GEORGE MASON UNIVERSITY

Data Analytics Engineering

FALL 2021

Use AI to Predict Labor Markets

DAEN 690

Project Report

Sponsored By: Allwyn Corporation

Guided By: Isaac K. Gang, PhD
College of Engineering & Computing

Sahithi Reddy Godishala

Chaya Vijaya lakshmi Adari

Vaishnavi Kammalampudi

Hamza Habib

Rishi Thodupunuri Rajender

George Mason University

12/10/2021

This Page Intentionally Left Blank

Table of Contents

Abstract

With the different working conditions as the result of the pandemic, the job market, and the salary scales have continue to fluctuate. In this project, we proposed a solution to predict the salaries of different job categories/roles in the ever-emerging IT industry. Our solution accurately and efficiently predicts the nature of labor market and forecasts the salaries for the upcoming three years using the data from indeed and Glassdoor. In the analytics phase, we first matched the job description provided by the client using natural language processing (NLP) algorithms of TF_IDF, N-gram in conjunction with the cosine similarity to match the description. To further improve the efficiency, we use latent similarity model to match the category labels and applied exponential smoothing techniques to predict the trend of the next three years. Despite the limited scope of the data that was available to us, we believe that our model accuracy was acceptable considering the context of the dataset.

Keywords: Forecasting, Natural Language Processing, Artificial Intelligence, TF_IDF, N-gram, Cosine Similarity, Latent Similarity, Random regression, Linear regression, Support vector machines and Exponential Smoothening.

1 Introduction

1.1 Background

Data changes every day and demand for data in predictions and making decisions has become a very important aspect in today's world of fast-moving technology. Especially in predicting labor markets today, perhaps more than any other part of the economy, the labor market is unpredictable. Allwyn corporation, a software solutions company needs a business solution to predict future wage data to compete better for government contracts, give them the ability to accurately predict future contract expenses. Using AI to predict these labor markets is our main goal. In this paper, we will go in detail about NLP, predictions using machine learning and potential data sources used for this project.

Allwyn Corporation

Allwyn Corporation, a software solutions company, needs wage data to solve a business problem. Accurate wage data will provide Allwyn with the resources to compete better for government contracts, give them the ability to accurately predict future contract expenses, avoid being financially ineffective, and allow them to pay their employees appropriately with respect to their labor category, education, and experience. The services provided by Allwyn are in the following categories: Artificial Intelligence and Machine Learning, Data Analytics and Business Intelligence, Agile Software Delivery, Enterprise Application, IT Modernization, and Low Code Development. At present, Allwyn corporation needs assistance to help predict the labor market rates for technical roles based on labor categories in the Washington, D.C., Maryland, and Virginia area.

The smartest AI technologies are, quite literally, prediction machines. They use algorithms to analyze large sets of data, to optimize towards a goal. As they optimize, they learn over time to improve their results. A predictive model powered by AI can take the data you already have and unlock immense value from it [1]. Whether AI can be used to accurately predict labor market trends, however, is an open question. As with all models, data issues can throw estimates off track, and biases can emerge from setting algorithms to learn from historical examples. Models are also limited to the bounds of an observed period, losing predictive power the further they attempt to glimpse into the future. [2]

To predict the most accurate labor cost for individuals, we will need to look at historical data. But the recent pandemic had a significant effect on labor market metrics in the United States in the year 2020. But the most recent metrics of July month shows that job gain of 943,000 follows another month of impressive job recovery across the U.S., shining a brighter light on economic recovery despite the challenges that persist for organizations in hiring and retaining talent [3]. The study Using machine learning and/or artificial intelligence will allow us to help Allwyn predict competitive labor rates in the Washington D.C. area market using the three data sources- Occupational Employment and Wage Statistics (OEWS) by the U.S. Bureau of Labor Statistics (BLS), CALC (The Contract-Awarded Labor Category), and LinkedIn.

Impact Due to covid

The COVID-19 epidemic triggered a worldwide health catastrophe unlike any other in recent memory. The influence on the global economy and society has been profound and far-reaching. The first shock struck huge portions of the economy, putting economic activity on pause in many nations due to fear of contagion and harsh limitations on social closeness.
Employment among low-wage employees declined by 11.7 percent between February 2020 and February 2021, from 28.1 million to 24.8 million. This contrasts to a 5.4 percent drop in employment for middle-wage employees, who lost 5.5 million jobs over the time. Meanwhile, employment among high-wage earners remained stable, at little more than 28 million.

1.2 Problem Space

Allwyn desires a solution that can aid in predicting an accurate cost estimate for future project bids. The solution model should factor in current labor market rates for the Washington, D.C. Metro area, economic inflation over a 3–5-year timeframe, and other job-specific information.

Allwyn provided us an input list of labor categories (in specific locations) and the number of years for the contract. Each labor category is associated with a written job description, required education level, and required number of years' experience provided by the General Service Administration (GSA) Federal Supply Service Federal Supply Schedule Pricelist. Based on this user input, the solution will need to obtain relevant cost information from publicly available data sources --- BLS (Bureau of Labor Statistics), GSA CALC and possibly private sector sources such as salary.com and provide visualization showing minimum, maximum, median, and standard deviation. Solving this problem needs finding reliable open data sources which contains labor information, and which aligns with the input dataset given by allwyn. The main challenge for this problem is to match the job descriptions given by allwyn with the job descriptions from the retrieved datasets.

Machine learning models and Natural language processing have the best approaches in dealing with such problems. These techniques can be applied to build a model that predicts and analyzes the future labor markets and will be useful for Allwyn for winning more contracts. The datasets from Indeed and Glassdoor will be the key sources in providing a solution to this problem

1.3 Research

In our initial research we have received initial inputs from the organization. We have got different job categories and we need to match them with available sources. We need to predict different labor market rates for different categories for the next 3-4 years. Our team also need to go through different data sources that are available online to match the categories. For data that is available from all other sources we are using NLP to web scrape and extract data for the analysis.

To obtain the primary goal of the project that is to match job category data we needed to use different methods using NLP but first and foremost the main part of our research was getting an

adequate amount of historical data from a reliable source to continue our analysis. In the data collection phase, we researched different available data sources and resources which were available through our library services. First, we contacted our librarian to help us to find good quality data, and we also approached the career services of George Mason in that quest to get an adequate amount of data, but unfortunately we weren't able to acquire the data which would align with the partners requirements. We also approached Virginia Tech's library for the data related to job postings which were dated back to 10 to 15 years but we weren't able to get this much data.

To overcome the problem of data and continue with the project we scraped some available online sources like Indeed and Glassdoor for the recent job postings data and we were able to scrape around more than 10,000 rows of job postings which were related to the job categories provided by the Allwyn corporation.

As it is described earlier we were able to manage data around 10,000 records but it was not enough to fulfill our goal of predicting the salaries for provided job categories for the next 3 to 5 years. To beef up our data and take it to an adequate amount in order to get good results out of it we generated synthetic data from the existing data.

After data generation our next step was to map the job titles from Allywn's data to ours data, so in order to achieve that we tried multiple methods like N-gram vectorizer, TF-IDF and other NLP techniques. At the end we used two approaches for our model one was Cosine similarity on TF-IDF vectors and Latent semantic analysis, and after manually analyzing the models we used latent semantic analysis for further model training.

1.4 Solution Space

According to our research it is difficult to find data for the analysis. As we need to predict the labor costs for next 3-4 years it plays very important role for the company to quote the total amounts to bid any contracts. The model needs to predict the future expenses. So, our system uses the Natural Language Processing and AI/ML techniques to train data which is obtained from multiple open sources to predict the salaries for next 3-4years.

Our system delivers labor cost information and visualizations showing which job titles have the highest and lowest average salaries and which job titles have the highest average salaries among the DMV areas by performing visualizations and analysis on the data and can assist the organization with bidding on contract.

1.5 Project Objectives

Develop a cost estimate for a project based on current labor market rates and predict labor market rates for the future 3-5 years. Take a data-driven approach to negotiate contracts by leveraging insights gleaned from internal and external data sources and using an automation AI driven "continuous learning" approach for smarter and faster decision making.

Our project objectives incorporate getting significant information sources that have the positions, necessities of an agreement, compensation of the situations concerning the given area (Washington D.C metro region).

1.6 Primary User Stories

Based on the information and user cases provided by allwyn corporation, we have created the following user stories to guide our project:

- "As a team, we want to build a cost estimate model to predict salaries for next 3-4 years for all the 10 job titles."

- "As a Corporation, I want to know the current labor market rate so that I can accurately estimate cost to pay my employees."

- "As a Corporation, I want to know how the labor market rate are for next 3-5 years so that I can bid my contracts with correct marker rates."

1.7 Product Vision - Sample scenarios

Scenario #1:

Our system can predict the following 3-4 years salaries forecast to give an expected expense needed to execute a contract, which helps organizations with bidding for government contracts effectively. Unlike the job titles in the datasets, our framework can show more explicit job titles and classifications, to assist clients with tracking down more exact expenses.

Scenario #2:

Our system can predict the following 3-4 years salaries forecast to give an expected expense needed to execute a contract, will help in better decision making for contracts.It can influence business outcomes. Users will be able to see detailed level information on how the estimate was made.

1.8 Definition of Terms:

Bidding: It is an offer made by an investor, trader, or dealer to buy an asset or compete for a contract.[4]

Contract: It can be defined as a promise enforceable by law. The promise may be to do something or to refrain from doing something. It requires the mutual asset of two or more persons. One of them making an offer and another accepting.[5]

NLP: Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.[6]

Machine Learning: Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.[7]

Artificial Intelligence: Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, NLP, speech recognition and machine vision.[8]

2 Data Acquisition

2.1 Overview:

The data sources that we are using are open-source dataset retrieved from Kaggle and a dataset retrieved by web scraping from indeed and the dataset given by allwyn corporation.
The Kaggle dataset is retrieved from glass door. It has 16 columns and around 2500 rows after filtering. In addition to this dataset, we used web scraping to retrieve the dataset from indeed.com. For this we have used python packages like pandas and beautiful soup. We will have to leverage these three datasets to match with the job categories given by Allwyn corporation and predict the labor markets.

2.2 Field Descriptions:

Kaggle
Dataset Owner: Kaggle
Dataset Type: Open Source
Dataset Size: 5 MB
Dataset Location: Internet
Dataset access: https://www.kaggle.com/josephgutstadt/data-jobs
Dataset Restrictions: Requires Training
Dataset Time range: 2011 - 2021
Dataset Collection Process: Downloaded from kaggle
Analytic/Algorithm that will use dataset: “In progress”

Indeed
Dataset Type: Open Source
Dataset Size: 15 MB
Dataset License: 2011 -2021
Dataset Location: Internet
Dataset access: https://www.indeed.com
Dataset Restrictions: Requires Training
Dataset Time range: Current date
Dataset Collection Process: Web scrapped from indeed
Analytic/Algorithm that will use dataset:” InProgress”

2.3 Data Context

Indeed:

The data scrapped from Indeed contains information about a job posting for the 10 labor categories given by the Allwyn Corporation from different companies in the DVM area. There are 5 attributes in this table like Title, Salary, Company Name, Location and Description.

In the Title, each category is labeled differently like for cloud engineer it is labeled as Senior cloud engineer, or AWS cloud engineer, or Devops Cloud engineer etc.
The salary column is in ranges for all the labor categories. In this attribute few of the ranges are in years, months and per hour.
Our dataset mainly focuses on the DMV area. The data scrapped is from 25 miles radius of the DMV area.
In Description Attribute, it specifies the skills and requirement from a candidate the company is looking for. In this, they provide various skill which a candidate must know and what all task he/she will be assigned, or they might have to do.

Attributes	**Description**
Title	Title of Labor Categories provided by Allwyn Corporation
Salary	Expected salary or wage rate provided by a particular company
Company Name	Name of the company
Location	Location where the job posting is available
Description	Describe task, related duties and responsibilities for a position

Glassdoor:

This data was downloaded from Kaggle website. There are around 16 attribute and 26981 records. Few of the attributes are explained below
The attribute “rating” is the rating of the company which is given by the employees who are currently working or used to work. This attribute let other people know how the company treats their employees. The rating is given from 0 to 5.
The “Headquarters” attributes tell the location where the company was founded or where most of the dealing or works is done.
The attribute “Type of owner”, indicate the type of owner of that company. There are various type like Non-Profit organization, Government, Public company, private company, franchise, self-employed etc.
The attribute “Industry”, let the applicant know the type of industry the company focus on like Health Care, Internet, Investment Company, Gas and oil, Tv Broadcast, Banking, Consultant, etc.
Revenue attributes show the revenue of the company. It mainly focus on the profit the company gained in a year. This ranges from millions to billions depending on the size and profits of the company.
Easy Apply attribute show the candidate whether is it easy to apply on Glassdoor or will the job posting redirect to the company webpage where the candidate has to apply. It is indicated by True or -1.

