the model's performance and the beta coefficients to compare with questions 1 and 2.

    b. My reasoning for Lasso regression followed much of the same logic as my process for question 2. I used the normalized data once again so that my beta coefficients would be comparable to those in questions 1 and 2. Additionally, I wanted to normalize the data to prevent the sklearn Lasso implementation from regularizing the intercept of the model. I again split my dataset to perform hyperparameter tuning to evaluate the performance of each model on a new test dataset. The values of lambda were chosen after trial and error and based on where I observed a local minimum in the RMSE, with 10001 values chosen for granularity and because an odd number allowed me to observe a single minimal value. I am again very suspicious of my results. I received convergence warnings for the Lasso model during tuning, which led me to increase the tolerance of the models to 1. The results of finding an optimal lambda were also highly variable depending on where I set my minimal and maximal lambda values, as well as the test size I used during the train_test_split(). This may again be due to the outliers in total yearly compensation; however, it does make me question the validity of my results. I also found an optimal lambda value extremely close to zero, which may suggest that the optimal model would be multiple linear regression (as a lambda value of 0 is linear regression without regularization). Lastly, I found the R-squared on the entire dataset to compare with the previous questions and outputted the beta coefficients to examine which variables shrunk to zero.

    c. Even though I remain suspicious of my results, I found an optimal lambda value of 0.002. Using the full dataset, the model's performance did not improve over ridge or unregularized multiple linear regression. I achieved an R-squared of 0.265166, which is slightly lower than the R-squared achieved using ridge regression (a difference of about 0.0003) and the R-squared found for multiple regression (a difference of about 0.0004). Some of the beta coefficients decreased slightly in magnitude when compared with multiple linear regression, but this change was not as drastic as that observed for ridge regression. Additionally, out of my 26 total predictors, my Lasso regression model shrunk 6 of the betas to 0.



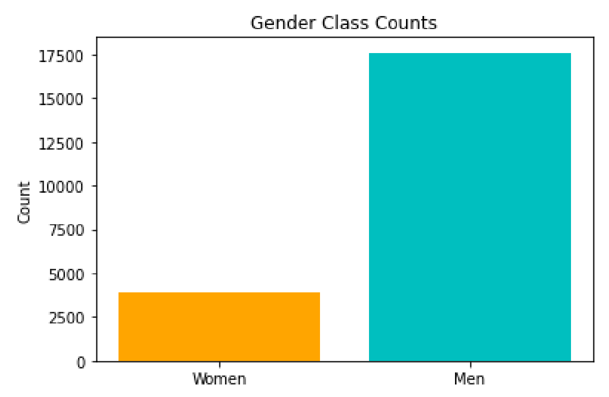
d. After performing hyperparameter tuning, I found an optimal lambda value of 0.002. The model performed slightly worse than in the previous questions with an R-squared of 0.265; however, this difference is quite small and likely insignificant (a maximal decrease of about 0.0004). Additionally, the model lowered the magnitude of some of the normalized beta coefficients and shrunk 6 of the 26 betas to 0. I expected a much more drastic change, which again makes me suspicious of my findings. However, with a lambda value close to zero, I am not surprised that so few of my predictors were removed with the model.

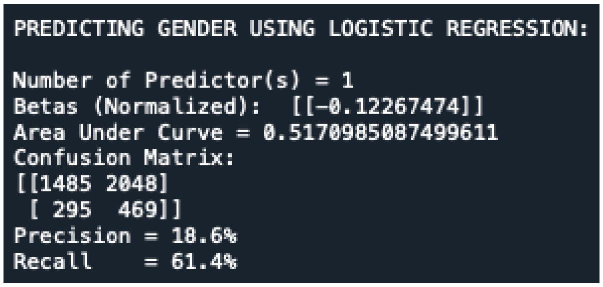
4. **There is controversy as to the existence of a male/female gender pay gap in tech job compensation. Build a logistic regression model (with gender as the outcome variable) to see if there is an appreciable beta associated with total annual compensation with and without controlling for other factors.**

   a. I used all quantitative variables in the dataset (after dropping NaNs), with gender re-coded as 1 = ‘Female’, 0 = ‘Male.’ I first assessed the balance between the two categories using a bar chart. I then split the dataset into a training and test set with the test set being 20% of the total data. I iterated through the 26 predictors, starting with total annual compensation and adding another predictor each iteration. I built logistic regression models each time using the training data and normalizing the predictor variable(s). I used the solver ‘liblinear’ and the class_weight 'balanced' in the logistic regression models. Then, using the withheld test data, I found the confusion matrix, precision, and recall for the model that just used total annual income to predict gender and found the beta coefficients and the AUC for all 26 models. Lastly, I plotted the values of the beta coefficients associated with total annual compensation across the 26 models, with a line marking the cutoff at which other beta coefficients in the model were larger in magnitude than the value for total annual compensation.

