**Principles of Big Data Management**
**COMP-SCI 5540 [Fall Semester 2023]**
**Research Project**
**11/20/2023**
**Group 7**

# The Effects on Data Engineering Salaries on Employment Factors

| | |
|---|---|
| **Semir Hot** | **16297750** |
| **Nikhil Jagadeesh Sriram** | **16352573** |
| **Rumana Taj Shaik** | **16357409** |
| **Krishna Teja Nuni** | **16356387** |
| **Radha Krishna Siram** | **16356525** |
| **Sai Kiran Basetty** | **16355550** |
| **Surendra Babu Gorantla** | **16352171** |

## Abstract

In the dynamic landscape of data engineering one can see that in the past four years there has been substantial change in job market trends, these changes are influenced by multiple factors such as the surge in remote work, salary adjustments, and diverse employment locations. In our research we will be analyzing a data set from Kaggle that contains a plethora of factors such as employment year, experience level, salary, company location, company size and many others that will be explored deeper. This dataset has been updated every year and has just recently been updated for 2023. By employing advanced analytical techniques this paper will guide the reader to better understand how these factors all have an effect on each other and trends that normally one would not be able to notice just by looking at the data.

## Introduction

In the ever changing world of data engineering, understanding the intricate patterns between the employment factors and salaries has become important. The project at hand dives into the comprehensive analysis of the “Data Science Salaries 2023” dataset sourced from Kaggle, offering a unique way to unravel the relationships which shape the compensation structures within the domain of data engineering. As organizations increasingly rely on data-driven decision making, which hiked the demand for skilled data engineers. Consequently, comprehending the factors influencing the data engineering salaries is a vital part for both the employers and professionals in the job market. This project methodology involves using advanced analytical techniques, data transformation techniques, migration of data to Hadoop Distributed File System (HDFS), and in-depth data analysis. Through the python based code and visualizations tools like pandas and other python libraries, we aim to unravel insights beyond surface level observations, by offering a deep understanding of the interplay between variables such as experience, education, geographical location, and company size. This report provides a walkthrough of the project’s goals and objectives, detailing the steps performed in data engineering, data analysis, modeling, and the challenges and constraints encountered in the process.

Overall, this projects stands at junction of data science and employment trends, offering valuable insights that can help employers, professionals make strategic decisions which in turn contribute to the broader world of the evolving data engineering landscape, by providing a roadmap for researchers and practitioners seeking to navigate the complexities of data engineering salaries and their influencing factors.

## Project Goals & Objectives

**1. Comprehensive Analysis of Data Science Salaries:**

- **Goal:** To conduct an in-depth analysis of the "Data Science Salaries 2023" dataset.

- **Objective:** Explore the dataset to understand patterns, trends, and relationships between various factors influencing data engineering salaries.

**2. Unraveling Relationships Between Employment Factors and Salaries:**
- **Goal:** Identify and analyze the intricate patterns between employment factors and salaries.
- **Objective:** Use advanced analytical techniques to uncover relationships such as the impact of experience, education, geographical location, and company size on data engineering salaries.

**3. Employ Advanced Analytical Techniques:**
- **Goal:** Utilize advanced analytical methods to gain insights beyond surface-level observations.
- **Objective:** Apply techniques such as machine learning models, statistical analysis, and data transformation to extract meaningful insights from the dataset.

**4. Understand the Impact of Remote Work on Salaries:**
- **Goal:** Investigate how remote work trends influence data engineering salaries.
- **Objective:** Analyze the dataset to identify correlations between remote work, salary adjustments, and employment locations.

**5. Provide a Roadmap for Decision-Making:**
- **Goal:** Offer valuable insights for employers and professionals in the data engineering job market.
- **Objective:** Create a roadmap based on research findings to guide decision-making for strategic choices in salary structures and employment practices.

**6. Explore Trends Over Time:**
- **Goal:** Understand the trends in data engineering salaries over the past four years.
- **Objective:** Visualize and analyze the changes in average salaries, considering factors such as employment year, experience level, and company size.

**7. Showcase the Significance of the Dataset:**
- **Goal:** Highlight the relevance and reliability of the "Data Science Salaries 2023" dataset.
- **Objective:** Emphasize the gold standard status of the dataset on Kaggle and its regular updates, ensuring its credibility for research purposes.

**8. Provide Insights for Data Science and Employment Trends:**
- **Goal:** Stand at the intersection of data science and employment trends.
- **Objective:** Contribute valuable insights to the evolving landscape of data engineering salaries, serving as a resource for researchers and practitioners navigating the complexities of the field.

**9. Develop and Validate Machine Learning Models:**
- **Goal:** Create machine learning models to predict and understand salary variations.

- **Objective:** Implement and validate models such as Decision Trees, Logistic Regression, and Random Forest to predict salary ranges based on factors like experience, job title, and company size.

**10. Address Project Limitations and Constraints:**
- **Goal:** Acknowledge and manage project limitations and constraints.
- **Objective:** Clearly outline and communicate limitations related to data quality, resource constraints, dataset scope, incomplete information, geographical representation, limited temporal scope, and external factors.

**11. Demonstrate Feasibility of Research:**
- **Goal:** Establish the feasibility and relevance of the research.
- **Objective:** Conduct a feasibility study to show the relevance and significance of the chosen dataset, emphasizing its applicability to the research goals.

**12. Provide a User-Friendly Interface for Data Exploration:**
- **Goal:** Design an interface for easy exploration of data and analytical outcomes.
- **Objective:** Develop a user-friendly interface, integrating analytical outcomes, visualizations, and key findings to facilitate effective communication of the research results.

**13. Implement Test Cases for Validation:**
- **Goal:** Ensure the accuracy and reliability of analytical outcomes.
- **Objective:** Develop and execute test cases to validate the accuracy of data transformations, analytical models, and visualization tools used in the project.

## Project Scope

This project encircles the entire data analysis pipeline, from transforming the taken dataset “Data Science Salaries 2023” and loading it into a SQL database to visualizing salary related trends using pandas and other python based tools and Machine learning models. While the primary focus is on providing insights into trends, changes and potential relationship between the data engineering salaries and employment factors.

To understand the magnitude of change we have visualized the Average salary by work year and shown how much it has actually changed. When we take a look at the average salary in the data engineering field in 2020 we can see that it has grown almost 50% in
2023. This is also taking into account new positions and many other factors.