Attributes	**Description**
Job Title	Different types of labor categories from glassdoor
Salary Estimate	Expected salary or wage rate provided by the company
Job Description	Describe task, related duties and responsibilities for a position
Rating	Rating of the company from 0 to 5
Company Name	Name of the company
Location	Location where the job posting is available

Headquarters	The main headquarters of that company
Size	Number of employees working throughout the company
Founded	The year the company was founded
Type of owner	Indicates the type of owner
Industry	Type of industry the company focus on
Sector	Type of sector the company focus on
Revenue	The total revenue the company generates
Competitors	Who are the competitors for that company
Easy Apply	Is it easy to apply on glass door or not

2.4 Data Conditioning:

Indeed:

The data was scrapped from indeed using libraries like beautiful soup and pandas. 10 different URLs were used to scrape data for 10 different job titles which gave 10 different csv files.
All the 10 csv files are merged into one single file.
In the title column there was unwanted character like “new” in the beginning of job title. We trim that and keep only the job title.
In salary column, we have removed all the null values and all the unwanted characters like ‘$’, ‘per hour’, ‘per month’ etc. The salary was in ranges, so we split the ranges into two column and took the mean of two salary columns and stored it in 3rd column. We have salary column in which few were in months and hours, so we converted all of them to years.
We have stored that data in three dataframes and removed two salary ranges from each so there will be only salary_1 in dataframe 1, salary_2 in dataframe_2 and salary_3 in dataframe_3.
Here we have added a column called Experience and have given random experience based on the salary. For low range of salary, we have given 3 to 5 years of experience, for mean range salary we have given 6 to 8 years of experience and for high range of salary we have given 9 to 12 years of experience.

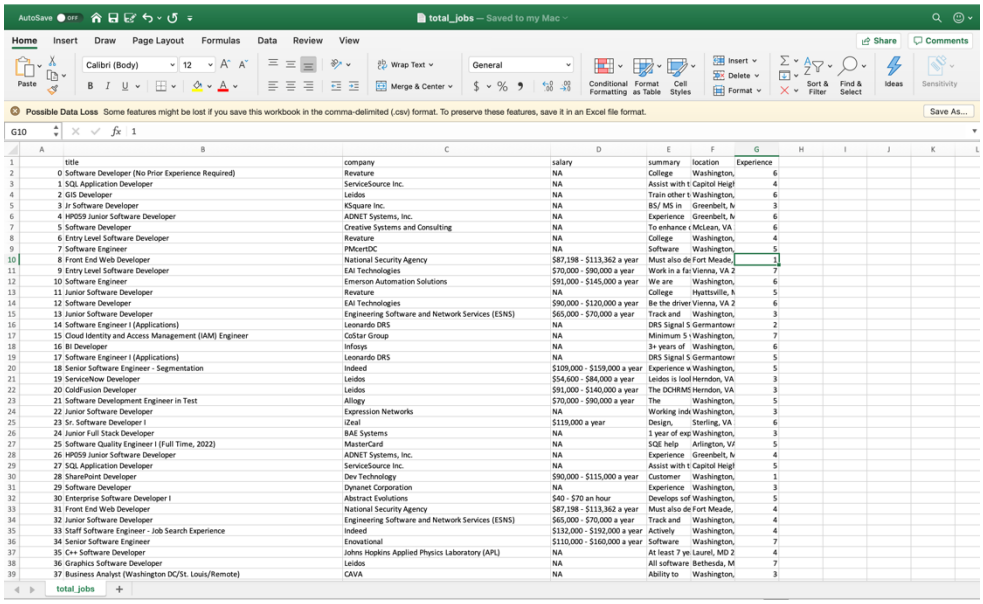
Glassdoor:

The dataset we got from Kaggle doesn’t contains any null values. There were many job titles which were unwanted for this project. So, we filtered all the unwanted job titles, and we only kept the job titles which were given to us by Allwyn Corporation.
We removed all the unwanted columns like rating, size, founder etc. which were not useful for this project and renamed few columns like job description to description, salary estimate to salary, etc so we can match it to the other dataset.
In salary column, we have removed all the unwanted characters like ‘$’, ‘per hour’, ‘per month’ etc. The salary was in ranges, so we split the ranges into two column and took the mean of two salary columns and stored it in 3rd column. We have salary column in which few were in months and hours, so we converted all of them to years.

We have stored that data in three dataframes and removed two salary ranges from each so there will be only salary_1 in dataframe 1, salary_2 in dataframe_2 and salary_3 in dataframe_3. Here same as Indeed data, we have added a column called Experience and have given random experience based on the salary. For low range of salary, we have given 3 to 5 years of experience, for mean range salary we have given 6 to 8 years of experience and for high range of salary we have given 9 to 12 years of experience.
After preprocessing Indeed and Glassdoor data we got 6 dataframes. We combined all the dataframes based on their column names into one dataframe.

Final Dataset:

We have used pandas and NumPy library. We merged all the indeed files to one file, read the new file into a data frame and removed all the 'new' from title. Then we removed unknown column from that data frame, added experience column with random value for now.
For Kaggle data we removed all unwanted columns and filtered out all the other job titles. then we removed 'Glassdoor est.' from salary column and added experience column. Then we concated both indeed data frame and Kaggle data frame into one data frame and saved the file.



	title	company	salary	summary	location	Experience
0	Software Developer (No Prior Experience Required)	Revature	NA	College	Washington,	6
1	SQL Application Developer	ServiceSource Inc.	NA	Assist with t	Capitol Heigl	4
2	GIS Developer	Leidos	NA	Train other t	Washington,	6
3	Jr Software Developer	KSquare Inc.	NA	BS/ MS in	Greenbelt, M	3
4	HPOS9 Junior Software Developer	ADNET Systems, Inc.	NA	Experience	Greenbelt, M	6
5	Software Developer	Creative Systems and Consulting	NA	To enhance o	McLean, VA	6
6	Entry Level Software Developer	Revature	NA	College	Washington,	4
7	Software Engineer	PMCertDC	NA	Software	Washington,	5
8	Front End Web Developer	National Security Agency	$87,198 - $113,362 a year	Must also de	Fort Meade,	1
9	Entry Level Software Developer	EAI Technologies	$70,000 - $90,000 a year	Work in a fa	Vienna, VA 2	7
10	Software Engineer	Emerson Automation Solutions	$91,000 - $145,000 a year	We are	Washington,	6
11	Junior Software Developer	Revature	NA	College	Hyattsville, M	5
12	Software Developer	EAI Technologies	$90,000 - $120,000 a year	Be the driver	Vienna, VA 2	6
13	Junior Software Developer	Engineering Software and Network Services (ESNS)	$65,000 - $70,000 a year	Track and	Washington,	3
14	Software Engineer I (Applications)	Leonardo DRS	NA	DRS Signal S	Germantowr	2
15	Cloud Identity and Access Management (IAM) Engineer	CoStar Group	NA	Minimum 5	Washington,	7
16	BI Developer	Infosys	NA	3+ years of	Washington,	6
17	Software Engineer I (Applications)	Leonardo DRS	NA	DRS Signal S	Germantowr	5
18	Senior Software Engineer - Segmentation	Indeed	$109,000 - $159,000 a year	Experience w	Washington,	5
19	ServiceNow Developer	Leidos	$54,600 - $84,000 a year	Leidos is loo	Herndon, VA	3
20	ColdFusion Developer	Leidos	$91,000 - $140,000 a year	The DCHRMS	Herndon, VA	3
21	Software Development Engineer in Test	Allogy	$70,000 - $90,000 a year	The	Washington,	5
22	Junior Software Developer	Expression Networks	NA	Working indi	Washington,	3
23	Sr. Software Developer I	iZeal	$119,000 a year	Design,	Sterling, VA	6
24	Junior Full Stack Developer	BAE Systems	NA	1 year of exp	Washington,	3
25	Software Quality Engineer I (Full Time, 2022)	MasterCard	NA	SQE help	Arlington, VA	5
26	HPOS9 Junior Software Developer	ADNET Systems, Inc.	NA	Experience	Greenbelt, M	4
27	SQL Application Developer	ServiceSource Inc.	NA	Assist with t	Capitol Heigl	5
28	SharePoint Developer	Dev Technology	$90,000 - $115,000 a year	Customer	Washington,	1
29	Software Developer	Dynanet Corporation	NA	Experience	Washington,	3
30	Enterprise Software Developer I	Abstract Evolutions	$40 - $70 an hour	Develops sof	Washington,	5
31	Front End Web Developer	National Security Agency	$87,198 - $113,362 a year	Must also de	Fort Meade,	4
32	Junior Software Developer	Engineering Software and Network Services (ESNS)	$65,000 - $70,000 a year	Track and	Washington,	4
33	Staff Software Engineer - Job Search Experience	Indeed	$132,000 - $192,000 a year	Actively	Washington,	4
34	Senior Software Engineer	Enovational	$110,000 - $160,000 a year	Software	Washington,	7
35	C++ Software Developer	Johns Hopkins Applied Physics Laboratory (APL)	NA	At least 7 ye	Laurel, MD 2	4
36	Graphics Software Developer	Leidos	NA	All software	Bethesda, M	7
37	Business Analyst (Washington DC/St. Louis/Remote)	CAVA	NA	Ability to	Washington,	3

Figure 1 screenshot of final dataset

Analyzing Trends in the Dataset:

We have created some visualizations for the dataset that we have web scrapped from Kaggle and indeed to analyze some trends in the dataset before proceeding with analytics and modelling.

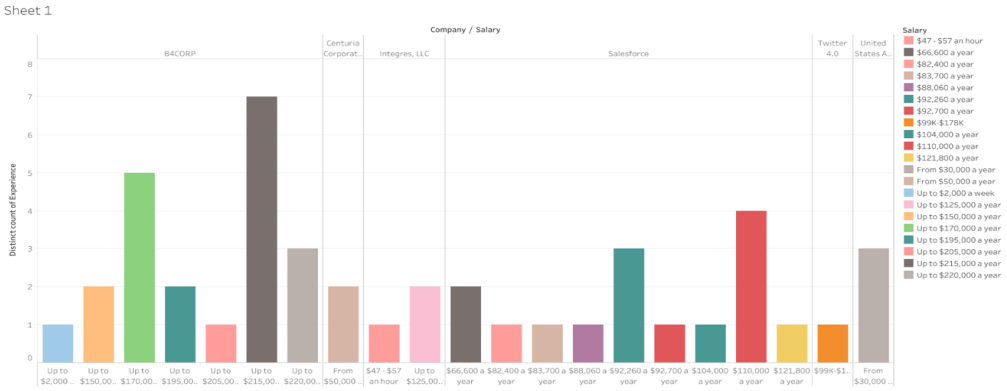


Figure 2 Salaries for employees wrt to years of experience.

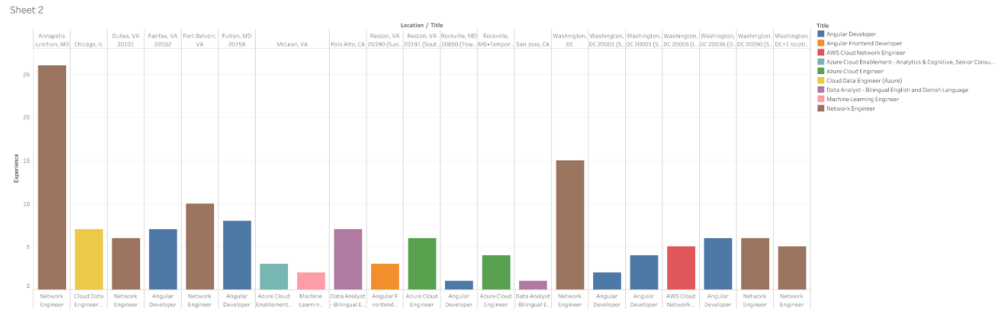


Figure 3 Experience of the employees with respect to their title and location

2.5 Data Quality Assessment:

Understanding the data is a crucial part for this analysis. After the pre-processing of data, it is easier to understand the data.

BLS

Data Quality	Assessment
Completeness	Data set is complete apart from some missing values in wage and different percentiles of wage.
Consistency	We have data from 2011 to 2020 and apart from 2019, the data is consistent apart from few nulls.

