Project Report

# GitHub URL

https://github.com/CorinnaDalligan/UCDPA\_CorinnaDalligan

# Abstract

My Final Submission is a study of tech salaries from multinational companies based in Ireland. My analysis was completed on salary data scraped from Levels.fyi merged with stock data imported from Yahoo! Finance. For Machine Learning - I tried to predict stock grants as part of a TCOMP package.

# Introduction

In recent years – many multinational technology companies established their European bases in Ireland due to its low corporation tax incentives and proximity to mainland Europe. I was fortunate to be employed in one such company for the past two decades with a relatively good compensation package. I was given several amazing opportunities over this time and never felt the desire to leave the company.

However, with the current worldwide economic downturn – my company has announced that will be reducing its workforce and I have found myself in an unfamiliar position of being offered a voluntary severance package. With my future in the company unclear, I chose to complete a study of TCOMP packages offered by multinational companies based in Ireland for my final submission.

With multi-nationals – the base salary offered is usually a lot smaller than the final total compensation package. There are many incentives such as bonus’s, stock grants etc which can accumulate to a significantly higher Total Compensation package. I pulled in Salary data from Levels.fyi and then merged this with stock data from Yahoo finance and using the combined data – I hoped to predict stock grants from Base Salaries offered.

# Dataset

Levels.fyi was established in 2017 to help people compare career levels across tech companies, making it the perfect datasource for me to study salary data. They believe in salary transparency and users can contribute their verified salary details to ensure data accuracy. Levels.fyi offer their salary data from a single json file, which allowed me to easily scrape this using the python requests library.   
My initial salary dataset consisted of 62642 rows and the following 17 Columns which I later filtered down as required.

I also wanted to incorporate up to date stock price data into my project. I chose the Yahoo! Finance as my datasource which was imported from the API using the pandas-datareader library.

My initial Stock dataset consisted of 1 row of data and 150 columns. I transposed and filtered these down as required for my analysis. I then ended up with a dataframe of 150 rows and three columns.

# Implementation Process

1. **Install Libraries**

I first installed required python libraries (mathplotlib and pandas-datareader) using the pip install command.

1. **Import Libraries**

I then imported all required libraries for my project. I use a combination of the following:

Text, letter

Description automatically generated

1. **Import Data**

I used two different techniques to import data from my datasources: Levels.fyi and Yahoo Finance!

Levels.fyi provide their salary data from a single endpoint:

https://www.levels.fyi/js/salaryData.json

I used requests.get to scrape the salaryData.json data and then converted the response object to JSON format using json().

I loaded this into a pandas dataframe using pd.dataframe()

Text

Description automatically generated

I compiled a list of Stock tickers for the multinationals for my project. I created a python list of tickers which I passed into the pandas-datareader package to read data from the Yahoo Finance! API and stored in a pandas dataframe.

Chart

Description automatically generated with medium confidence

1. **Inspect the dataframes**

I used head() to confirm the data had imported ok from my datasources. I had 62642 rows and 17 columns in my Levels.fyi dataframe. My stock dataframe was very different with one row and 150 columns.

I used dtypes to check my column datatypes. All columns in my Levels.fyi dataframe were objects so required conversion for my data analysis to work correctly. My stock dataframe was a combination of Objects (ticker symbol), Float (stock price) and int64 (stock volume) which looked correct.

I examined the dataframes columns and structure more closely using info().

Table

Description automatically generated with medium confidence

1. **Examine the data**

I used the unique() function to identify anomalies in yearsatcompany, yearsofexperience, gender and title columns

I found values of 90 for yearsatcompany and ‘Title: Senior Software Engineer’ in the gender column requiring some cleaning up later.

Titles looked fine but I could see some values that were not relevant for my project so these rows could get dropped.

I checked for N/A values using df.isna().sum and Null values using df.isnull().sum(). All columns returned 0’s for both indicating there were none that required cleaning.

I checked for duplicates using df.duplicated().sum(). Occasionally a duplicate row would get pulled into my dataframe after importing so I decided to add code to remove the duplicate row.

1. **Cleaning the data**

I dropped the ‘otherdetails’ and ‘dmaid’ columns permanently using drop().