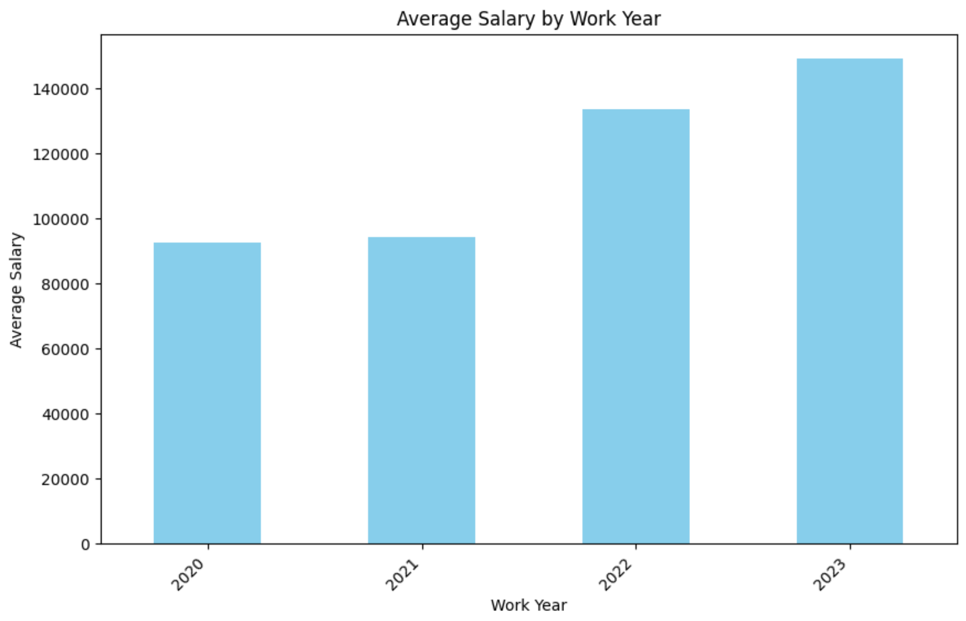
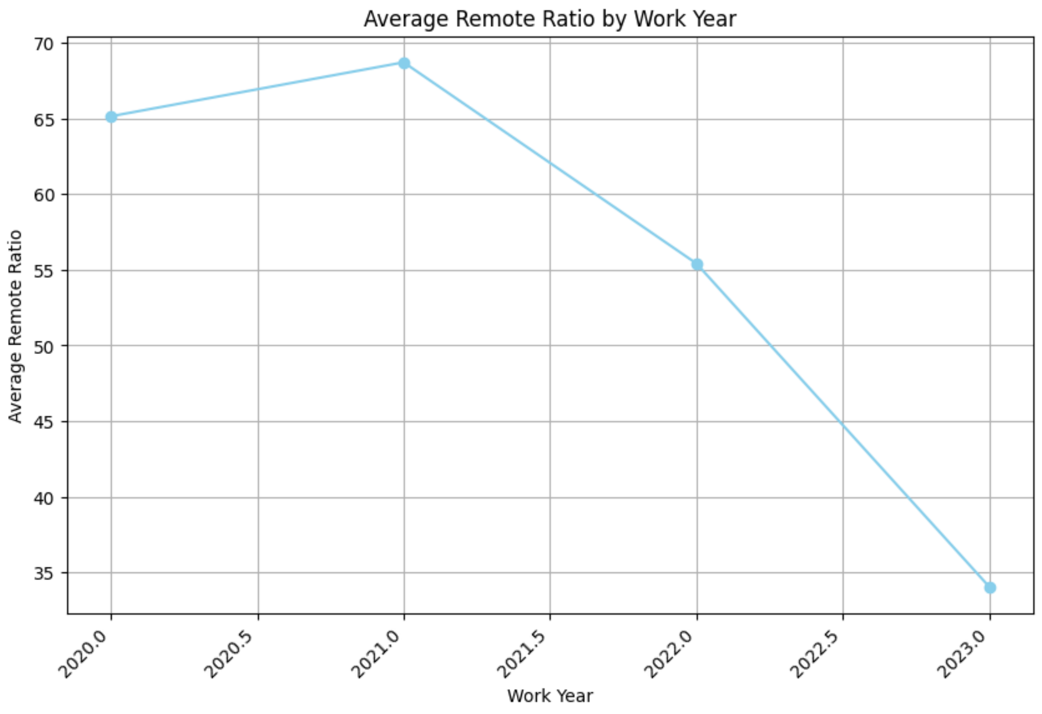


**Figure 1**

In 2020 and 2021 we had covid and we can see by this visualization that almost 65 to 68% of all jobs were remote during those years and this shows an effect on the salaries when we take a look at figure 1

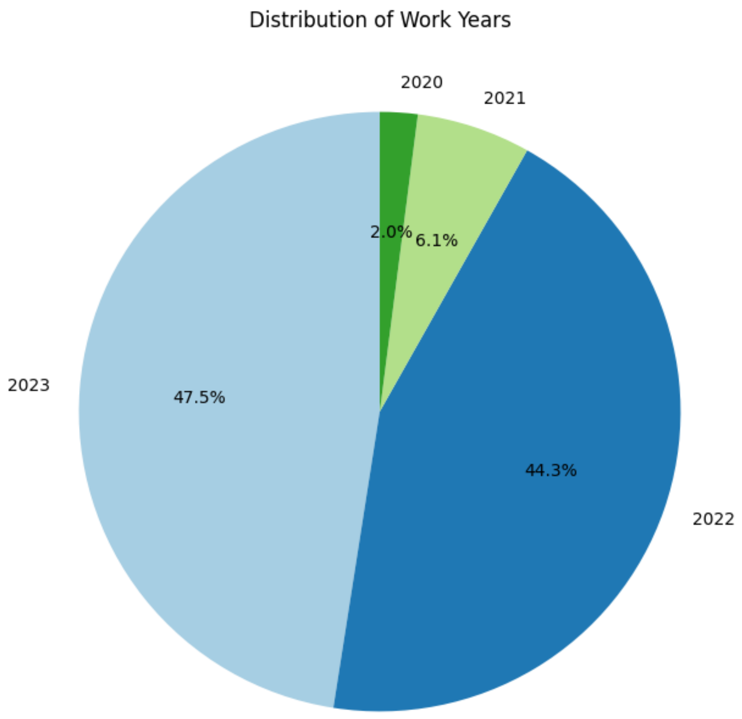


## Project Limitations & Constraints

Several limitations and constraints in the project are:

### Data Quality:

The accuracy of the analysis will depend on the quality of the "Data Science Salaries 2023" dataset. So any errors or inconsistencies may impact the reliability of the results. Taking a look at how relevant the data is, we can see that a majority of the data is from 2023, meaning that this data is very current and updated. With all of the years of salary data present we will be able to draw conclusions to changes and what things affect these salaries both from year to year and important factors that have more weight over others.



**Note:** *Kaggle was the source of our data, and we took special care in selecting a dataset of high reliability and relevance. The dataset, titled "Data Science Salaries 2023," has received a Kaggle gold medal, signifying its exceptional quality and accuracy within the Kaggle community. Notably, this dataset is consistently updated, maintaining its status as a gold standard on Kaggle.*

### Resource Constraints:

This project operates with the limitations of time, storage, and processing power available for detailed comprehensive analysis.

### Scope of DataSet:

The insights derived are specific or limited to the “Data Science Salaries 2023” dataset, and the applicability to broader contexts is limited.

### Incomplete Information:

The dataset might have records or variables for which there is insufficient information. Limitations in the analysis may arise from handling missing data and putting imputation procedures into practice.

### Geographical Representation:

Conclusions about data engineering salaries and employment factors that are applicable to a wide range of situations may be limited by the dataset's possible lack of global representation.

### Limited Temporal Scope:

This dataset captures information for the year 2023, providing a snapshot of that specific timeframe. It’s possible that the long-term developments or patterns that happened before or after this time aren’t properly represented.

### External Factors:

External Factors such as industry-specific trends or economic conditions (such as inflation, crisis, etc) may impact the interpretation of salary related findings.

### Feasibility Study:

The project demonstrates the relevance and significance of data engineering salary trends, utilizing a well-curated and regularly updated dataset from Kaggle.Using Apache Spark, the whole analysis includes data extraction, processing, and loading. Advanced analytics and machine learning models are then applied for predictive insights.The restrictions include limitations in terms of resources and scope as well as possible problems with data quality.Despite these challenges, the project's feasibility is supported by the dataset's gold standard reputation on Kaggle and the successful implementation of analytical techniques.Employers and professionals navigating the ever-changing world of data engineering salaries can benefit greatly from the findings, which make the attempt both realistic and significant.

## System Requirement Specifications (SRS)

### 3.a. Software Requirements:

The following software packages and libraries were used for the implementation of this project:

- PySpark: A powerful distributed computing environment for processing large datasets
- Pandas: A powerful data manipulation library in python.
- Matplotlib: A data visualization library in python.
- Seaborn: popular data visualization library built on top of Matplotlib

**3.b. Hardware Requirements:**

Google Colab, a cloud-based platform for data analysis and machine learning.

As such, there are no specific hardware requirements for running the project on a local machine. However, a stable internet connection is required to access and use the Google Colab platform.