Uniqueness	Our main focus is on Washington metropolitan area and the Job titles are also specific to IT related fields.
Integrity	This is a public source and is controlled by the bureau of labor statistics so the data is credible.
Conformity	The data is according to the requirements except the job titles column which we will transform it as the project goes along.
Accuracy	Apart from some missing values and Null values most of the data is accurate.
Quality	BLS has very efficient data collection mechanism, so the quality is up to the mark.

Indeed

Data Quality	Assessment
Completeness	Data set is complete apart from some missing values in salaries and years of experience
Consistency	We have data which is consistent apart from null values.
Uniqueness	Our main focus is on Washington metropolitan area and the Job titles are also specific to IT related fields.
Integrity	This is a public source and which is scrapped from the website using python libraries.
Conformity	The data is according to the requirements, but have few missing values, which we will transform it as the project goes along using feature engineering.
Accuracy	Apart from some missing values and Null values, data is accurate.
Quality	Indeed has very efficient data collection mechanism, so the quality is up to the mark.

Kaggle

Data Quality	Assessment
Completeness	Data set is complete apart from some missing values in salaries, years of experience and unwanted data.
Consistency	We have data that is consistent apart from null values.
Uniqueness	Our main focus is on Washington metropolitan area and the Job titles are also specific to IT related fields.
Integrity	This is a public source and which was downloaded from kaggle website..
Conformity	The data is according to the requirements, but have few missing values and unwanted data, which we will transform it as the project goes along and remove the unwanted data.
Accuracy	Apart from some missing values and Null values, data is accurate.
Quality	kaggle has very efficient data collection mechanism, so the quality is good..

2.6 Other Data Sources:

The GSA CALC TOOL:
The General Services Administrative (GSA) has a Contract Awarded Labor Category (CALC) tool through which we can observe recent labor market rates by category for around 50,000 categories. GSA tool searches through all the categories and gives the price of each job category. The tool has filters wherein we give input like education level, experience, location and labor category to retrieve the prices of each contract [10].

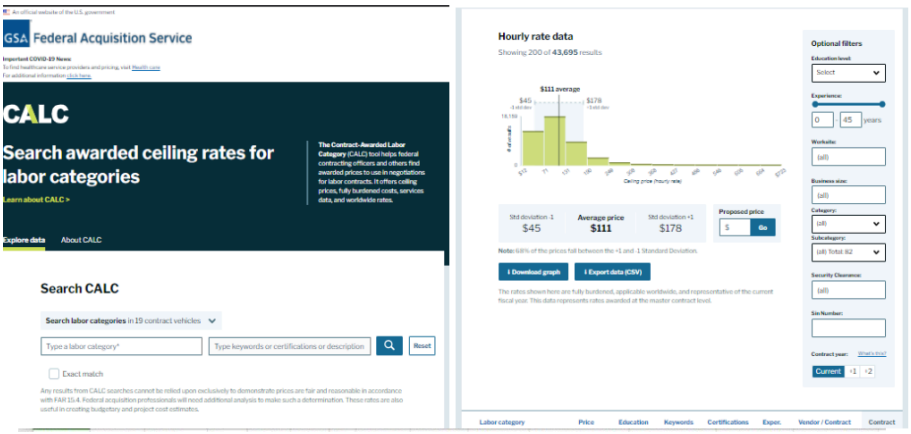


Figure 4 The GSA CALC TOOL data source

BLS:

The Occupational Employment and Wage Statistics (OEWS) program conducts a semiannual survey designed to produce estimates of employment and wages for specific occupations. The OEWS program collects data on wage and salary workers in nonfarm establishments in order to produce employment and wage estimates for about 800 occupations. Data from self-employed persons are not collected and are not included in the estimates. The OEWS program produces these occupational estimates for the nation as a whole, by state, by metropolitan or nonmetropolitan area, and by industry or ownership. The Bureau of Labor Statistics produces occupational employment and wage estimates for approximately 415 industry classifications at the national level. The industry classifications correspond to the sector, 3-, 4-, and selected 5- and 6-digit North American Industry Classification System (NAICS) industrial groups [13].

3 Algori (OECDI Library, n.d.) (Kochhar, n.d.)thms & Analytics

The below are the models that we are planning to use for this project. We are going to use NLP to build models to match the job descriptions and use the algorithms to match the labor wages and predict for those job titles.

3.1 Building Models using NLP

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software [6]. Our main goal is to determine the similarities between Allwyn's labor category descriptions and job category descriptions from the dataset that we have retrieved by web scrapping. Allwyn corporation have provided us a document with labor categories and their descriptions for each labor category. The dataset that we have web scrapped from indeed and Kaggle have different labor categories and their job descriptions. We built models to match these descriptions and to accomplish precise job titles from external data source by perceiving how well their job descriptions match the descriptions of the job titles in Allwyn's document. After matching the descriptions our goal to predict future wages for these job titles.

3.2 Preprocessing

We merged all the data from indeed dataset into one data frame and downloaded the csv file. Then read that new csv file and removed the characters 'new' from the title column. Then we removed all the unwanted characters from the column salary like $, a year, an hour, a week, etc. strip the salaries into two columns as salary1, salary2 based on '-'.
As the salary was object datatype, converted to an integer. Based on the salary pattern, we have converted all the salaries which were per hour, per month, per week to per year for both the salary column. Deleted the main salary column which was in range. After that we have copied that data frame in two different data frames indeed1, indeed2. Deleted salary2 column in and salary1 column in indeed2. Added experience from 1 to 5 for indeed1 data frame and experience from 6 to 11 for indeed2 data frame. As most of the values are missing in the salary column for both the data frame, I have taken minimum and maximum for that particular data frame for salaries and given a range between the minimum and maximum for missing values. Renamed salary1, salary2 columns in both the data frame to salary.

For the Glassdoor dataset, we have filtered out all the unwanted rows in excel. Read that file in python and removed all the unwanted columns. Renamed the column names and shown in indeed data. Removed unwanted characters from salary column like Glassdoor est, Employ, a year, etc. Then split the salary column which is in range into two columns which are salary1 and salary2. Again removed the unwanted characters from both the salary columns like k, $. Multiplied all the salaries row with 1000 to convert them to year salary like100k to 100000. Deleted the main salary column and copied that data frame into two different data frames glassdoor1 and glassdoor2. Removed salary2 column from Glassdoor 1 and salary1 column from glassdoor2. Added experience from 1 to 5 for

glassdoor1 data frame and experience from 6 to 11 for glassdoor2 data frame. Renamed the salary1, salary2 column in both the data frame to salary. Concat all the 4 data frames indeed1, indeed2, glassdoor1, glassdoor2 into one data frame total_data. Added a column posted_date and gave random date values from 2017 to 2019. Downloaded that data frame to csv file.

3.2 NLP Modelling:

Term Frequency–Inverse Document Frequency (**TF-IDF**):

TF-IDF stands for **"Term Frequency — Inverse Document Frequency"**. This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining [11].
IDF is calculated as follows where t is the term(word), n is no. of documents,

$$idf\,(t, D) = log\,(\ \frac{N}{count\,(d \in D : t \in d)}\ )$$

Scikit-Learn

- $\text{IDF}(t) = \log\frac{1+n}{1+\text{df}(t)} + 1$

Standard notation

- $\text{IDF}(t) = \log\frac{n}{\text{df}(t)}$

Putting it together, TF-IDF formula is mentioned below.
Putting it together:

$$tf\,idf\,(t, d, D) = tf\,(t, d)\ .\ idf\,(t, D)$$

We have first dropped all the null values and using the stopwords library from NLTK we have removed all the stopwords from the dataset. We have also looked for any punctuation marks that are present in the dataset and cleaned them. We have done tokenization to the description column.

Using tokenization, we have splitted the strings into list of words. Then we have done lemmatization that is reducing the word to its root form. It helps in reducing the corpus of the words we include in the model. We have also created the dictionary of the important words from Allwyn dataset. We have done lemmatization on the dataset. The words in the dataset are converted into vectors to measure the weight of the word. The weight of a word is determined by the number of documents it appears in, multiplied by its frequency. Originality is measured by calculating the frequency of the words (TF) divided by the documents the words appear in, for each word, multiply by their frequency. We have calculated TF IDF, how often the word occurs in the document.

Out[67]:

	across	actionable	additionally	agile	algorithm	also	amazon	analysis	analyst	...	unit	update	usability	used	violat
0	0.062067	0.000000	0.142711	0.000000	0.285421	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
1	0.062067	0.000000	0.142711	0.000000	0.285421	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
2	0.050132	0.000000	0.000000	0.000000	0.000000	0.000000	0.115267	0.000000	0.089659	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
3	0.036210	0.097939	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.064760	0.000000	...	0.000000	0.072840	0.000000	0.000000	0.097
4	0.054464	0.000000	0.000000	0.147312	0.000000	0.000000	0.125229	0.147312	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
5	0.033969	0.000000	0.000000	0.000000	0.000000	0.091877	0.000000	0.000000	0.060752	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
6	0.066005	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.178528	...	0.000000	0.000000	0.178528	0.000000	0.000
7	0.066549	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.459048	0.133871	0.000000	0.153016	0.000
8	0.066549	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.459048	0.133871	0.000000	0.153016	0.000
9	0.041467	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.148326	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000

10 rows × 235 columns

In [68]:
```
X_tfidf_df.drop([''], axis=1)
```

Out[68]:

	across	actionable	additionally	agile	algorithm	also	amazon	analysis	analyst	analytic	...	unit	update	usability	used	violat
0	0.000000	0.142711	0.000000	0.285421	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
1	0.000000	0.142711	0.000000	0.285421	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.115267	0.000000	0.089659	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
3	0.097939	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.064760	0.000000	0.000000	...	0.000000	0.072840	0.000000	0.000000	0.097
4	0.000000	0.000000	0.147312	0.000000	0.000000	0.125229	0.147312	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000
5	0.000000	0.000000	0.000000	0.000000	0.091877	0.000000	0.000000	0.060752	0.000000	0.091877	...	0.000000	0.000000	0.000000	0.000000	0.000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.178528	0.000000	...	0.000000	0.000000	0.178528	0.000000	0.000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.459048	0.133871	0.000000	0.153016	0.000
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.459048	0.133871	0.000000	0.153016	0.000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.148326	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000

10 rows × 234 columns

Figure 5 TF-IDF Output

After preprocessing both datasets then we created TF-IDF vectors for allwyn dataset then after that using those vectors transformed new vectors for our dataset. Finally, we had two datasets of vectors of both the datasets, we applied the cosine similarity on both datasets of vectors and took the highest similarity value and gave the corresponding title from allwyn dataset to our datasets

Latent semantic Analysis:

It is majorly used for finding reoccurring of topics across documents. It involves creating structured data from a collection of unstructured texts. It is mostly implemented in the areas where there is a need for dimension reduction or noise reduction. (Navlani, 2018)
Below image shows comparison between text classification and topic modelling.