I replaced the invalid Gender and tag values with the value ‘NotDisclosed’ using .replace()

I then converted some of my object datatypes to numeric values using .astype(float) and astype(int).

I removed any rows of data with yearatcompany > 50 or yearsofexperience > 50 as it is unlikely that these rows are accurate.

Some company names were entered in mixed case sensitivity. I converted Company names to uppercase using upper() and stripped trailing spaces using strip() so that they would be treated as the same company during the analysis phase.

I converted my timestamp column to datetime using the to\_datetime function and dropped time portion using .dt.date

I created an index on rownumber as each of these were unique values.

I then renamed my columns for readability using df.columns.

I identified a high standard deviate for TCOMP, BaseSalary, StockGrant and Bonus using describe so decided to focus my analysis on the 5th to 95th percentile.

I could also see that the min value for BaseSalary was 0 and removed these rows from my dataframe.

Text

Description automatically generated

1. **Filtering the Data**

Salary data varies greatly between countries, so I only wanted to look at values relevant for my project.

I filtered my Levels.fyi dataframe for rows with Ireland in the Location column and then generated a list of companies.

A screenshot of a computer

Description automatically generated with medium confidence

**8**. **Examining the Data**

With my dataframe now structured as desired and restricted to relevant data I used the pandas pivot\_table function to do some initial analysis such as identifying the top Companies for average TCOMP, top companies for average StockGrant, top job titles for average salary, top companies for retaining staff

I also sorted my dataframe by the BaseSalary Column to see who the top earner was. I was interested to see which company they were from, what job title they had, what sex they identified as and what was their Base Salary.

I also used the msno package to visualize any missing numbers but did not any.

9. **Visualizing the Data**

I created several visualizations for my project using matplotlib and seaborn.

I first sorted my dataframe by Average TCOMP and then used matplotlib to create a basic bar chart of TCOMP mean by Company.

My second visualization was using seaborn. I was interested to see how Base Salary compared by gender within companies in Ireland.

My third visualization was a scatter plot of TCOMP versus years of experience, split again by gender.

Lastly, I created a pairplot of YearsOfExperience, StockGrant, Bonus, YearsOfService and BaseSalary to try and identify candidates for Machine Learning.

10. **Custom Function**

I defined a function called CorinnasPlotFunction to plot a line chart.

The function takes in parameters for x axis, y axis, line colour, linestyle, line width, marker style, and label as input.

Chart, line chart

Description automatically generated

11. **Stock Data**

I then started to work on my stock dataframe so that I could merge this with my Levels.fyi data. My stock data imported as 1 row and 150 columns from Yahoo Finance! To make this more workable, I used the transpose() function and reset the index. I also filtered the dataframe to the stock opening price value and renamed my columns for readability.

12**. Merging DataFrames**

I needed a common column in my Levels.fyi dataframe and stock dataframes so that I could merge them together.

I created a StockTicker dictionary with values for company and Ticker.

I then added a new column to my Levels.fyi dataframe called Ticker and added the ticker symbol values to it using the map function.

A picture containing graphical user interface

Description automatically generated

With a ticker column now in both my stock data and Levels.fyi dataframes, I was able to successfully merge them together.

Graphical user interface, text, application, email

Description automatically generated

Machine Learning

**Test 1**

I used Multiple Linear regression Modelling to try and predict stock grant based of Years of Experience, Years of Service, Base Salary and TCOMP.

I imported and cleaned my data and extracted the following input features as X:

Years of Experience, Years of Service, Base Salary, Bonus, Stock Grant.

feature\_cols = ['YearsOfExperience','YearsOfService','BaseSalary','Bonus','TCOMP']

x=SalaryPredictionData\_filtered[feature\_cols].values

I extracted StockGrant as y

y = SalaryPredictionData\_filtered['StockGrant'].values

I split dataset 80/20, random state 0 so results will remain the same on each run.

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=0)

I then trained the model using Linear Regression on the training set.

ml = LinearRegression()

ml.fit(x\_train, y\_train)

I predicted the test set results.

y\_pred = ml.predict(x\_test)

r2\_score evaluated my model at 100%. I plotted the results to confirm.