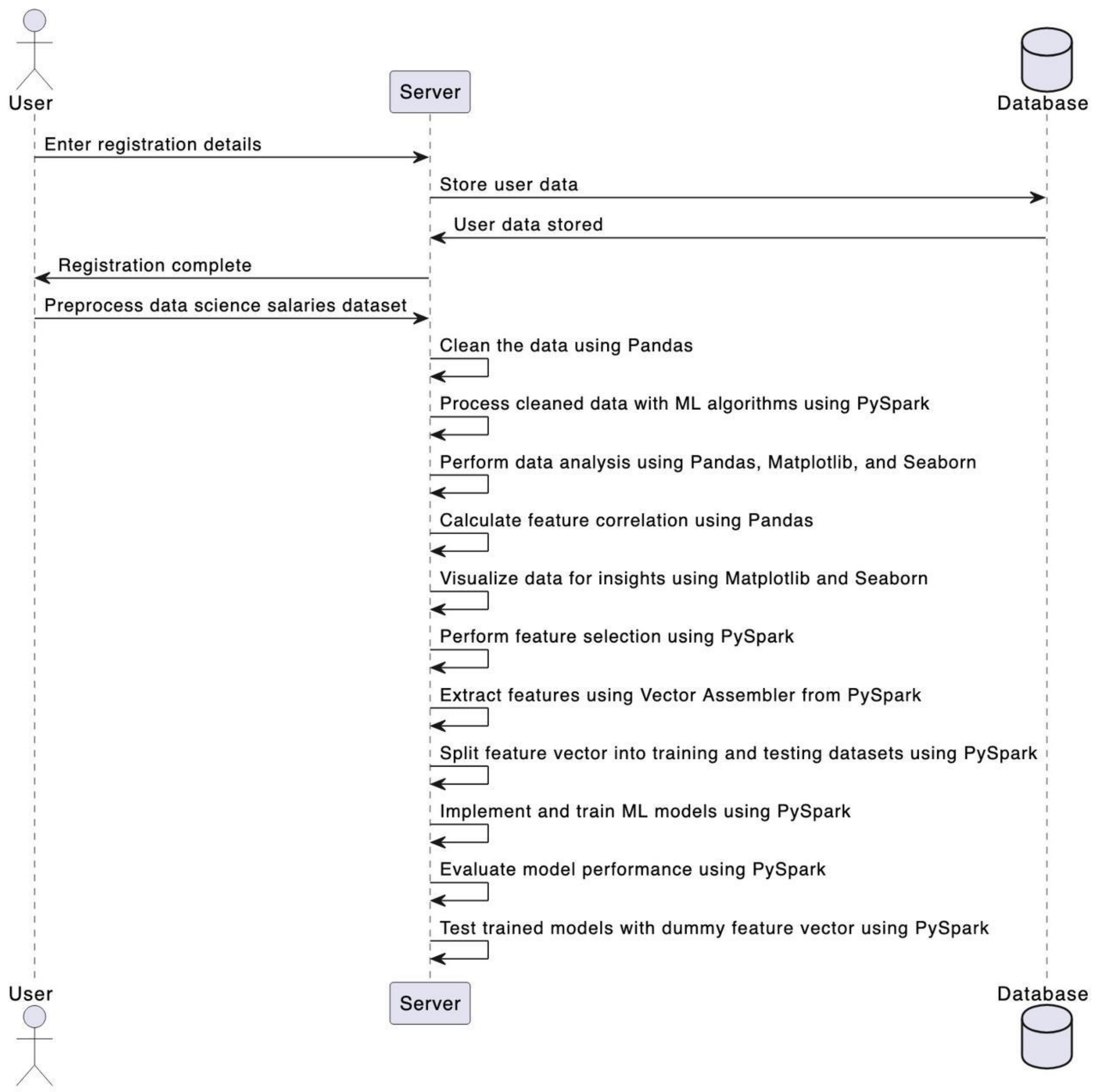
**3.c. Functional Requirements:**

The project's functional requirements include:

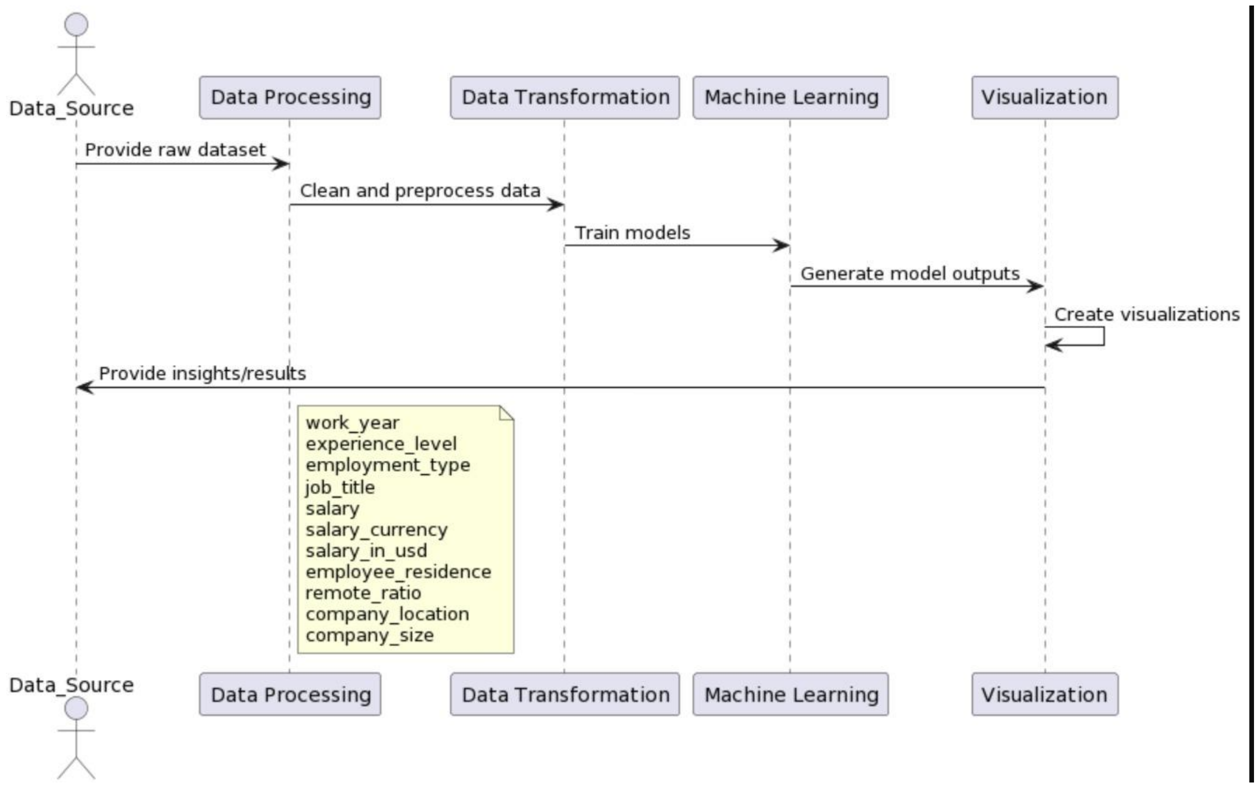
- Preprocessing the data science salaries dataset by cleaning the data.
- Processing the cleaned data according to machine learning algorithms.
- Performing data analysis to calculate correlation between all the features and to visualize the data for valuable insights, which helps in the feature selection.
- Extracting the features using a feature transformer, Vector Assembler.
- Splitting the feature vector into training and testing datasets.
- Implementing and training machine learning models on the training dataset.
- Evaluating the performance of the model using different evaluation metrics such as accuracy.
- Testing the trained models using a dummy feature vector to evaluate the quality of the trained models and their performance on any unseen dataset.