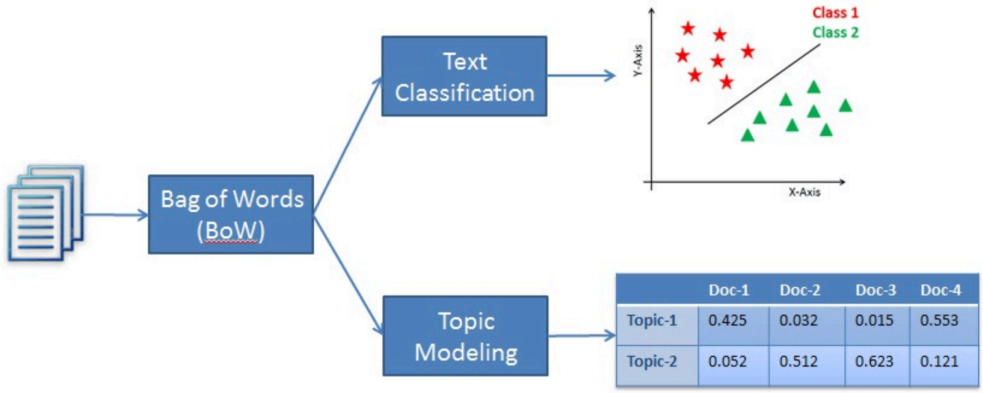


Figure 6 Comparison between text classification and topic modelling

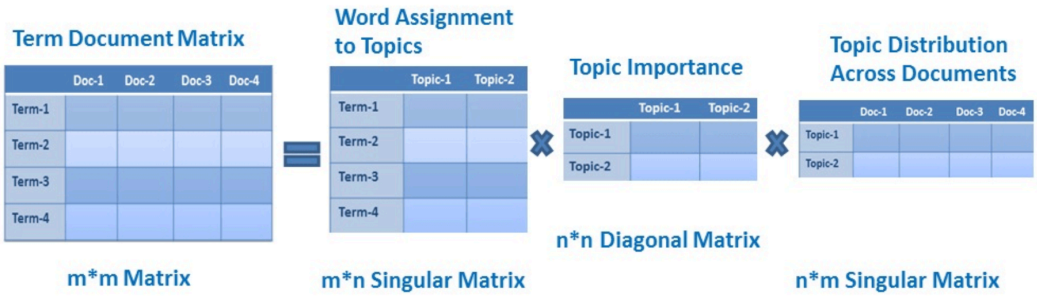
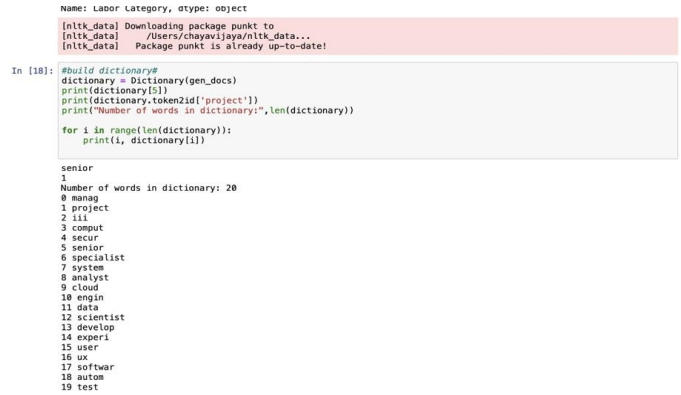


Figure 7 LSA workflow

We have used LSI to measure the similarity between the descriptions. It can transfer Vectors to diff patterns, and it can also establish relationship between words and topics. By using LSI and creating a corpus of words for the datasets we have matched the job descriptions to the job titles given by the Allwyn corporation.



```
Name: Labor Category, dtype: object
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/chayavijaya/nltk_data...
[nltk_data]   Package punkt is already up-to-date!

In [18]: #build dictionary#
dictionary = Dictionary(gen_docs)
print(dictionary[5])
print(dictionary.token2id['project'])
print("Number of words in dictionary:",len(dictionary))

for i in range(len(dictionary)):
    print(i, dictionary[i])

senior
1
Number of words in dictionary: 20
0 manag
1 project
2 iii
3 comput
4 secur
5 senior
6 specialist
7 system
8 analyst
9 cloud
10 engin
11 data
12 scientist
13 develop
14 experi
15 user
16 ux
17 softwar
18 autom
19 test
```

Figure 8 Building dictionary

```
In [24]: query_doc_bow= [dictionary.doc2bow(content)for content in query_doc]
         print(query_doc_bow)

         query_doc_lsi= [lsi[i]for i in query_doc_bow]
         print(query_doc_lsi)

         b=index[query_doc_lsi]
```

```
[[(14, 1)], [(0, 1), (7, 1), (17, 1)], [], [], [(13, 3), (14, 1), (17, 2)], [(9, 1), (13, 1), (17, 1)], [(14, 1)],
[(17, 1)], [(14, 1), (17, 2)], [], [(5, 1), (10, 1), (17, 3), (18, 1), (19, 1)], [(14, 1)], [], [(0, 1), (7, 2)], [
(10, 2), (13, 2), (17, 2)], [(4, 1), (7, 1), (10, 1), (14, 1), (17, 1)], [(7, 1), (13, 1), (14, 1), (17, 2)], [(10,
2), (13, 2), (17, 2)], [(14, 1), (17, 1), (19, 4)], [(1, 1), (13, 1)], [(7, 1), (13, 1), (18, 1)], [(14, 2), (17, 2
), (19, 2)], [(13, 2)], [(13, 1), (17, 1), (19, 1)], [(13, 1), (14, 1), (17, 2)], [(10, 1), (15, 1), (17, 1)], [(13
, 3), (14, 1), (17, 2)], [(0, 1), (7, 1), (17, 1)], [(17, 1)], [(13, 1), (14, 2)], [(0, 1), (7, 1), (13, 1), (17, 1
)], [(14, 1), (17, 2)], [(0, 1), (7, 2)], [(10, 1), (13, 1), (14, 1), (17, 1)], [(13, 2), (17, 1)], [(13, 2), (14,
1), (17, 2)], [(13, 1), (14, 1), (17, 2)], [(14, 1), (15, 1), (17, 2)], [(0, 1), (10, 2), (13, 1), (17, 1)], [], [(
0, 1), (13, 1)], [(0, 1), (7, 1), (14, 2), (15, 1), (17, 1)], [(9, 1), (13, 1), (17, 1)], [], [(13, 1), (17, 1)], [
(13, 1)], [(1, 1), (8, 1), (13, 1), (17, 1)], [(11, 1), (13, 1)], [(4, 1), (13, 2), (17, 1)], [(13, 2), (14, 2), (1
7, 3), (19, 1)], [(1, 1), (13, 3), (14, 2), (17, 1)], [(0, 1), (13, 2), (17, 1)], [(13, 1), (17, 1)], [(10, 1), (17
, 1)], [(13, 2), (14, 1), (17, 2)], [(0, 1), (7, 2)], [(7, 1), (13, 1)], [(0, 1), (1, 1), (13, 1), (16, 1)], [(17,
1)], [(13, 1), (17, 1), (19, 1)], [], [(5, 1), (10, 1), (17, 3), (18, 1), (19, 1)], [(13, 2)], [], [(10, 1), (13, 2
)], [(13, 2), (17, 2), (19, 1)], [(11, 1), (13, 2), (17, 2)], [(17, 2)], [(0, 1), (7, 1), (13, 2), (17, 3)], [(0, 2
), (1, 1), (7, 1), (10, 1), (17, 1)], [], [(5, 1), (10, 1), (17, 3), (18, 1), (19, 1)], [], [(4, 1), (7, 1)], [(17,
2), (19, 1)], [(14, 1)], [(15, 1)], [(7, 1), (13, 1), (16, 1)], [(0, 1), (13, 2), (17, 1)], [(7, 2), (13, 1), (14,
1), (17, 1)], [(14, 1), (17, 2)], [], [(5, 1), (10, 1), (17, 3), (18, 1), (19, 1)], [(13, 1)], [(14, 1)], [(17, 1)]
, [(13, 1), (17, 1)], [(0, 1), (17, 2)], [(7, 1), (13, 2)], [(13, 2), (17, 1), (19, 1)], [], [(4, 1), (10, 1), (13,
1), (14, 1), (17, 1)], [(13, 2), (17, 2)], [(13, 1), (14, 1), (16, 1)], [(9, 1), (13, 2)], [], [(5, 1), (10, 1), (1
```

```
In [36]: A0=b[:,0]

         B0= A0.argsort()[-50:][::-1]
         df0=coldata.loc[B0.:]
```

Figure 9 Latent Semantic Analysis Output

N-gram Tokenizer:

The ngram tokenizer first breaks text down into words whenever it encounters one of a list of specified characters, then it emits N-grams of each word of the specified length. N-grams are like a sliding window that moves across the word - a continuous sequence of characters of the specified length. They are useful for querying languages that don't use spaces or that have long compound words, like German.

Here for the total_data we have done lemmatization where is converts words to its base form or dictionary form of word. Then as it was in list, we converted that to string. Then we performed ngram like bigram and trigram. Then we got frequency distribution for trigram.
For client_data after storing it in a data frame, we converted the description column to string. [stop word removal, etc.] we have done lemmatization where is converts words to its base form or dictionary form of word. Then as it was in list, we converted that to string. Then we performed ngram like bigram and trigram. Then we got frequency distribution for trigram.

	sentences	word_tokens	words_no_punctuation	words_no_sw	lemmas	full	tri-grams	bi-grams
0	[manages projects and development teams execut...	[Manages, projects, and, development, teams, e...	[manages, projects, and, development, teams, e...	[manages, projects, development, teams, execut...	[manages, project, development, team, execute,...	manages project development team execute range...	[manages project development, project developm...	[manages project, project development, develop...
1	[manages projects and development teams execut...	[Manages, projects, and, development, teams, e...	[manages, projects, and, development, teams, e...	[manages, projects, development, teams, execut...	[manages, project, development, team, execute,...	manages project development team execute range...	[manages project development, project developm...	[manages project, project development, develop...
2	[analyzes and defines security requirements fo...	[Analyzes, and, defines, security, requirement...	[analyzes, and, defines, security, requirement...	[analyzes, defines, security, requirements, mu...	[analyzes, defines, security, requirement, mul...	analyzes defines security requirement multilev...	[analyzes defines security, defines security r...	[analyzes defines, defines security, security ...
3	[analyzes security measures for more than one ...	[Analyzes, security, measures, for, more, than...	[analyzes, security, measures, for, more, than...	[analyzes, security, measures, one, functional...	[analyzes, security, measure, one, functional,...	analyzes security measure one functional area ...	[analyzes security measure, security measure o...	[analyzes security, security measure, measure ...
4	[experience with cloud services - including op...	[Experience, with, cloud, services, -, includi...	[experience, with, cloud, services, including,...	[experience, cloud, services, including, open,...	[experience, cloud, service, include, open, so...	experience cloud service include open source t...	[experience cloud service, cloud service inclu...	[experience cloud, cloud service, service incl...
5	[data scientist will have necessary statistica...	[Data, Scientist, will, have, necessary, stati...	[data, scientist, will, have, necessary, stati...	[data, scientist, necessary, statistical, mode...	[data, scientist, necessary, statistical, mode...	data scientist necessary statistical model mat...	[data scientist necessary, scientist necessary...	[data scientist, scientist necessary, necessar...
6	[responsible for creating front-end design sol...	[Responsible, for, creating, front-end, design...	[responsible, for, creating, front, end, desig...	[responsible, creating, front, end, design, so...	[responsible, create, front, end, design, solu...	responsible create front end design solution w...	[responsible create front, create front end, f...	[responsible create, create front, front end, ...
	[develop, modify,	[Develop, ,,		[develop, modify,	[develop, modify,	develop modify	[develop modify	[develop modify

```
    for word in FreqDist.keys():
        FreqDist[word] = (FreqDist[word]/max_freq)
    return FreqDist


ngram_freqs = find_weighted_frequency(fdist)
list (ngram_freqs.items())[0:20]
```

```
[('manages project development', 1.0),
 ('project development team', 1.0),
 ('development team execute', 1.0),
 ('team execute range', 1.0),
 ('execute range methodology', 1.0),
 ('range methodology include', 1.0),
 ('methodology include waterfall', 1.0),
 ('include waterfall agile', 1.0),
 ('waterfall agile lean', 1.0),
 ('agile lean ensures', 1.0),
 ('lean ensures project', 1.0),
 ('ensures project meet', 1.0),
 ('project meet scope', 1.0),
 ('meet scope schedule', 1.0),
 ('scope schedule budget', 1.0),
 ('schedule budget serve', 1.0),
 ('budget serve scrum', 1.0),
 ('serve scrum master', 1.0),
 ('scrum master agile', 1.0),
 ('master agile project', 1.0)]
```

Figure 10 N-gram Output

Preprocessing for Machine Learning Modelling:

Prior to modelling and building models to match the job descriptions, we are doing some preprocessing to get efficient results. Then we will use lemmatization process to map the words to the root. We are using genism library to do the summarization of the description. We are also using NLTK library for building models.
We are taking the description from the dataset and removing the stop words, punctuation, doing text preprocessing to get all the important key words. In this process we have created vectors for both the datasets and the technique we used to vectorize is TF-IDF.

3.3 Machine Learning Models

Random Regression

Random Regression models are used for the analysis of longitudinal dada or for repeated individuals over time. RRM allows the researcher to study changes in genetic variability with time and allow selection of individuals to alter the general patterns of response over time.[14]
We have used random regression on the dataset for prediction. We have predicted that employees with high experience have highest salary

```
X_grid = np.arange(min(X), max(X), 0.5)

# reshape for reshaping the data into a len(X_grid)*1 array,
# i.e. to make a column out of the X_grid value
X_grid = X_grid.reshape((len(X_grid), 1))

# Scatter plot for original data
#plt.scatter(X, y, color = 'blue')

# plot predicted data
plt.plot(X_grid, regressor.predict(X_grid),
         color = 'green')
plt.title('Random Forest Regression')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.show()
```

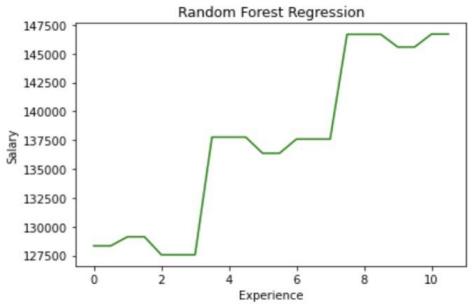


Figure 11 Random Forest regression

We have used random regression for indeed dataset as we were getting high rmse error because of the random values and synthetic data. We have got R-square value as 0.719 and rmse values 411591.