b. I plotted the class counts for gender to see if there was imbalance. I found significantly fewer women, which is why I used the class_weight 'balanced' in my models to reweight the data so that men are down weighted and women are up weighted. I also used the solver liblinear because the dataset became significantly smaller after dropping missing values (21,485 rows), and liblinear works well for small datasets. I split my dataset so that I could calculate metrics on data that is separate from the set used to build the models, which allows me to better evaluate the models' performance and used 80% of the data to train the model because this is a standard train/test ratio. I built 26 models, adding more predictors to total annual compensation because I wanted to see if total annual compensation was the largest predictor of gender with and without controlling for other factors. I normalized the predictors so that the beta coefficients were comparable and not affected by scaling differences. Lastly, I found the AUC for each model to see if gender could be predicted better than chance using the various predictors (i.e. to evaluate model performance).

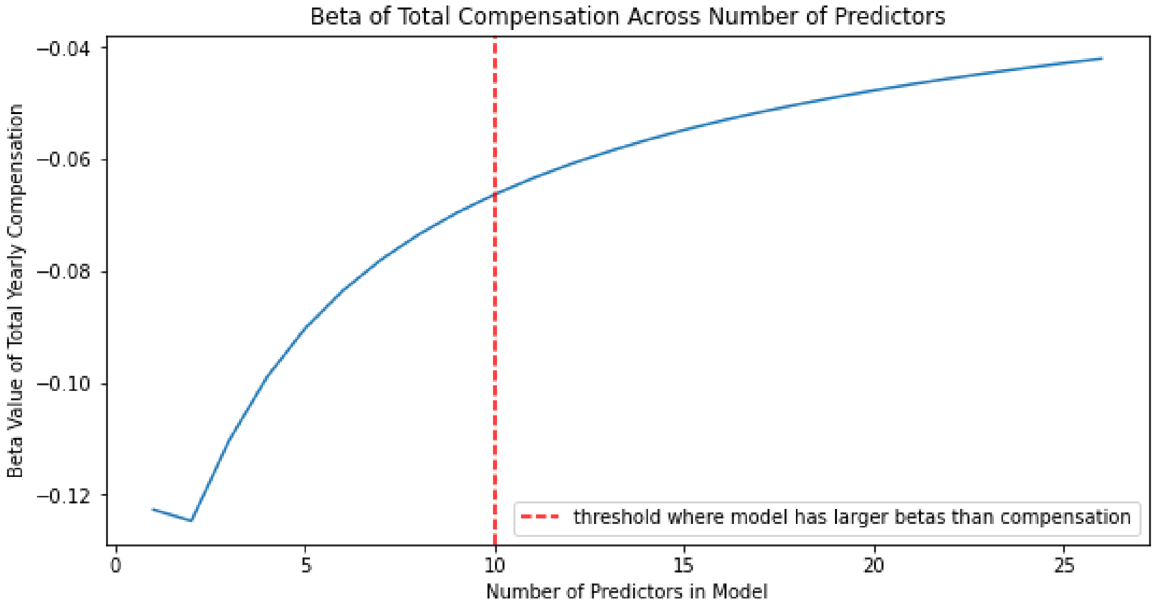
c. My bar chart showed significant imbalance between men and women in the dataset, with a much greater amount of men than women. The model that just used total annual compensation to predict gender had a beta coefficient of about -0.12 (although this value varies at each runtime based on the random train/test split). The negative direction implies that as compensation increases, the model is less likely to classify an observation as 'Female.' The confusion matrix for this model shows more false positives than false negatives, with a precision of 18.6% and a recall of 61.4%. The AUC was 0.517, which did not significantly change as more predictors were added. Additionally, the magnitude of the beta coefficient associated with total annual compensation decreased as more predictors were added to the model, eventually becoming less extreme than other beta coefficients. The cutoff at which this occurs also varies at runtime; however it is typically around 10-13 predictors.





```
PREDICTING GENDER USING LOGISTIC REGRESSION:

Number of Predictor(s) = 1
Betas (Normalized):  [[-0.12267474]]
Area Under Curve = 0.5170985087499611
Confusion Matrix:
[[1485 2048]
 [ 295  469]]
Precision = 18.6%
Recall    = 61.4%
```