## System Design

### Architectural Diagram And Use-Case Diagram


User
Server
Database
Enter registration details
Store user data
User data stored
Registration complete
Preprocess data science salaries dataset
Clean the data using Pandas
Process cleaned data with ML algorithms using PySpark
Perform data analysis using Pandas, Matplotlib, and Seaborn
Calculate feature correlation using Pandas
Visualize data for insights using Matplotlib and Seaborn
Perform feature selection using PySpark
Extract features using Vector Assembler from PySpark
Split feature vector into training and testing datasets using PySpark
Implement and train ML models using PySpark
Evaluate model performance using PySpark
Test trained models with dummy feature vector using PySpark
User
Server
Database

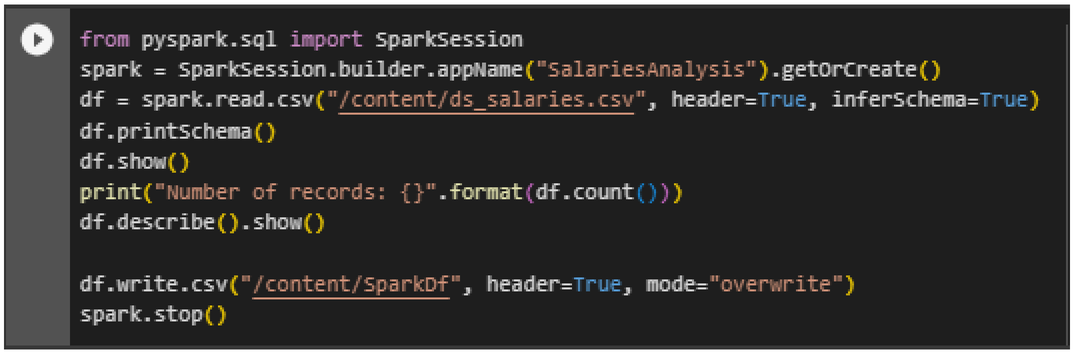
## Sequence Diagram



## Data Design

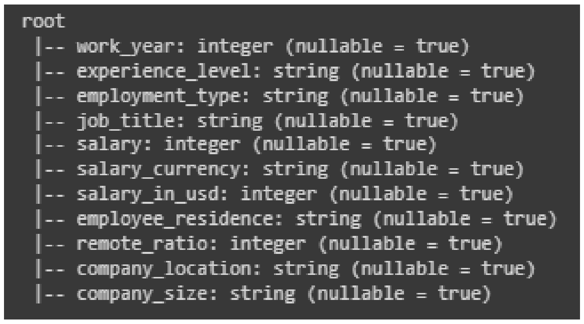
### ETL Process (Explanation)

Going into this Extraction, transformation, and loading of our data, we originally got our dataset from Kaggle, this data set contained around 3755 rows of data and had around 11 columns, making it a very manageable size to work with. I started out by using Apache Spark to import the file and had it give me a report on what the file contained in the rows as well as information about what columns were in the data set. As you can see below, I ran all the commands to run the CSV file through Spark.

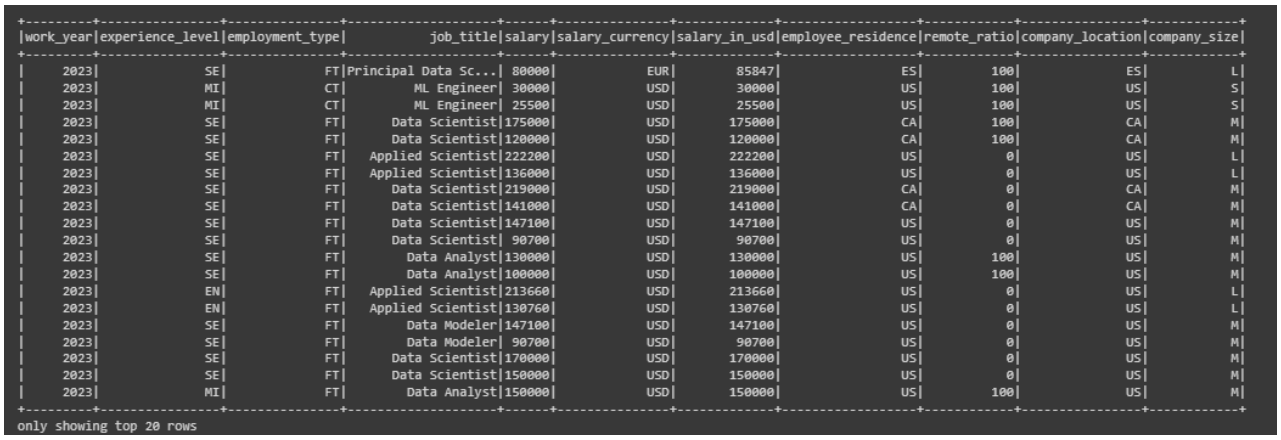


```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("SalariesAnalysis").getOrCreate()
df = spark.read.csv("/content/ds_salaries.csv", header=True, inferSchema=True)
df.printSchema()
df.show()
print("Number of records: {}".format(df.count()))
df.describe().show()

df.write.csv("/content/SparkDf", header=True, mode="overwrite")
spark.stop()
```