Result:

```
Ranger result

Call:
 ranger(salary ~ ., data = train, mtry = 2, num.trees = 500, write.forest = TRUE,      importance = "permu
tation")

Type:                             Regression
Number of trees:                  500
Sample size:                      3501
Number of independent variables:  6
Mtry:                             2
Target node size:                 5
Variable importance mode:         permutation
Splitrule:                        variance
OOB prediction error (MSE):       411591694
R squared (OOB):                  0.719328
```

Figure 12 Results of Random Forest regression

Linear Regression

Linear regression analysis predicts the value of one variable depending on the value of another. The variable you wish to forecast is referred to as the dependent variable. The variable you are using to forecast the value of the other variable is known as the independent variable. This type of analysis calculates the coefficients of a linear equation that includes one or more independent variables that best predict the value of the dependent variable. We will be using the AUC,F1 Score, Accuracy and other data outputs to consider the accuracy of this model.[15]

First, we split the data into training and test dataset. Then we have specified feature engineering recipe. We specified that salary is our response variable, and all others are predictor variables. We have prepped the recipe on the training data and applied on the test data. Then we have specified the model as linear regression and created a workflow object by combining recipe and model, then we processed the workflow with last_fit. Here we obtained the performance metrics and predictions on the test set, we used collect_metrics() and collect_predictions() functions on our workflow object. Then we predicted the salaries on the test dataset.

Results:

```
> jobs_fit %>%
+   collect_metrics()
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard   26523.
2 rsq     standard       0.524
```

Figure 13 Linear Regression results

Support Machine Vector:

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \qquad c(x, y, f(x)) = (1 - y * f(x))_+$$

Hinge loss function (function on left can be represented as a function on the right)

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Loss function for SVM

We have used SVM vector machine for the dataset and we have predicted the R-square value as 0.80 and rmse 16751. The R-square value attained is highest among all the three models.

Predicted Values:

```
y_pred = regressor.predict(X_test)
y_pred = sc_y.inverse_transform(y_pred)
```

```
df = pd.DataFrame({'Real Values':sc_y.inverse_transform(y_test.reshape(-1)), 'Predicted Values':y_pred})
df
```

	Real Values	Predicted Values
0	110998.0	111536.519061
1	137500.0	133639.288295
2	134000.0	130306.778985
3	68500.0	72097.836881
4	47000.0	87093.163198
...	...	...
995	65500.0	69085.793995
996	70000.0	73871.539452
997	62500.0	74951.420483
998	40500.0	81431.431739
999	118000.0	114172.592176

1000 rows × 2 columns

Figure 14 SVM Predicted values

RMSE and R-Square values:

```
from sklearn.metrics import mean_squared_error
from numpy import sqrt

mse = mean_squared_error(sc_y.inverse_transform(y_test.reshape(-1)), y_pred)
rmse = sqrt(mse)
rmse
```

```
16334.439666477487
```

```
from sklearn.metrics import r2_score
r2_score(sc_y.inverse_transform(y_test.reshape(-1)), y_pred)
```

```
0.8143384796770219
```

Figure 15 SVM RMSE and R-square values

4 Visualizations:

Model Visualizations:

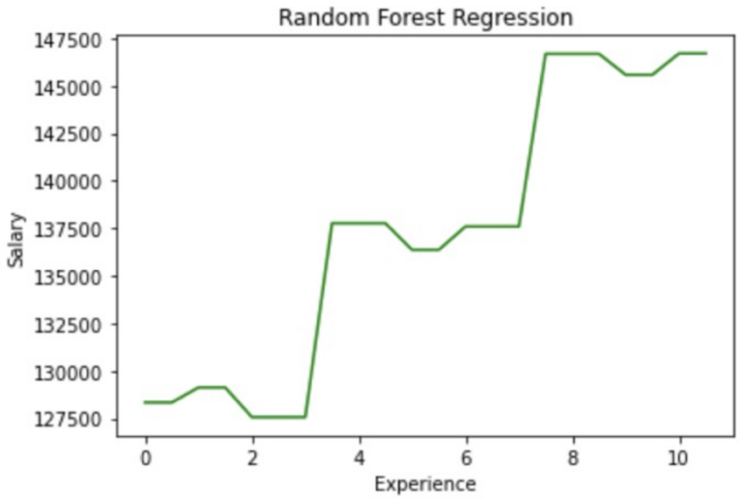


Figure 16 Visualization 1

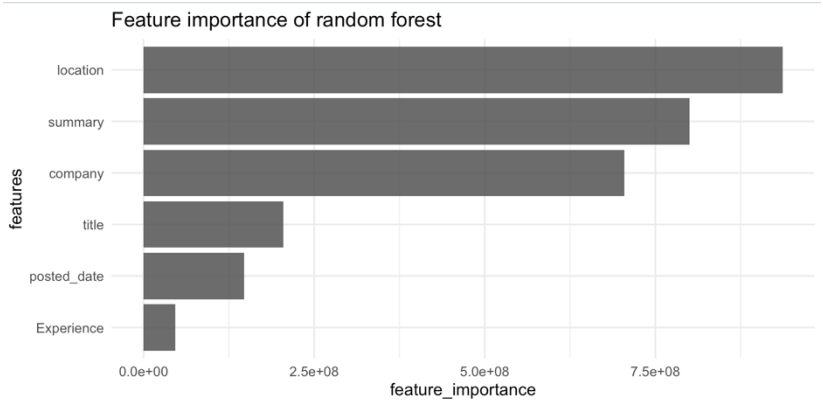


Figure 17 Visualization 2

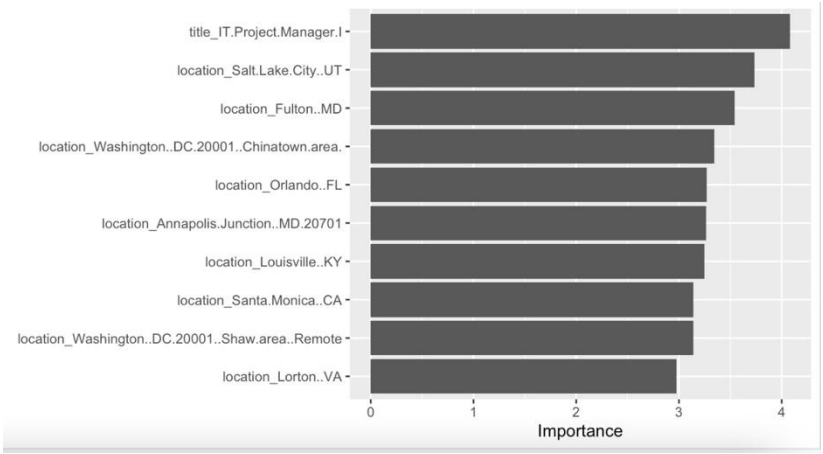


Figure 18 Variable importance plot from linear regression

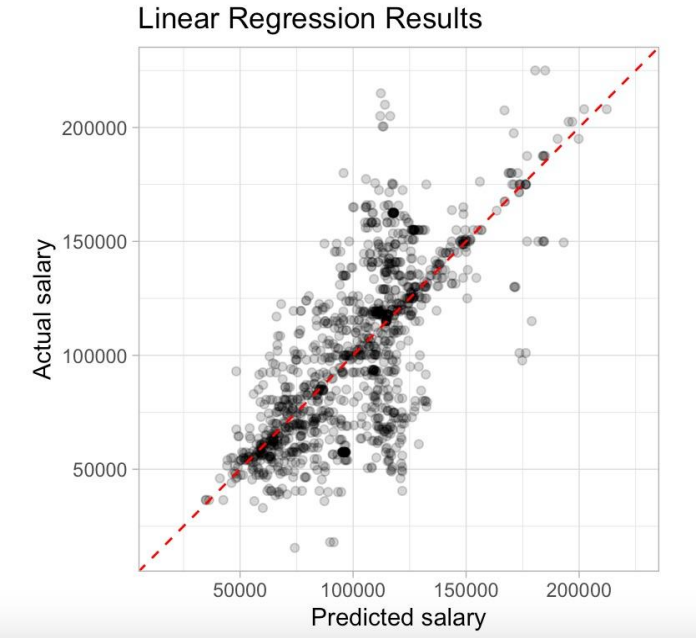


Figure 19 Linear regression results

Tableau Visualizations

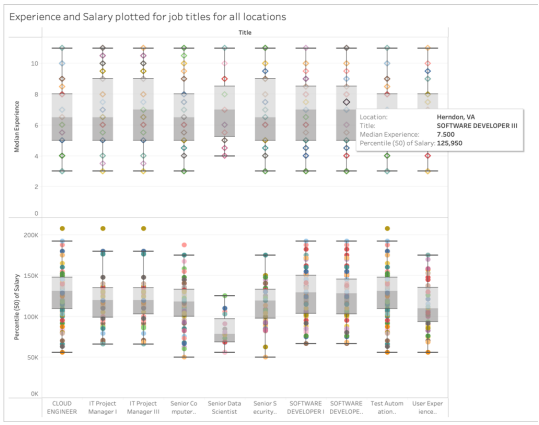


Figure 20 Tableau Visualization1

The above box plot has been created for the salaries and experience for all job titles and locations. From the plot we can observe that for the 50 percentile of salary the medians are much more skewed that that of medium experience employees.

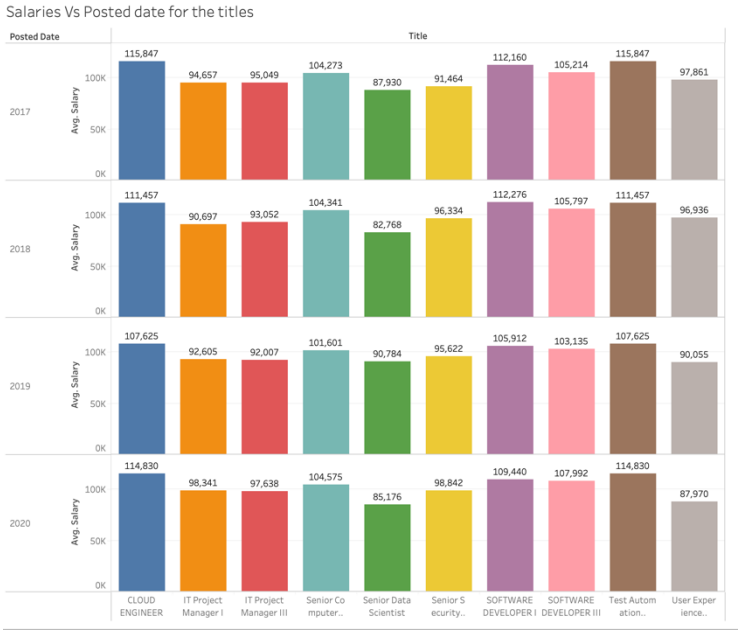


Figure 21 Tableau Visualization 2

The above visualization shows us the salaries for each year and each job title. From this we can understand that the highest salary is of the Cloud Engineer and Testing Automation. The maximum variation in salaries is observed to be of Senior Data Scientist and User experience tester.

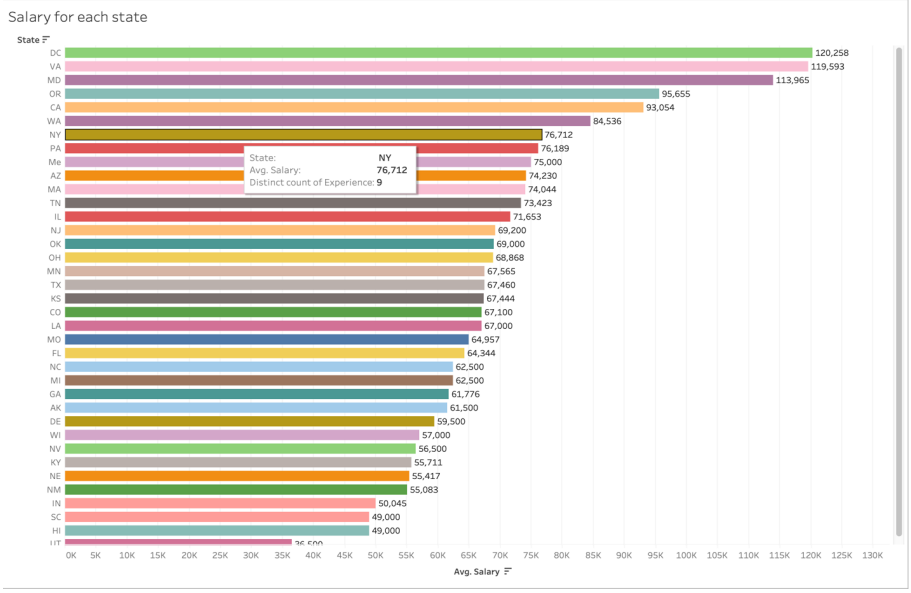


Figure 22 Tableau Visualization 3

The above visualization is of the average salary for each state in ascending order. From the plot we can observe that the highest salary is from the DC state and the from UT. Though DC has the highest salary it is in par with Virginia.