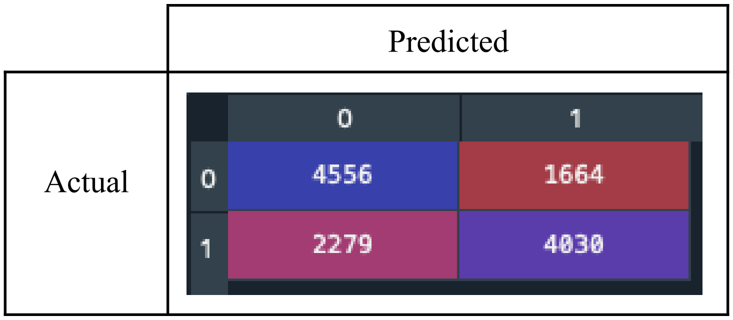
d. I did not find an appreciable beta coefficient associated with total annual compensation when predicting gender. The AUC across all 26 models was around 0.52, which is only a slightly better performance than random chance, which would have an AUC of 0.5. The model that just used total annual compensation also had a poor performance, with a high recall (meaning that the model had few false negatives) but very low precision (meaning there were many false positives). The beta for total annual compensation also became less extreme (i.e. less predictive of gender) as more predictors were added, eventually becoming less extreme than the other betas in the model. This implies that total annual compensation is not the best predictor of gender in a full model. This and the low AUC show there is no appreciable beta for total annual compensation when predicting gender, negating the existence of a gender pay gap in the dataset.

5. **Build a logistic regression model to see if you can predict high and low pay from years of relevant experience, age, height, SAT score and GPA, respectively.**

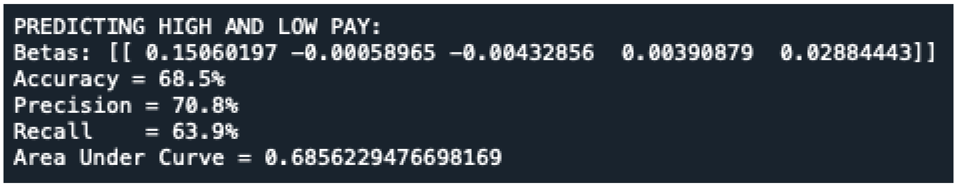
a. To predict high and low pay, I created a new outcome variable where the high pay group was observations where total yearly compensation was greater than the median compensation (coded as 1) and low pay was observations with compensation <= to the median (coded as 0). I then split the dataset containing the high/low pay outcome variable and the five predictor variables into a training and test set, with 20% of the data going into the test set. I built a logistic regression model using the training data with a liblinear solver and found the accuracy, precision, recall, and AUC for the model using the test data. I also outputted the confusion matrix.

b. I used the median to create the cutoff between high and low pay because the dataset had several outliers, and the median is more robust to outliers than other measures of central tendency (e.g. the mean). I also chose the median because, by definition, this would mean the two classes have the same number of observations, so there would be balance. I then split the data so, as in the previous question, I could use different data to assess the model's performance than the data used to build the model. I used a 80/20 training/test split as in question 4 because this is a standard split size. I also used the liblinear solver because, while other solvers are faster for large amounts of data, I didn't notice any lag in runtime using the liblinear solver on the full dataset. I found the accuracy, precision, recall, and AUC (as well as the confusion matrix) to assess the model's performance.

c. The model had an accuracy of 68.5% and an AUC of 0.6856. The precision was 70.8% and the recall was 63.9%. The confusion matrix also shows more false negatives than false positives (2279 > 1664).



		Predicted	
		0	1
Actual	0	4556	1664
	1	2279	4030



```
PREDICTING HIGH AND LOW PAY:
Betas: [[ 0.15060197 -0.00058965 -0.00432856  0.00390879  0.02884443]]
Accuracy = 68.5%
Precision = 70.8%
Recall    = 63.9%
Area Under Curve = 0.6856229476698169
```

d. The model performed better than chance; however, it did not have very high performance metrics. The AUC and accuracy were both 0.685, which is higher than chance (0.5) but still much lower than a perfect accuracy or AUC score of 1.