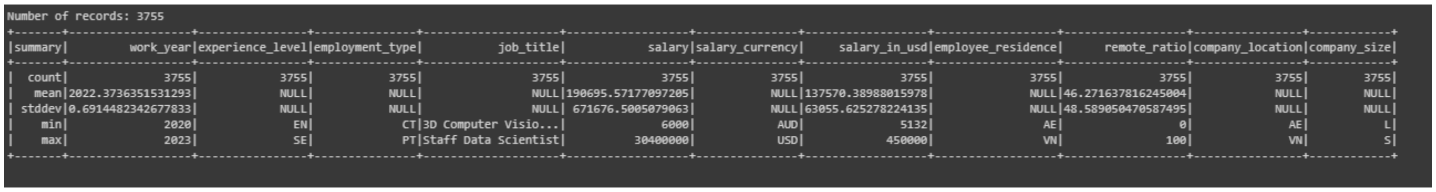
Then once I ran this I was given the Schema, number of records, description of data, and I also saved this new data frame into a file and named it SparkDf for the time being.



```
root
 |-- work_year: integer (nullable = true)
 |-- experience_level: string (nullable = true)
 |-- employment_type: string (nullable = true)
 |-- job_title: string (nullable = true)
 |-- salary: integer (nullable = true)
 |-- salary_currency: string (nullable = true)
 |-- salary_in_usd: integer (nullable = true)
 |-- employee_residence: string (nullable = true)
 |-- remote_ratio: integer (nullable = true)
 |-- company_location: string (nullable = true)
 |-- company_size: string (nullable = true)
```



```
+---------+----------------+---------------+--------------------+------+---------------+-------------+------------------+------------+----------------+------------+
|work_year|experience_level|employment_type|           job_title|salary|salary_currency|salary_in_usd|employee_residence|remote_ratio|company_location|company_size|
+---------+----------------+---------------+--------------------+------+---------------+-------------+------------------+------------+----------------+------------+
|     2023|              SE|             FT|Principal Data Sc...| 80000|            EUR|        85847|                ES|         100|              ES|           L|
|     2023|              MI|             CT|         ML Engineer| 30000|            USD|        30000|                US|         100|              US|           S|
|     2023|              MI|             CT|         ML Engineer| 25500|            USD|        25500|                US|         100|              US|           S|
|     2023|              SE|             FT|      Data Scientist|175000|            USD|       175000|                CA|         100|              CA|           M|
|     2023|              SE|             FT|      Data Scientist|120000|            USD|       120000|                CA|         100|              CA|           M|
|     2023|              SE|             FT|   Applied Scientist|222200|            USD|       222200|                US|           0|              US|           L|
|     2023|              SE|             FT|   Applied Scientist|136000|            USD|       136000|                US|           0|              US|           L|
|     2023|              SE|             FT|      Data Scientist|219000|            USD|       219000|                CA|           0|              CA|           M|
|     2023|              SE|             FT|      Data Scientist|141000|            USD|       141000|                CA|           0|              CA|           M|
|     2023|              SE|             FT|      Data Scientist|147100|            USD|       147100|                US|           0|              US|           M|
|     2023|              SE|             FT|      Data Scientist| 90700|            USD|        90700|                US|           0|              US|           M|
|     2023|              SE|             FT|        Data Analyst|130000|            USD|       130000|                US|         100|              US|           M|
|     2023|              SE|             FT|        Data Analyst|100000|            USD|       100000|                US|         100|              US|           M|
|     2023|              EN|             FT|   Applied Scientist|213660|            USD|       213660|                US|           0|              US|           L|
|     2023|              EN|             FT|   Applied Scientist|130760|            USD|       130760|                US|           0|              US|           L|
|     2023|              SE|             FT|        Data Modeler|147100|            USD|       147100|                US|           0|              US|           M|
|     2023|              SE|             FT|        Data Modeler| 90700|            USD|        90700|                US|           0|              US|           M|
|     2023|              SE|             FT|      Data Scientist|170000|            USD|       170000|                US|           0|              US|           M|
|     2023|              SE|             FT|      Data Scientist|150000|            USD|       150000|                US|           0|              US|           M|
|     2023|              MI|             FT|        Data Analyst|150000|            USD|       150000|                US|         100|              US|           M|
+---------+----------------+---------------+--------------------+------+---------------+-------------+------------------+------------+----------------+------------+
only showing top 20 rows
```



```
Number of records: 3755
+-------+------------------+----------------+---------------+--------------------+------------------+---------------+------------------+------------------+------------------+----------------+------------+
|summary|         work_year|experience_level|employment_type|           job_title|            salary|salary_currency|     salary_in_usd|employee_residence|      remote_ratio|company_location|company_size|
+-------+------------------+----------------+---------------+--------------------+------------------+---------------+------------------+------------------+------------------+----------------+------------+
|  count|              3755|            3755|           3755|                3755|              3755|           3755|              3755|              3755|              3755|            3755|        3755|
|   mean|2022.3736351531293|            NULL|           NULL|                NULL|190695.57177097205|           NULL|137570.38988015978|              NULL|46.271637816245004|            NULL|        NULL|
| stddev|0.6914482342677833|            NULL|           NULL|                NULL| 671676.5005079063|           NULL|63055.625278224135|              NULL|48.589050470587495|            NULL|        NULL|
|    min|              2020|              EN|             CT|3D Computer Visio...|              6000|            AUD|              5132|                AE|                 0|              AE|           L|
|    max|              2023|              SE|             PT|Staff Data Scientist|          30400000|            USD|            450000|                VN|               100|              VN|           S|
+-------+------------------+----------------+---------------+--------------------+------------------+---------------+------------------+------------------+------------------+----------------+------------+
```

I took this CSV file that was created and saved it into google Colab in order to make it available for everyone to manipulate and work with. From here Pandas was used with other libraries such as matplot and seaborn in order to visualize that dataframe we have now.

## Data Management

First, we used Apache Spark to handle the data. Spark gave us a clear picture of the CSV file – the schema, record count, and a quick data rundown. We saved this clarity as a Spark dataframe, calling it SparkDf. We then used Google Colab and took the CSV file there for easy collaboration. With Pandas, Matplotlib, and Seaborn, we made our data visualization using charts and plots, turning it into clear insights. This straightforward approach of Data Management helped us deep dive into data engineering salaries.

### Collection (Types, data sources)

Kaggle was the source of our data, we took a long time choosing a dataset as we wanted something that was relevant and something that was also accurate and not full of misinformation. This data set contains regularly updated information over the data engineering field and has achieved the highest reliability score allowed on Kaggle. This data set contained 11 columns that had Work Year, The year the salary was paid. Experience level, The experience level in the job. Employment type, the type of employment such as part time or full time or contract. Job Title, The role worked in during the year. Salary, The total gross salary amount paid. Salary currency, The currency of the salary paid as an ISO 4217 currency code. Salary in USD, The salary in USD. Employee residence, Employee's primary country of residence during the work year as an ISO 3166 country code. Remote Ratio, The overall amount of work done remotely. Company location, The country of the employer's main office or contracting branch. Company size, The median number of people that worked for the company during the year.

### Storage (Format, replication, security)

As seen above I took the files and ran them through Apache Spark and was able to get an overview of that data as well as a clean save of the data. Once I ran my CSV file through Spark I was able to get a description of all of the columns and was able to see the data and how reliable it was. Once my data was ready to be saved I saved it in Google Colab that only those who are involved in this project are able to access. The data once into Google Colab was not replicated but it was set to be unchangeable, meaning that if someone wanted to make some changes to the dataframe they had to save it under a new file and the original would not be changed no matter what changes were made.

## Data Engineering

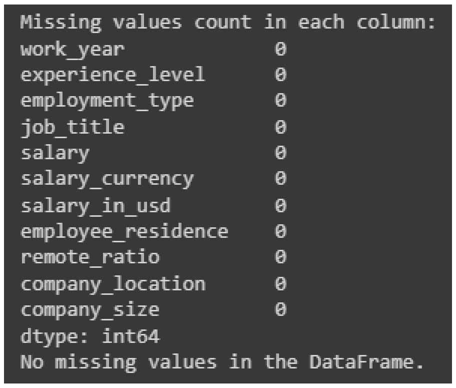
### Planning (Goals)

The dataset titled "Data Science Salaries 2023" obtained from Kaggle will undergo an examination, as part of the study on the impact of Data Engineering Salaries on Employment Factors. The objective is to unravel the relationships between employment characteristics and data engineering wages using state of the art techniques like statistical analysis and machine learning models. The primary aims include exploring the influence of work trends providing insights for decision making in the data engineering job market and understanding how experience, education, geography and company size affect salaries.

The significance of this study lies in its approach to showcasing salary patterns over a span of four years demonstrating the value of the selected dataset and offering insights at the intersection of data science and employment trends. To predict salary fluctuations based on criteria we will. Validate machine learning models. We acknowledge limitations such as data quality, resource constraints and temporal scope while emphasizing the reliability and relevance of our dataset. Ultimately our project aims to provide insights into the landscape of data engineering wages and employment by ensuring accuracy and dependability through an intuitive user interface along with validation, through test cases.

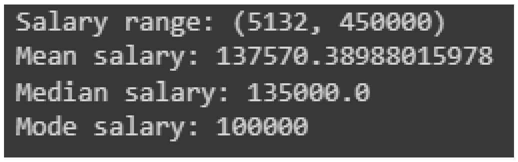
## Processing (Cleaning, transforming)

This data set when we received it had very little cleaning that needed to be done to it. Nonetheless the first thing that i did was i checked for any NA values after my initial cleaning of NA or blank values that i do to most dataset before i work with them and as you can see below we were left with no NA values in any row.



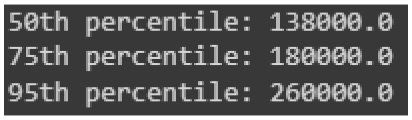
```
Missing values count in each column:
work_year            0
experience_level     0
employment_type      0
job_title            0
salary               0
salary_currency      0
salary_in_usd        0
employee_residence   0
remote_ratio         0
company_location     0
company_size         0
dtype: int64
No missing values in the DataFrame.
```

Going over the transformations that we made in the dataset, we started off with a simple one just checking the mean, median and mode of the salaries in USD and that can be seen below.