Unfortunately - I then realized that TCOMP must have been a derived column based on BaseSalary, Bonus and Stock Grant so this was not a valid real-world test

**Test 2**

# Using the same technique described above I tried to predict StockGrant from Years of Experience, Years of Service and Base Salary – leaving out the TCOMP column. The accuracy of the model reduced significantly to 31%.

**Test 3**

As My accuracy in test 2 was so low – I tried to improve my score by predicting the Stock Grant based on Years of Experience, Years of Service for an individual job title in an individual company in a defined location.

For this test - I re-imported my cleaned Levels.yi dataset. I further filtered my dataframe down to the company 'Amazon' in the city 'Seattle' and limited my job title to 'Software Engineer'. I performed multiple linear regression and this time my model improved slightly to 37%.

# Results

The top Company in Ireland for TCOMP is Groupon with an average of 263k

Chart, bar chart

Description automatically generated

Where gender is disclosed – Males employees earn the highest Base Salary.

Chart, bar chart

Description automatically generated

TCOMP increases per years of experience across all genders identified.

Chart, scatter chart

Description automatically generated

Calendar

Description automatically generated

Chart, line chart

Description automatically generated

Machine Learning linear regression model of stock grant based of Years of Experience, Years of Service, Base Salary and TCOMP – 100% accurate. From this analysis – I suspect that the TCOMP data column is a derived column from BaseSalary, Bonus and Stock Grant data and was skewing the model. My assumption on starting the project was that TCOMP would have factored in additional benefits such as Shift allowance or medical insurance.

Chart, line chart

Description automatically generated

Machine Learning linear regression model of StockGrant from Years of Experience, Years of Service and Base Salary with 31% accuracy.

Chart, scatter chart

Description automatically generated

Machine Learning linear regression model of StockGrant from Years of Experience, Years of Service and Base Salary for a Software Engineer working in Amazon based in Seattle – 37% accurate

Chart, scatter chart

Description automatically generated

# Insights

1. Top Company in Ireland for TCOMP is Groupon with an average of 263k.
2. Top Company in Ireland for StockGrant is MongoDB with an average of 119k.
3. Top individual earner in Ireland was a male Software Engineer in STRIPE with 7 years of experience is earning a TCOMP package of 297K.
4. Top 10 Company in Ireland for retaining staff was IBM at 10.67 years of service.
5. Software developers have the best average salary at 191k.
6. Where gender is disclosed – Female employees earn a lower TCOMP than males.
7. Predicting StockGrant using TCOMP, BaseSalary, Bonus and Years of Experience data from Levels.FYI was 100% accurate – however I then discovered that TCOMP was derived from BaseSalary, Bonus and StockGrant. It was therefor not a valid test. It did not include other perks such as shift allowance or medical insurance.
8. StockGrant is not easily predicted using from Years of Experience, Years of Service and Base Salary as input features – yielding 31% accuracy, and only 37% accuracy even if the dataset is filtered to company and location.

# References

# <https://www.levels.fyi/?compare=Google,Facebook,Microsoft&track=Software%20Engineer>

<https://towardsdatascience.com/a-beginners-guide-to-grabbing-and-analyzing-salary-data-in-python-e8c60eab186e>  
<https://www.datasciencelearner.com/yahoo-finance-api-python>  
<https://www.youtube.com/watch?v=sgndYho8RyI>  
<https://www.analyticsvidhya.com/blog/2021/12/stock-market-analysis-with-pandas-datareader-and-plotly-for-beginners/>  
<https://www.shanelynn.ie/bar-plots-in-python-using-pandas-dataframes/>  
<https://github.com/JatinSadhwani02/Employee-Salary-Predictio-in-Machine-Learning/blob/master/Salary_Prediction_ML_Model.ipynb>  
<https://www.youtube.com/watch?v=WngoqVB6cXw>  
<https://www.statology.org/scatterplot-with-regression-line-python/>  
<https://medium.com/analytics-vidhya/pairplot-visualization-16325cd725e6>  
<https://www.datasciencelearner.com/drop-unnamed-column-pandas/>