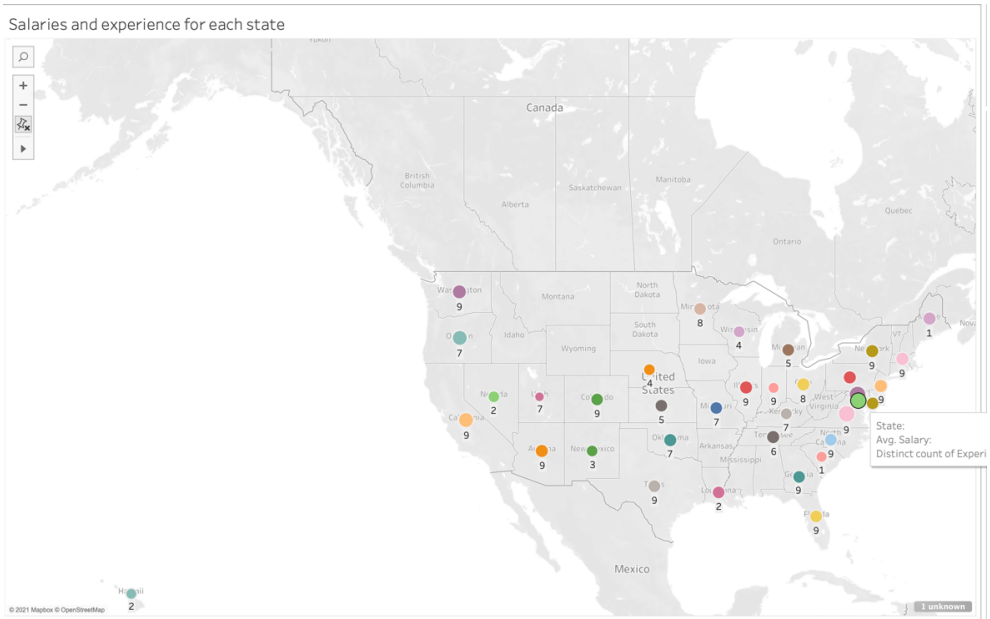


Figure 23 Tableau Visualization 4

This visualization is of the experience and average salaries for all the states. We can observe that the DC and VA have the highest average salaries.

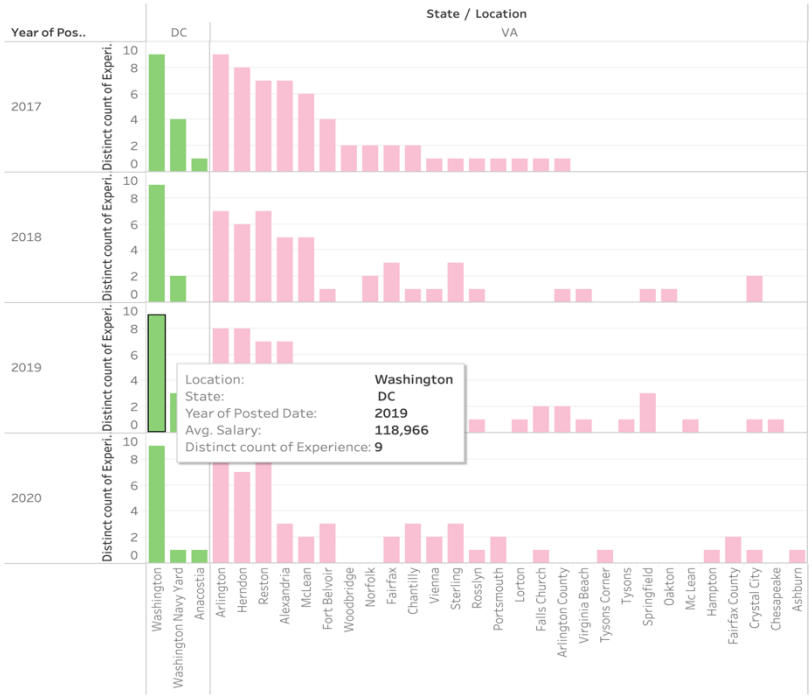


Figure 24 Tableau Visualization 5

The above visualization is of the experience and average salaries for the years 2017-2019 in the DC and VA areas.

Below are the visualizations for job posting for each role in different companies.