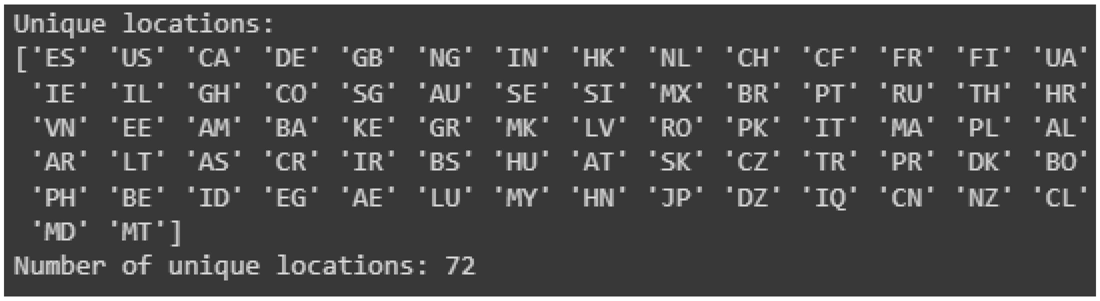
```
Salary range: (5132, 450000)
Mean salary: 137570.38988015978
Median salary: 135000.0
Mode salary: 100000
```

After that I wanted to see the percentiles of the salaries so I broke them down into the 50th, 75th, and 90th as you can see below.

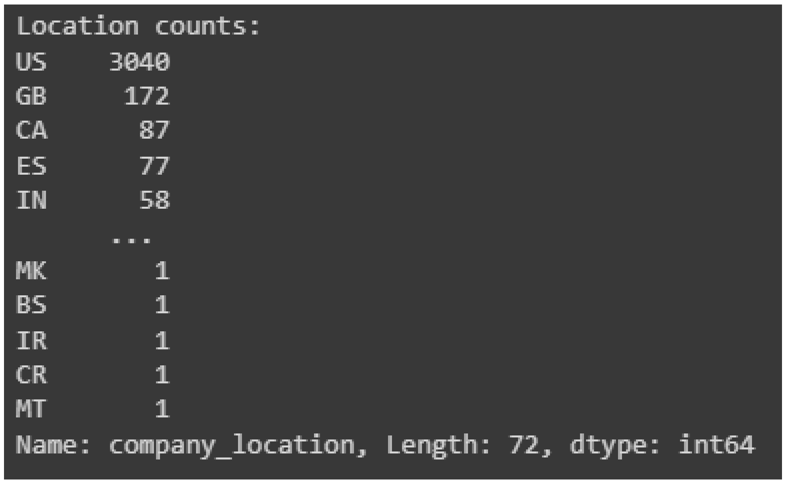


```
50th percentile: 138000.0
75th percentile: 180000.0
95th percentile: 260000.0
```

Finally before we got into any major visualizations I wanted the group to understand what things were in the data and just how diverse the data was so I checked for unique locations and count of companies in each country.



```
Unique locations:
['ES' 'US' 'CA' 'DE' 'GB' 'NG' 'IN' 'HK' 'NL' 'CH' 'CF' 'FR' 'FI' 'UA'
 'IE' 'IL' 'GH' 'CO' 'SG' 'AU' 'SE' 'SI' 'MX' 'BR' 'PT' 'RU' 'TH' 'HR'
 'VN' 'EE' 'AM' 'BA' 'KE' 'GR' 'MK' 'LV' 'RO' 'PK' 'IT' 'MA' 'PL' 'AL'
 'AR' 'LT' 'AS' 'CR' 'IR' 'BS' 'HU' 'AT' 'SK' 'CZ' 'TR' 'PR' 'DK' 'BO'
 'PH' 'BE' 'ID' 'EG' 'AE' 'LU' 'MY' 'HN' 'JP' 'DZ' 'IQ' 'CN' 'NZ' 'CL'
 'MD' 'MT']
Number of unique locations: 72
```



```
Location counts:
US    3040
GB     172
CA      87
ES      77
IN      58
      ...
MK       1
BS       1
IR       1
CR       1
MT       1
Name: company_location, Length: 72, dtype: int64
```

These transformations allowed us to see what the data was about and the things that my group was going into and allowed us to better know what was in the data and to know that we could trust the data because of how diverse it is. With this information that we gained here we were able to make all the visualizations that we made with that data.

## Validation

Given the projects focus on examining salary and employment related variables, in data engineering there are three validation processes for the ELT (Extraction, Loading, Transformation) pipeline

1. **Data Profiling**;
**Purpose**; Analyzing data distributions identifying outliers or discrepancies and understanding the structure of the dataset.

**Application to Project**: Analyzing the "Data Science Salaries 2023" dataset to assess the accuracy of pay records differences in experience levels, regional representation and other factors that influence salaries. This ensures that the dataset is robust for analysis.

2. **Schema Verification**:

**Purpose**: Ensuring that the data types, formats and restrictions align with the expected schema.

**Application to Project**: Validating that data in columns such as pay, experience level and company size is accurate and consistent.

3. **Data Quality Verification**:
**Purpose**:. Addressing missing values, duplicates and inconsistencies within the dataset. In addition we went online and found a reliable source taking a look at all the different averages of salaries in each of the fields that we were going over. According to US News we are able to see all the average salaries that we came up with in our data set were within 5% of the average salaries that were posted on the US News Database for jobs in our scope. With this we knew that this data was very credible and we could believe the information was accurate and help us come to a conclusion.

**Application to Project:** Implementing quality control measures by detecting disparities in pay records, handling incorrect values appropriately and ensuring consistency across fields.

These validation processes play a role in evaluating the reliability of data used for analysis purposes. To ensure the accuracy of the analysis and conclusions derived from the project's study, on data engineering salaries and their influencing factors it is crucial to implement these validation procedures that guarantee the accuracy, comprehensiveness and dependability of the dataset.

## Data Analytics and Modeling

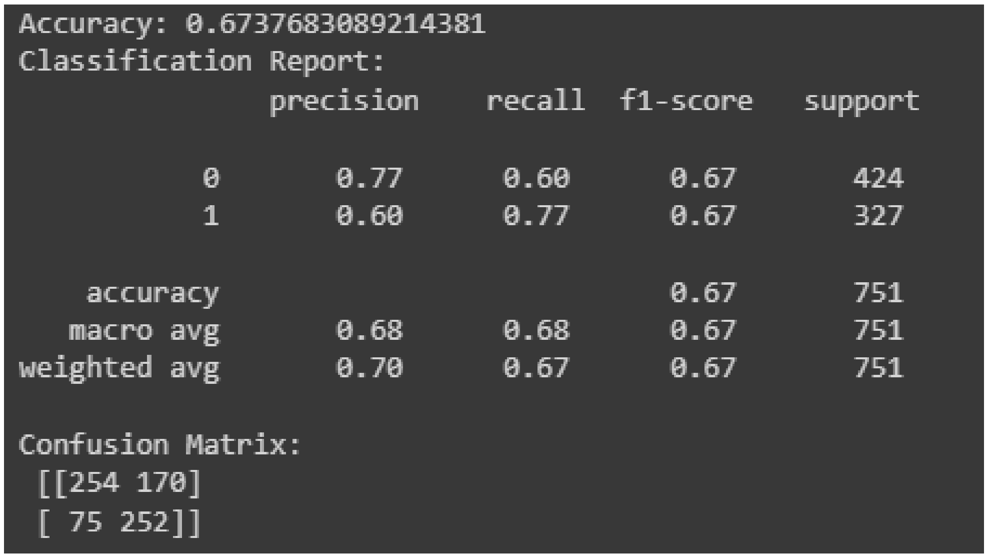
### Descriptive / Predictive

Descriptive and predictive elements are crucial to the study project "Effects of Data Engineering Salaries on Employment Factors," as they help to clarify the intricate connections between data engineering wages and employment factors. Understanding historical data is necessary for descriptive analytics to provide insightful information. This includes, but is not limited to, illustrating the influence of variables such as remote work on compensation adjustments, presenting fluctuations in average pay, and analyzing the salary distribution across various job titles and experience levels.