I suspect this is because some of the predictors used (e.g. height) likely have little to do with high vs. low pay. The model also had higher precision than recall, which suggests that there were fewer false positives than false negatives (as confirmed by the confusion matrix). While both precision and recall were also above 50%, they again were not close to a perfect score of 100%. For this model, the higher false negative rate means that fewer observations were classified as having high pay and that the model was more likely to incorrectly classify someone as having a low pay. Overall, the model was able to predict high vs low pay from the predictors decently well, but there is definite room for improvement.

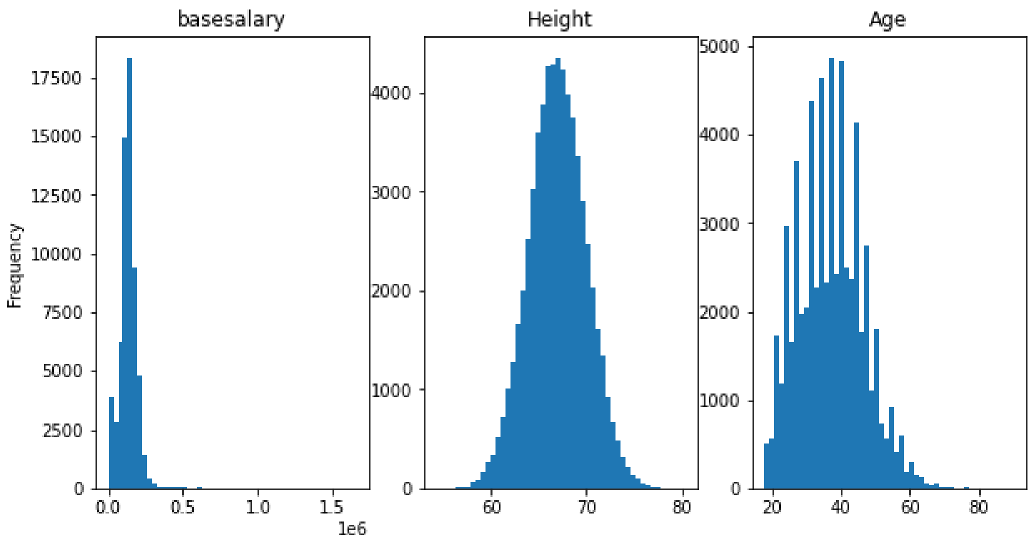
Extra credit:

A. **Is salary, height or age normally distributed? Does this surprise you? Why or why not?**

   a. For this question, I plotted three histograms for the variables 'basesalary,' 'Height,' and 'Age' respectively. This was done on the original data set before any normalization or cleaning took place. I used 50 bins in each histogram. I then looked at the distributions to determine whether or not the variables were normally distributed.

   b. I chose to plot histograms because they're useful in showing the underlying distribution of the variables, which allows me to visually inspect whether or not the variables follow a normal distribution. I used the data before any processing took place because I wanted to ensure that any normalization or cleaning (e..g dropping rows with missing values) didn't affect the distributions of the variables. I used 50 bins because, after testing several options, I found these reflected the shape of the distributions well.

   c. Height appears to follow a bell curve, with a mean of about 67 and symmetrical tails around this value. However, salary does not appear normally distributed. Instead, there is a very tight curve around ~$140,000 before it quickly tapers off on either side. Additionally, the tails are not symmetrical around the peak of the distribution, with lower salaries appearing with a greater frequency than higher ones. Lastly, Age does appear to follow a curve; however, the tails are also not symmetrical around the mean. There are no values before about 20 years old, while the extreme opposite values continue until almost 80 years old. This causes the distribution to appear right-skewed.