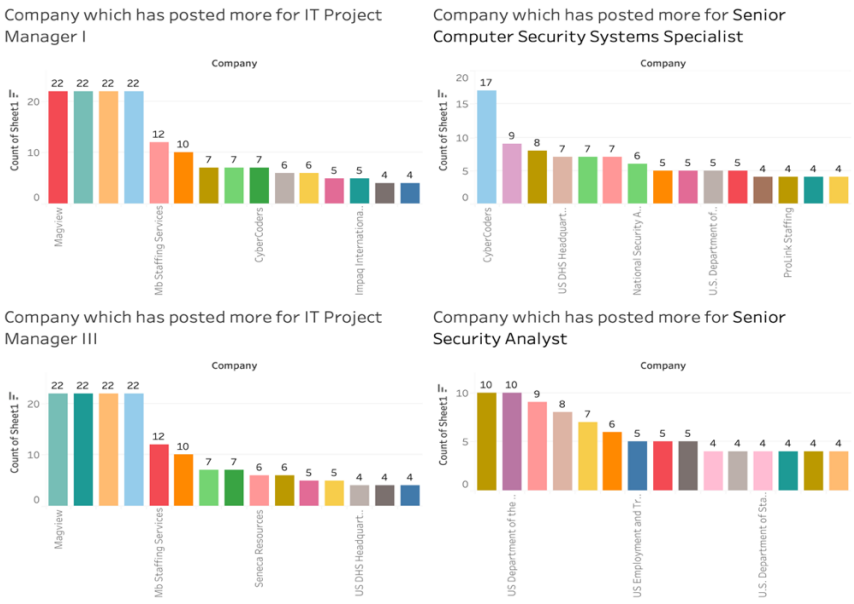


Figure 25 Tableau Visualization 6

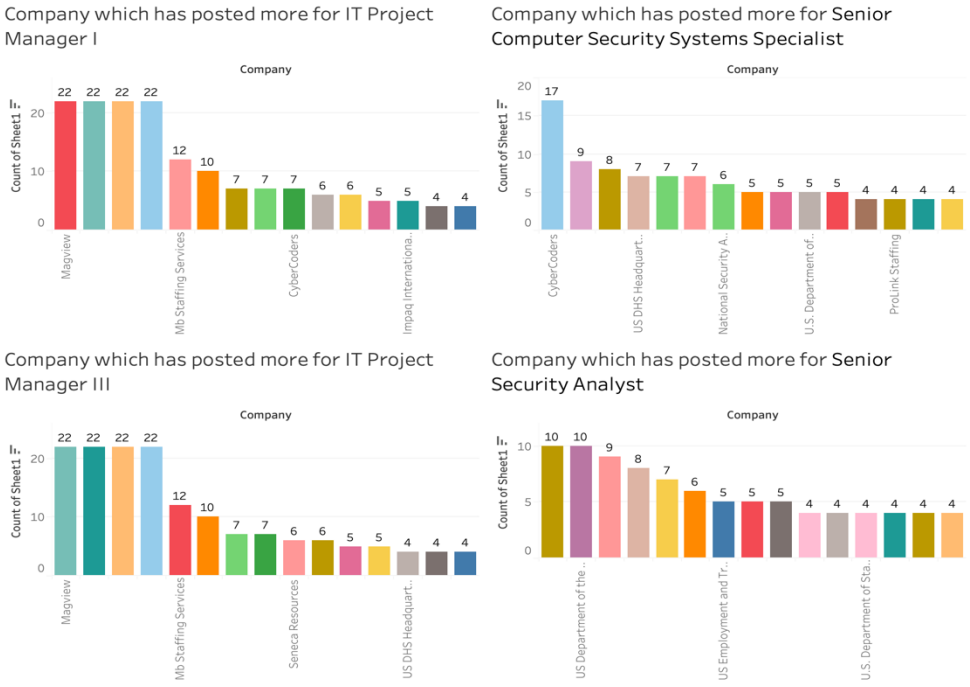


Figure 26 Tableau Visualization 7

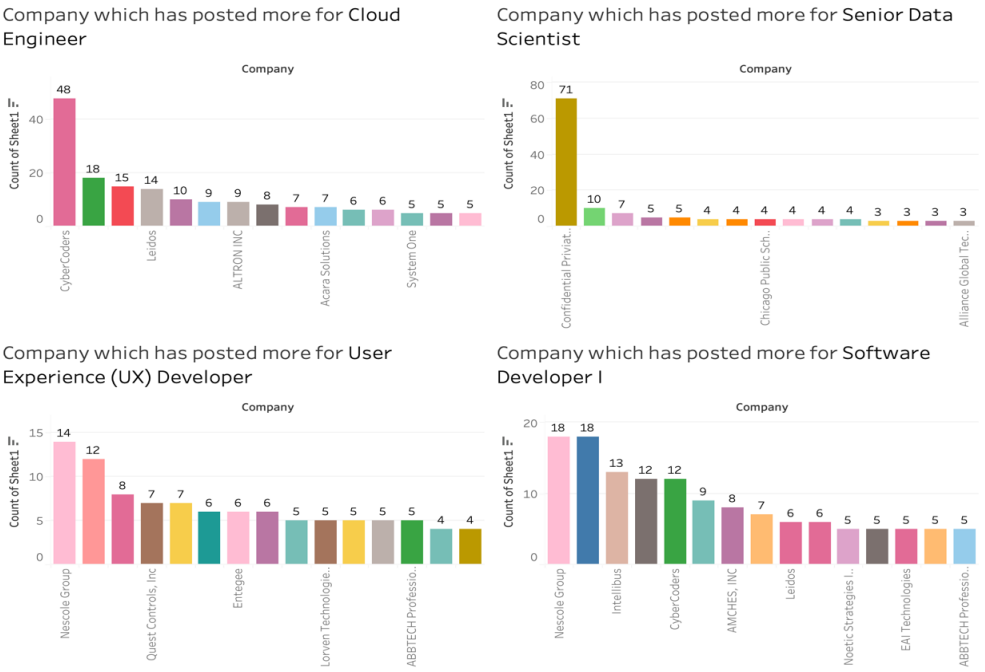


Figure 27 Tableau Visualization 8

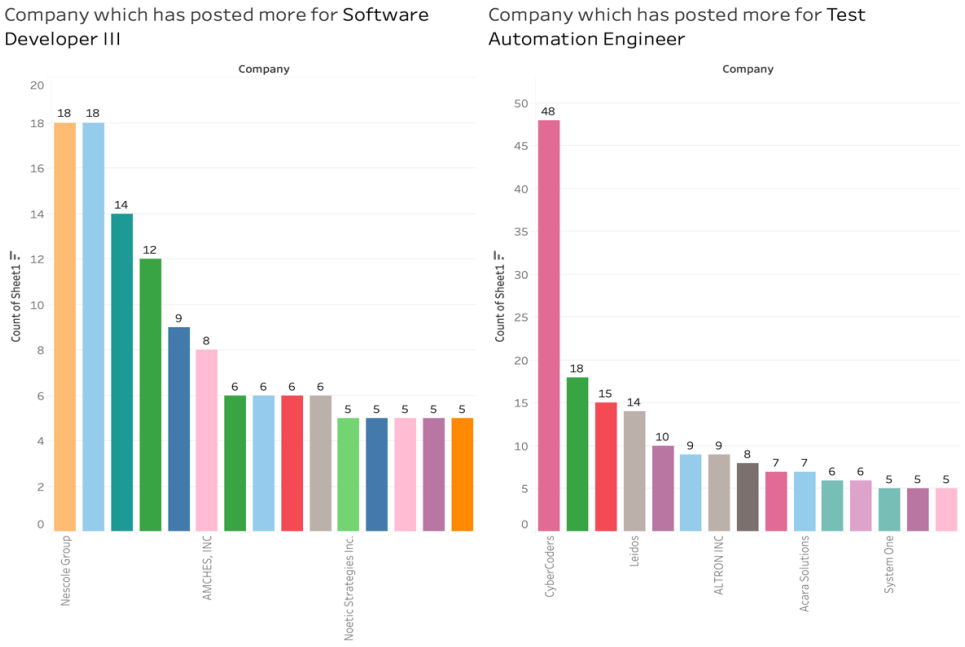


Figure 28 Tableau Visualization 9

Below are the visualizations for the average salary for different jobs

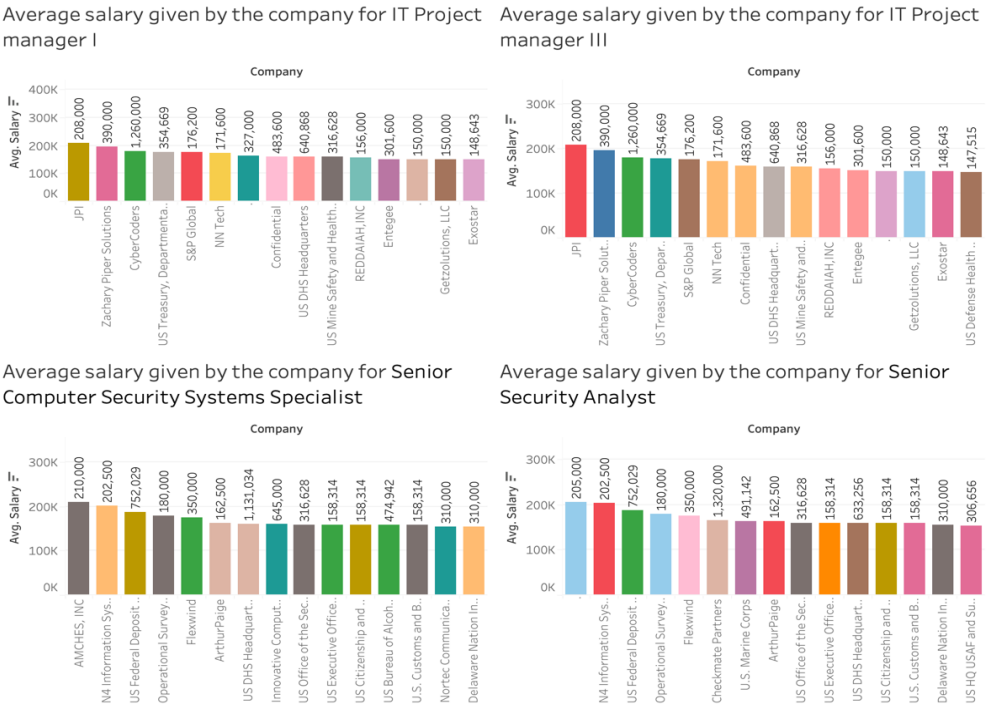


Figure 29 Tableau Visualization 10

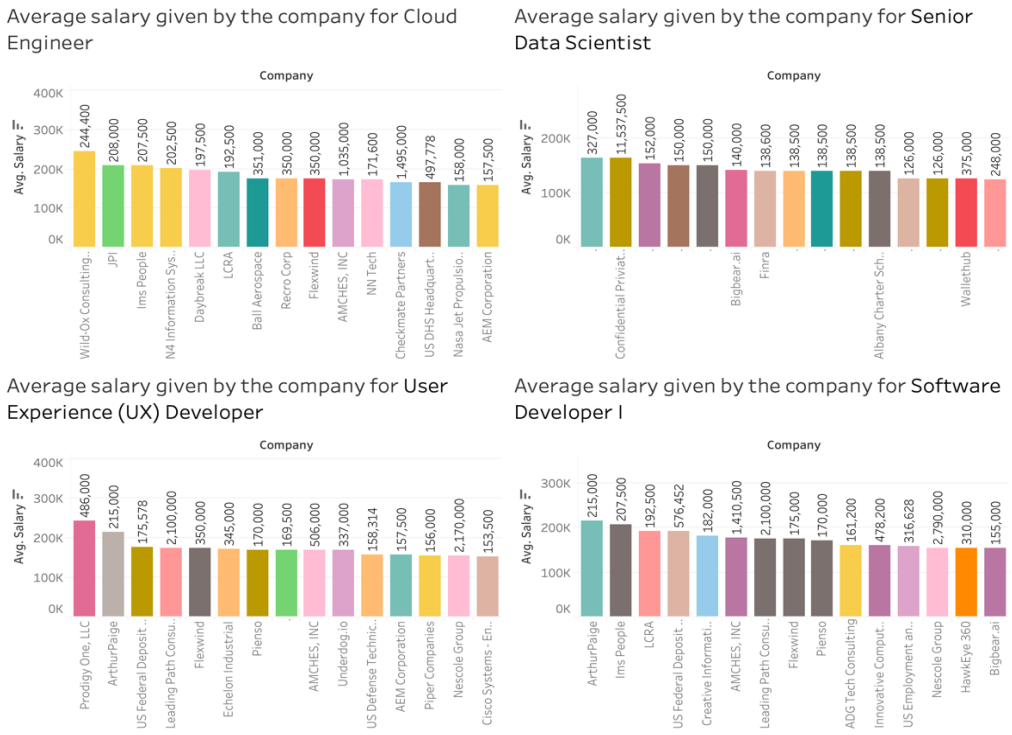


Figure 30 Tableau Visualization 11

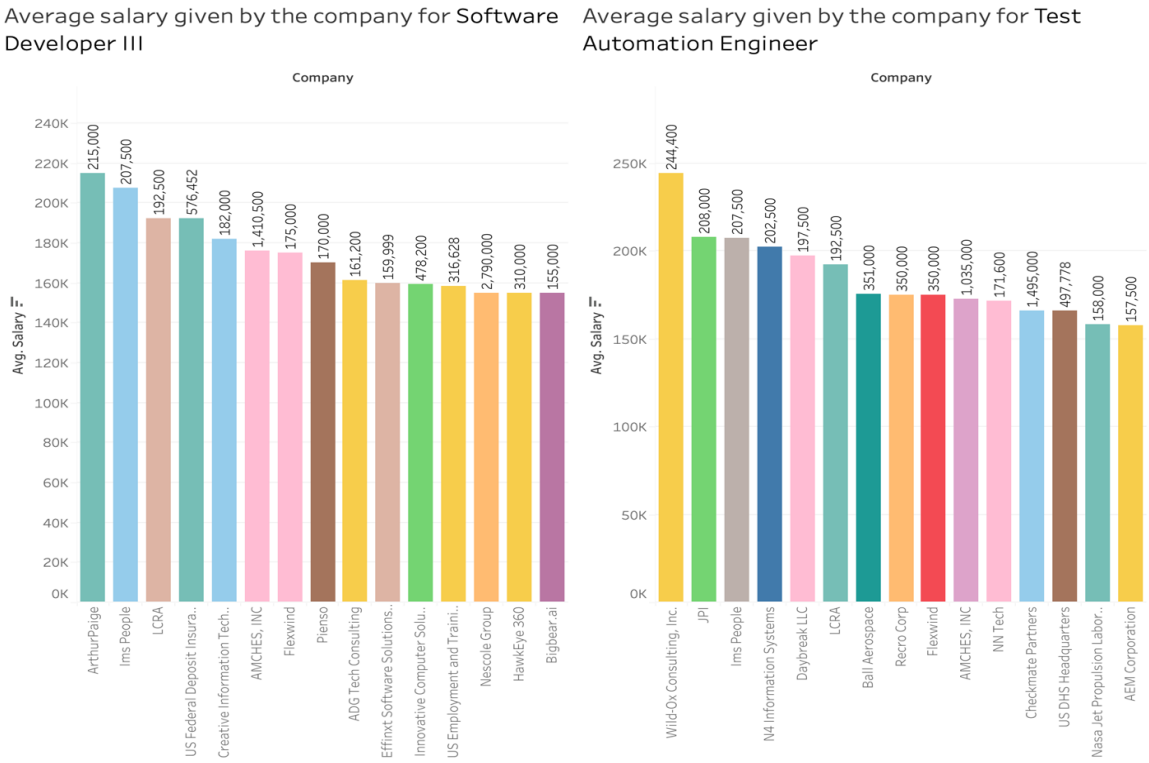


Figure 31 Tableau Visualization 12

Forecast Visualization

Exponential Smoothening

Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight. This framework generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in industry.[17]