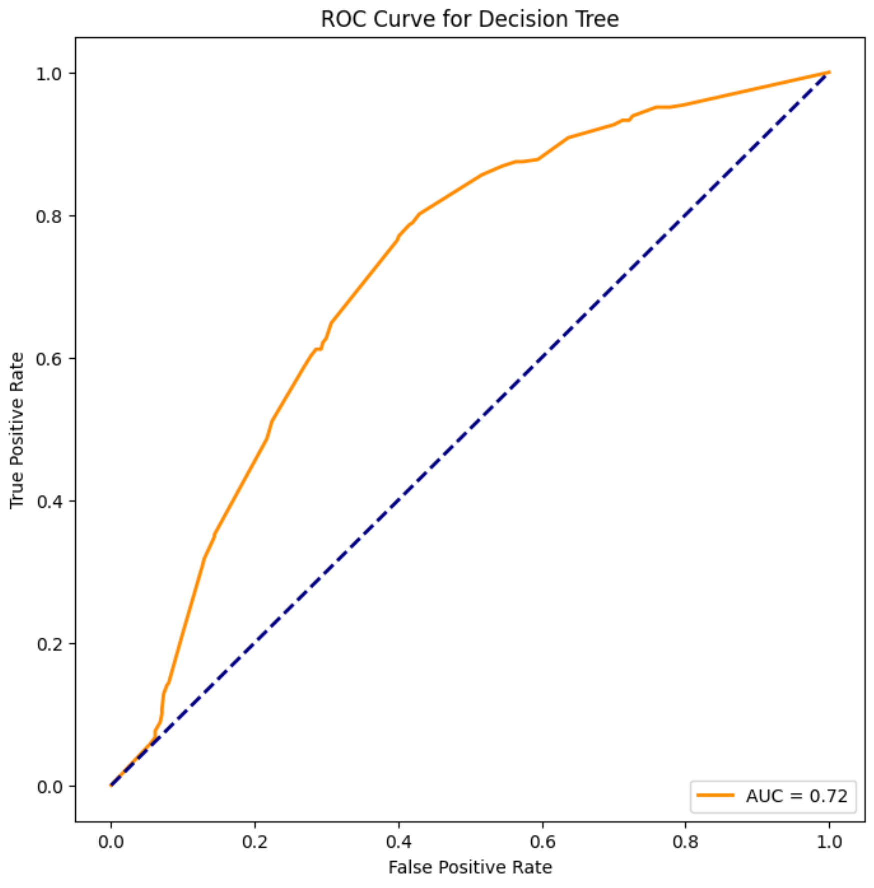
Predictive analytics, on the other hand, is concerned with projecting future trends and results from past data patterns. In order to forecast compensation fluctuations, machine learning models must be developed and validated.This project's predictive component comprises building and evaluating models like Random Forest, Logistic Regression, and Decision Trees. Based on characteristics such as experience level, job title, firm size, and remote work ratios, these models intend to forecast if people fall above or below the 50th percentile in terms of compensation. Decision-makers are empowered to strategically plan within the labor market environment by using the prediction models, which not only give accuracy metrics but also insights into the most critical aspects driving data engineering wages.

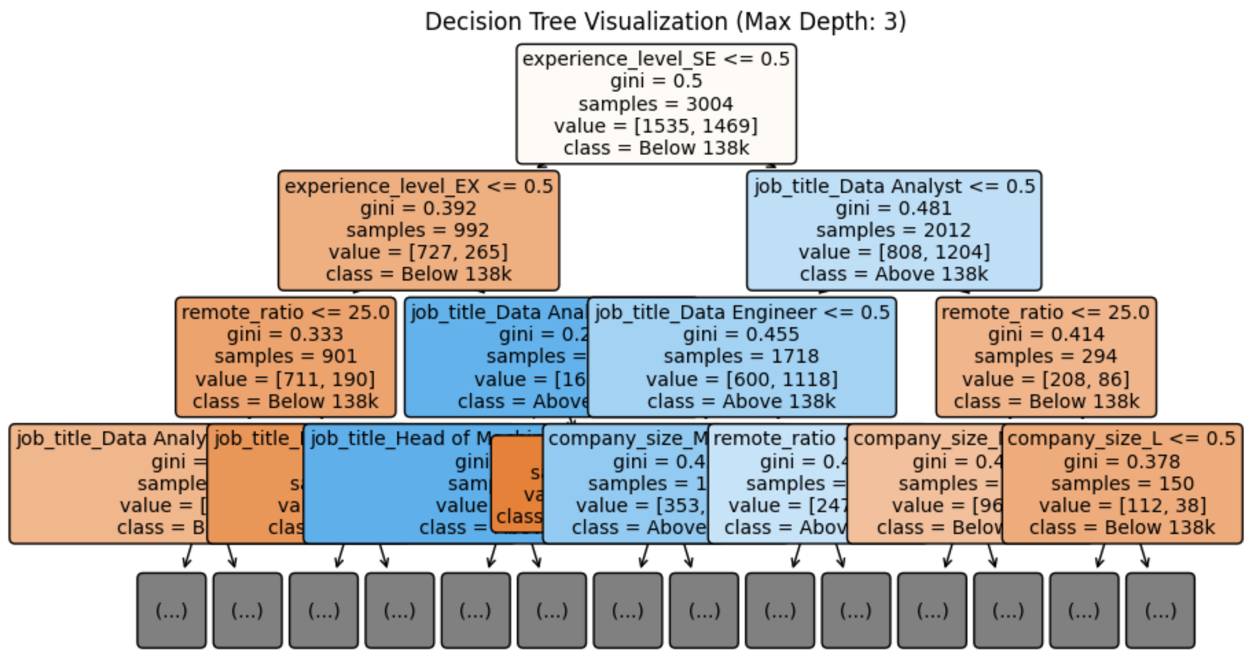
## Statistical / ML

Looking at the dataset the first thing that came to my mind when creating a model was trying to decipher if a person was either above or below the 50th percentile, in this context if they are above the 50th percentile they would be a high earner and if they were below it then they would be a low earner. Starting out I wanted to try a Decision Tree, I started out by choosing the features that would guide the model to the highest results with what we were given. I choose Remote Ratio, Company size, Job title, and experience level. Getting into this I ran into the issue of the strings in some of the features not being able to be converted into floats so what I did was i One Hot Encoded the values to be able to run them though my tree. With all of this I was able to achieve an accuracy score of 67%.In addition to the ROC curve and results I also added just a visualization of what the tree looks like. The Tree takes the features I have chosen and bases the tree on them being a Yes or No and sending it down the path it follows to get the most homogeneous results. You might say this isn't a particularly high score but this isn't the best model for this dataset as you will see next.