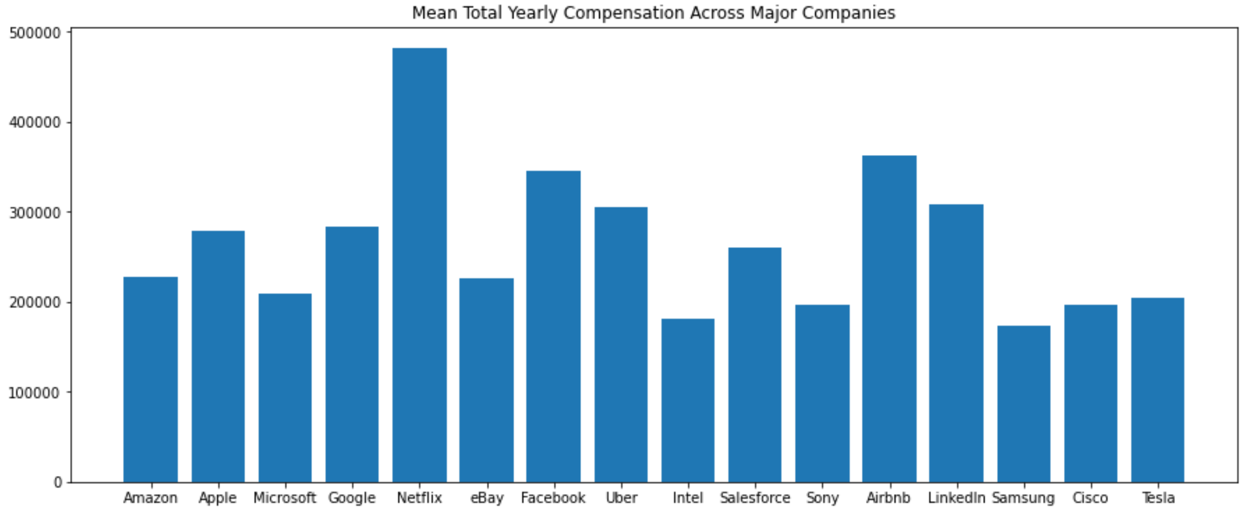
d. Based on my results, Height appears to be the only variable that is normally distributed. This doesn't surprise me since height tends to follow a normal distribution in the population at large. I would expect Age to also follow a normal distribution (as it is normally distributed in the population as a whole); however, I'm not surprised that this is not the case when considering tech employees - as there would be no values below the legal working age. I also expected salary to have a much tighter distribution, and not necessarily be normally distributed as companies compete with one another and would likely offer a narrow range of possible salaries with fewer high salaries and a greater frequency of low salaries.

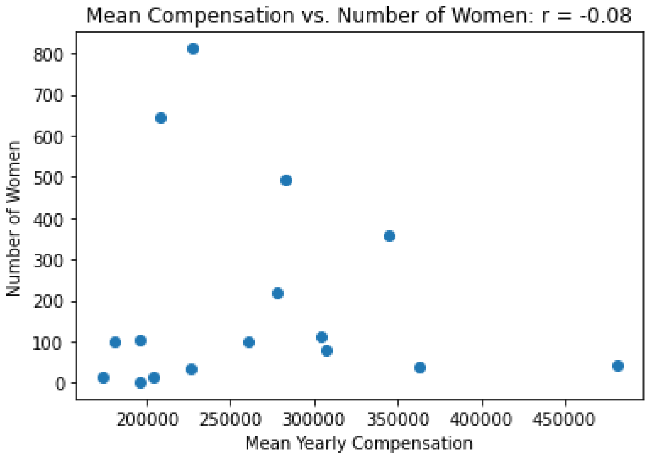
B. Tell us something interesting about this dataset that is not already covered by the questions above and that is not obvious.

a. I calculated the mean total yearly compensation across sixteen companies and plotted them on a bar chart. I then found the companies with the minimum and maximum average compensation out of these sixteen and performed a t-test for independent samples to see whether or not compensation differed with statistical significance. As a separate exploration, I found the total number of women in these sixteen companies and found the Pearson's correlation coefficient between the number of women and average compensation. I also plotted a scatter plot to examine this.

b. I was curious to see what some of the most recognizable companies were providing, on average, in terms of total yearly compensation. I used a t-test for independent samples to see whether or not the extreme values differed significantly because it allowed me to see whether or not mean values of two independent samples differ from the null hypothesis of no difference (i.e. $\mu1 = \mu2$). Lastly, I found the number of women in companies because I was curious to see whether or not companies with fewer women, or more men, provided different compensation on average from companies with more women. The Pearson's correlation coefficient allowed me to see if there was a strong linear relationship between the two variables.

c. I found that Netflix provided the highest average yearly compensation ($481,376.87) and Samsung provided the lowest ($173,279.66). They differed significantly, with a t statistic of 23.57 and a p value < 0.001. There was also no strong linear relationship between the number of women at a company and the company's average yearly compensation (r = -0.08).





d. My results show that, out of sixteen recognizable companies, Netflix provided the highest average annual compensation while Samsung provided the lowest. This difference was statistically significant (p < 0.001). Additionally, the average total yearly compensation offered by companies does not appear to be explained by the total number of women respondents from these companies - suggesting that there is not a large gender bias in how men vs. women are compensated in these companies.