Figure 32 Forecasting results for IT Project Manager I

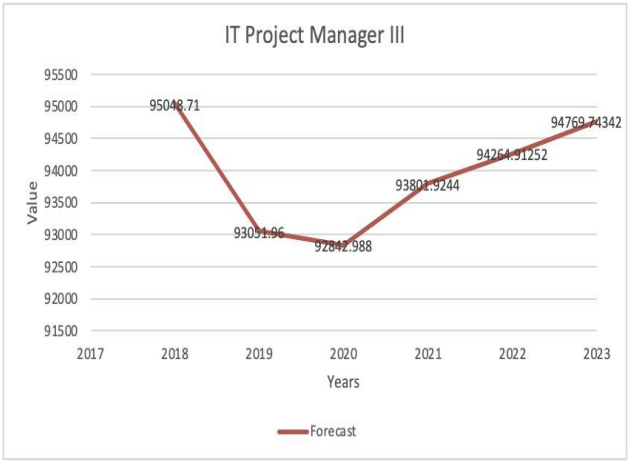


Figure 33 Forecasting results for IT Project Manager II

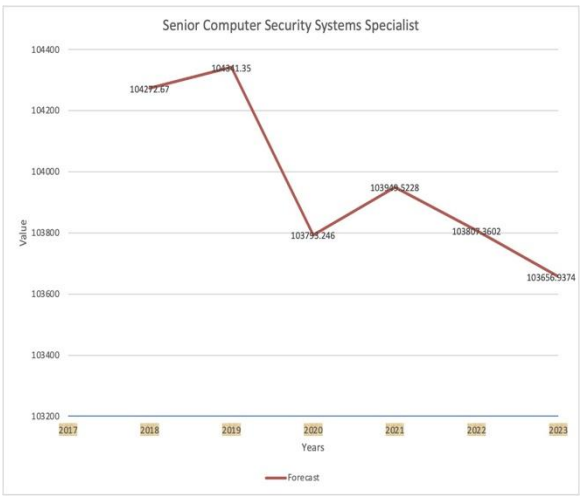


Figure 34 Forecasting results for senior computer systems specialist

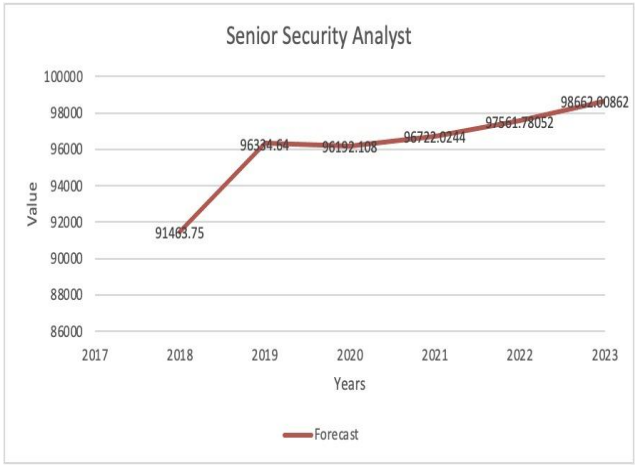


Figure 35 Forecasting results for Senior Security Analyst

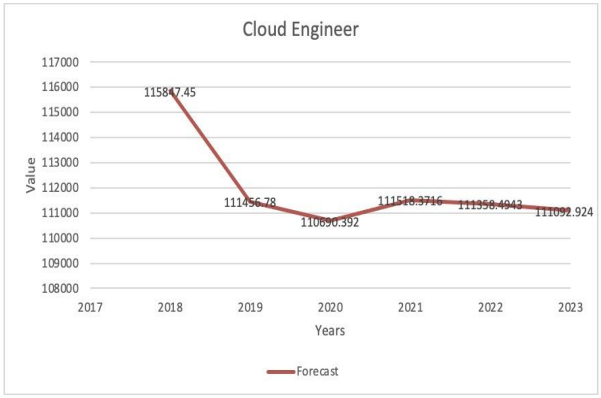


Figure 36 Forecasting results for Cloud Engineer

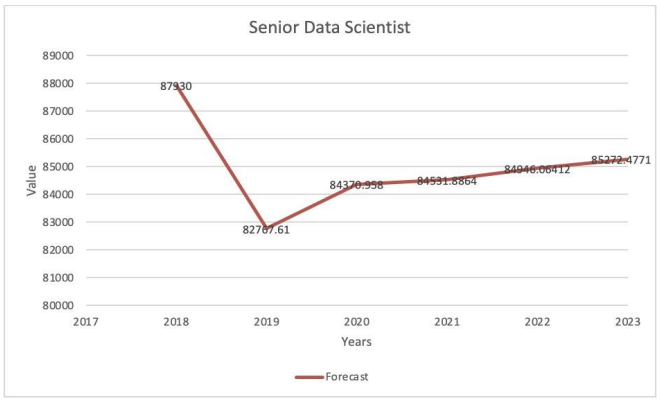


Figure 37 Forecasting results for Senior Data Scientist

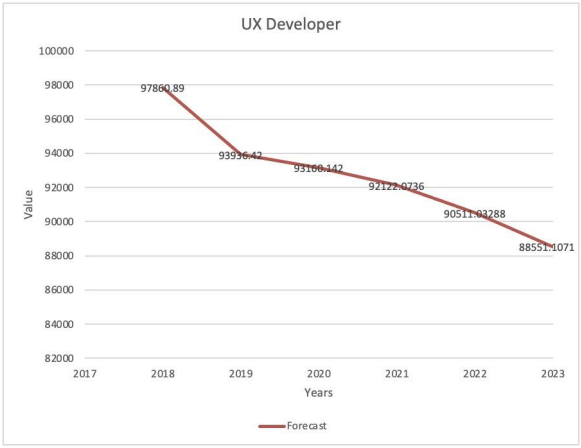


Figure 38 Forecasting results for UX Developer

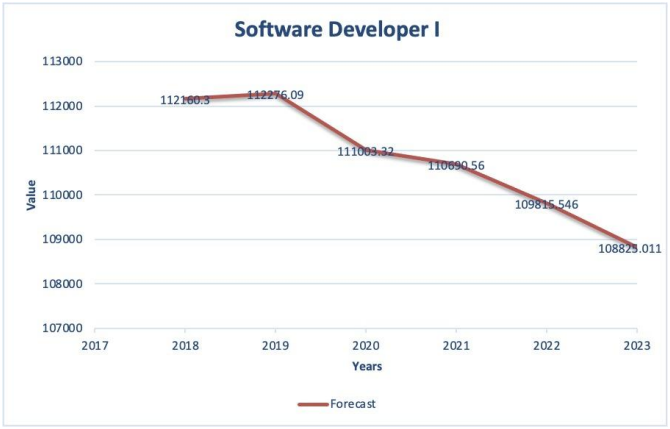


Figure 39 Software Developer 1

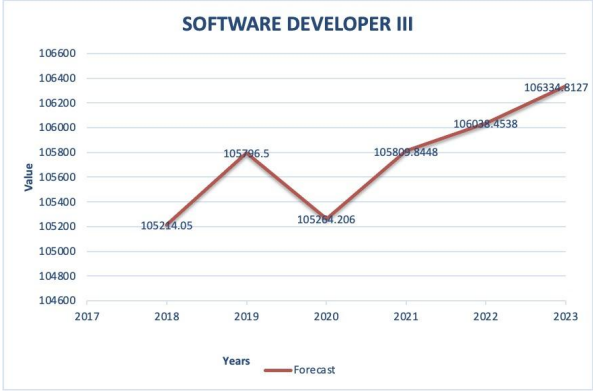


Figure 40 Forecasting results for Software developer III

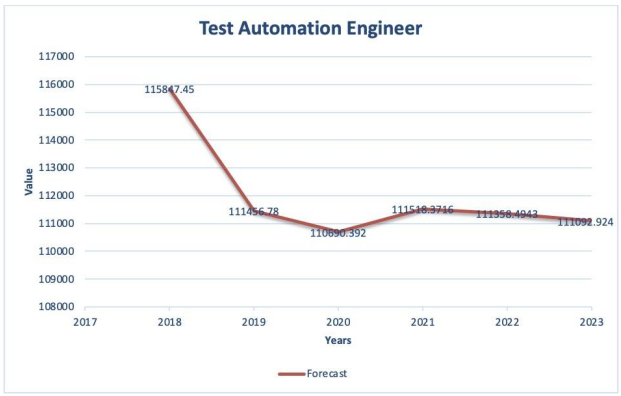


Figure 41 Test Automation Engineer

5 Finding:

These are the forecasted values for all the job titles, the units for exponential smoothing are notated in dollars/year.

IT Project Manager I	
Forecast	**Exponential smoothing**
2021	93487.93116
2022	94512.68073
2023	95591.72118

Senior Data Scientist	
Forecast	**Exponential smoothing**
2021	84946.06412
2022	85272.4771
2023	85528.67828

IT Project Manager III	
Forecast	**Exponential smoothing**
2021	94264.91252
2022	94769.74342
2023	95308.04853

UX Developer	
Forecast	**Exponential smoothing**
2021	90511.03288
2022	88551.1071
2023	86312.07328

Senior Computer Security Systems Specialist	
Forecast	**Exponential smoothing**
2021	103807.3602
2022	103656.9374
2023	103499.9063

Software Developer I	
Forecast	**Exponential smoothing**
2021	109815.546
2022	108825.011
2023	107742.0592

Senior Security Analyst	
Forecast	**Exponential smoothing**
2021	97561.78052
2022	98662.00862
2023	99970.61429

Software Developer III	
Forecast	**Exponential smoothing**
2021	106038.4538
2022	106334.8127
2023	106685.3713

Cloud Engineer	
Forecast	**Exponential smoothing**
2021	111358.4943
2022	111092.924
2023	110742.7994

Test Automation Engineer	
Forecast	**Exponential smoothing**
2021	111358.4943
2022	111092.924
2023	110742.7994

We have built Random regression, Linear regression, and Support Vector Machine models for the matched data. The features used for salary prediction are experience, posted date. Below are the model results.

MODEL	RMSE/annum	R-Square
Random forest	37694.46	13.47%
Random forest (with k-fold validation=5)	37492.33	13.80%
Linear Regression	26523	52.4%
Support Vector Machine	16335	81.0%

The above table shows the comparison between different model metrics like RMSE and R-square. Among all the models, support vector machine model performed better with RMSE values as $16,335/annum and R-square value as 81.0%.

6 Future Work

Limitations:

There were two major limitations for our project data and lack of historical data. We had a great challenge in acquiring data sources online as most of them are paid sources who were quoting too high which is difficult to retrieve. The second challenge was historical data. Most of the open-source data were lacking historical data. We could only get 1yr of data from open sources.

Recommendations:

As mentioned above, we had few limitations when working on the project and with the acquired data resources and time we could successfully implement the use case and got better performance results. With government contracts, other than the work, there are different costs that regularly don't get calculated in, for example, the expense of assets needed notwithstanding the work cost. With that in mind, one of the ideas we have for future work is to factor in other anticipated costs for a more thorough expense expectation. To factor in different expenses, we suggest using the public authority contract ID giving inside the BLS information to allude back to the agreements. Utilizing the agreement, we suggest making a python or potentially java script that will remove the vital data and extra to the BLS information source that was used for this venture.

In addition to that, we recommend allwyn corporation to implement extracting of indeed data every month and store it in a database to keep record of historical data. This could be done daily to a record of more current data and to get better performance results. Lastly, one last recommendation is to get data from paid sources as they have different labor categories by their wage rate and salary, education, experience, etc.

8 Appendix A: Code Reference

All the code files and data files have been uploaded to Git repository and it can be accessed through the following link:

https://github.com/Sahithi0664/DAEN-Project

9 Appendix B: Risk Selection:

We have the potential risks that could affect our could prevent our project outcomes and meet its requirements. We have also analyzed the level of risks and listed mitigation steps to be followed to avoid the risks.

Risk Table:

Risk Name	Description	Probability	Impact	Mitigation
Data Limitation	Availability of data; sometimes the data sources are changed or can restrict their data access.	Medium	High	Find multiple data sources and try to use the public data sources.
Data Preprocessing	Data also require a lot of preprocessing using NLP	Low	High	Use proper tools and technique.
Cost of resources	Cost of processing data can get high.	Low	High	Use free services like Azure for student, AWS educate, or GMU resources.
Data integration	data is from multiple sources, so need extra time to integrate them.	High	High	Identifying common attributes to integrate the data.
Job Description Matching	As job descriptions differ from our dataset to client provided description, we need to build models to match those job titles.	High	High	Need to Use NLP to build models.
Predicting the Salaries	Predict the salary using the correct model and getting good accuracy	Medium	Medium	Need to try different models for prediction.

Forecasting	Forecasting 3-4 years salaries	low	low	Performing exponential smoothing and forecasting the values
Decreasing Salaries	Salaries decreasing for future years	Medium	Medium	Try different methodologies to compare

10 Appendix C: Agile Methodology:

We have followed agile development framework throughout the project. We have assigned different roles to the team members depending on the various areas of the work.

Role	**Member Assigned**	**Duties Assigned**
Project Owner	Sahithi Reddy Godishala	Served as primary contact to the team for all the communications with the client.
Scrum Master	Vaishnavi Kammalampudi	Lead the team with daily scrums and responsible for creating and updating tasks in YouTrack sprint board and user stories.
GitHub Manager	Chaya Vijaya Lakshmi Adari	Responsible in maintaining teams GitHub repository.
Lead Developers	Sahithi Reddy Godishala Hamza Habib Chaya Vijaya Lakshmi Adari Rishi Thodupunuri Rajendra	Responsible for creating all the scripts that predictive models that are used for project.

Figure 42 Agile methodology

11 References

(n.d.). Retrieved from https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

About Linear Regression. (2021, Oct 20). Retrieved from https://www.ibm.com/topics/linear-regression

Brownlee, J. (2019, Aug 7). *What is natural language processing? Machine learninng Mastery*. Retrieved from https://machinelearningmastery.com/natural-language-processing/.

Burns, E. L. (2021, July 1). *what is artificial intelligence? - AI definition and how it works*. Retrieved from https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence

Chen, J. (2021, Sep 4). *Bid definition, Investopedia*. Retrieved from https://www.investopedia.com/terms/b/bid.asp.

Contract- Awarded Labor Category. (n.d.). Retrieved from calc.gsa.gov/.

Daffodil. (2020, Aug 12). *Top 10 Pre- Trained NLP Language Models*. Retrieved from https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-ngram-tokenizer.html"

Denton, J. (2021, Aug 6). *U.S. Labor Market on track for recovery with almost 1M jobs added*. Retrieved from https://www.thinkwhy.com/news-detail/july-2021-national-jobs-report-and-labor-forecast/

Encyclopaedia Britannica, inc. (n.d.). Retrieved from https://www.britannica.com/topic/contract-law.

Exponential smoothing: Forecasting principles and practice. (2021, Nov 03). Retrieved from Available: https://Otexts.com/fpp2/

Gandhi, R. (2018, June 7). *Support Vector Machine - Introduction to machine learning algorithms* . Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

IBM Cloud Education. (n.d.). Retrieved from https://www.ibm.com/cloud/learn/machine-learning

Kaput, M. (2021, May 12). *AI for Predictive Analytics: Everything You Need to Know*. Retrieved from https://www.marketingaiinstitute.com/blog/ai-for-predictive-analytics

Kochhar, R. (n.d.). Retrieved from https://www.pewresearch.org/fact-tank/2021/04/14/u-s-labor-market-inches-back-from-the-covid-19-shock-but-recovery-is-far-from-complete/

Navlani, A. (2018, Oct 9). Retrieved from https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

NAvlani, A. (2018, Sep 9). Retrieved from https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

OECDI Library. (n.d.). Retrieved from https://www.oecd-ilibrary.org/sites/5a700c4b-en/1/3/1/index.html?itemId=/content/publication/5a700c4b-en&_csp_=d31326a7706c58707d6aad05ad9dc5ab&itemIGO=oecd&itemContentType=book

Scott, w. (2019, Feb 19). *TF-IDF from scratch in python on a real-world dataset*. Retrieved from https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

U.S Bureau of Labor Statistics U.S. (2021, Mar 31). Retrieved from www.bls.gov/oes/oes_emp.htm#overview.

Wiggers, K. (2021, July 16). *AI Weekly: Can AI predict labor market trends?* Retrieved from https://venturebeat.com/2021/07/16/ai-weekly-can-ai-predict-labor-market-trends/