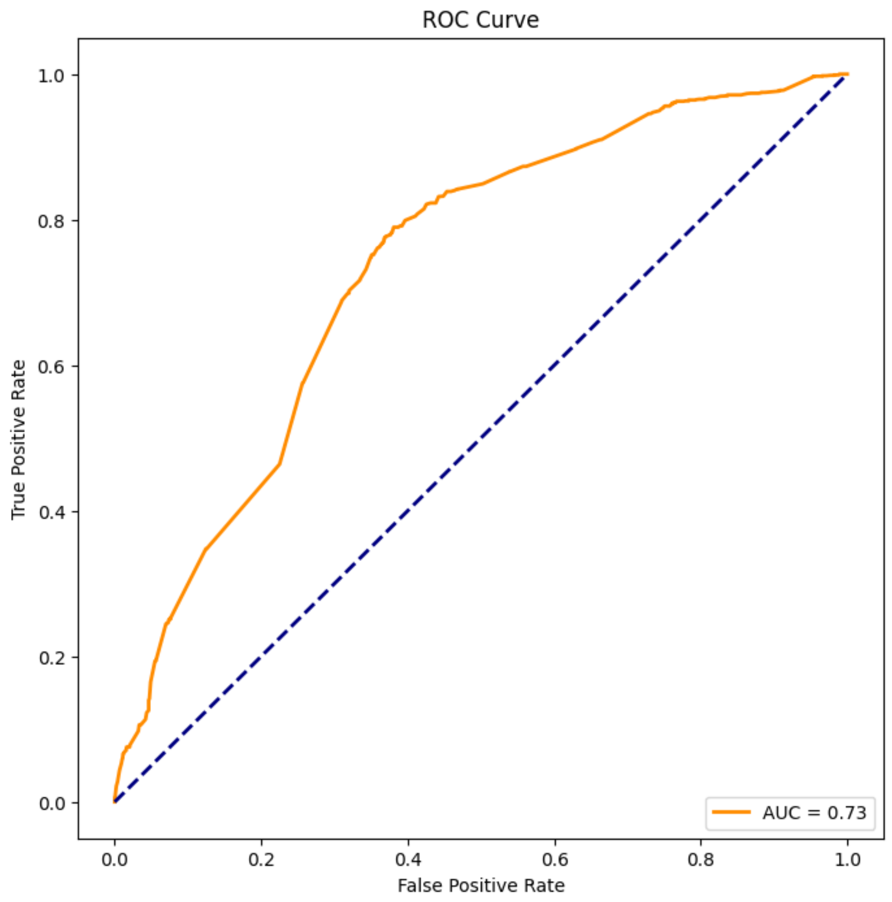


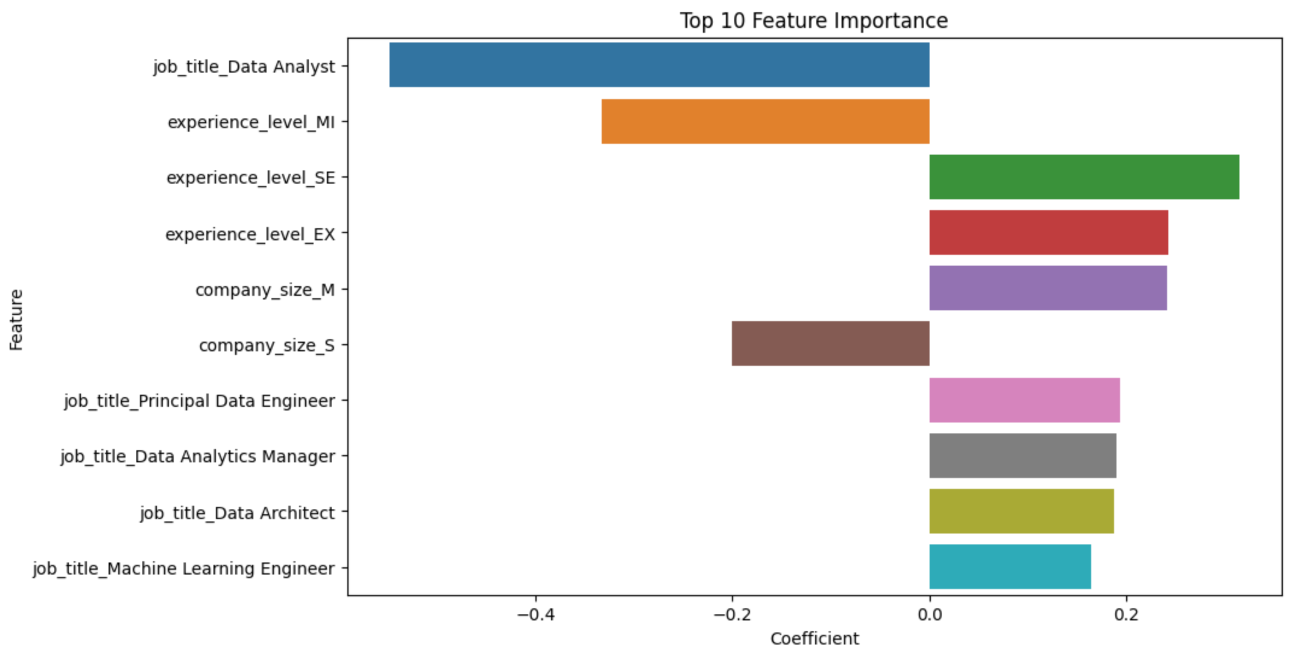
```
Accuracy: 0.6737683089214381
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.60      0.67       424
           1       0.60      0.77      0.67       327

    accuracy                           0.67       751
   macro avg       0.68      0.68      0.67       751
weighted avg       0.70      0.67      0.67       751

Confusion Matrix:
 [[254 170]
 [ 75 252]]
```


ROC Curve for Decision Tree
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC = 0.72

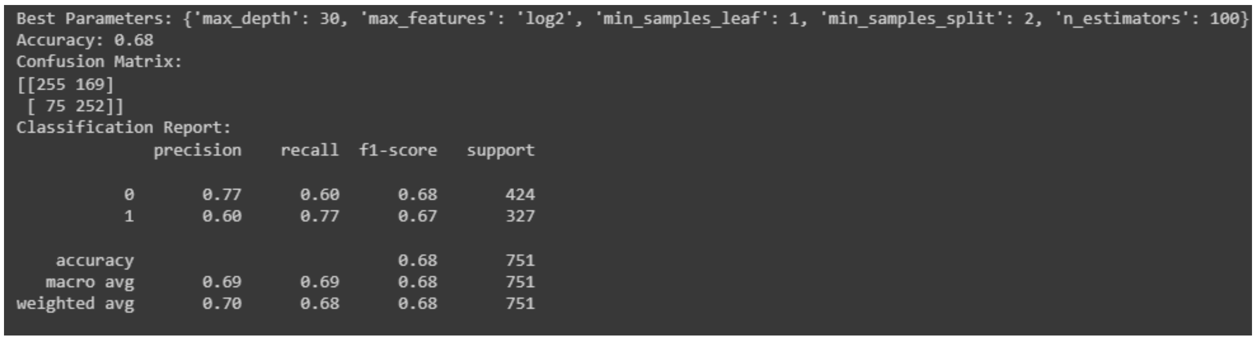

Decision Tree Visualization (Max Depth: 3)
experience_level_SE <= 0.5
gini = 0.5
samples = 3004
value = [1535, 1469]
class = Below 138k
experience_level_EX <= 0.5
gini = 0.392
samples = 992
value = [727, 265]
class = Below 138k
job_title_Data Analyst <= 0.5
gini = 0.481
samples = 2012
value = [808, 1204]
class = Above 138k
remote_ratio <= 25.0
gini = 0.333
samples = 901
value = [711, 190]
class = Below 138k
job_title_Data Engineer <= 0.5
gini = 0.455
samples = 1718
value = [600, 1118]
class = Above 138k
remote_ratio <= 25.0
gini = 0.414
samples = 294
value = [208, 86]
class = Below 138k
company_size_L <= 0.5
gini = 0.378
samples = 150
value = [112, 38]
class = Below 138k
(...)

Going forward I tried a logistic regression model as this by definition would be the model that is better suited for this type of dataset. The first thing I did was hyperparameter tuning. I ran code in order to take care of cases of oversampling, undersampling and using synthetics. From here I was able to find the best parameter and then ran my Logistic Regression model based on that. With that This model was able to achieve an accuracy score of 69%. While this might not seem like a massive difference it is still a gain over the last model that I used. Taking a look at some of the visualizations we can see that the ROC curve of this model looks better than that of the one in the Decision Tree.I also wanted to help show what the Parameter Tuning was so I visualized what features it found to be the best as well.

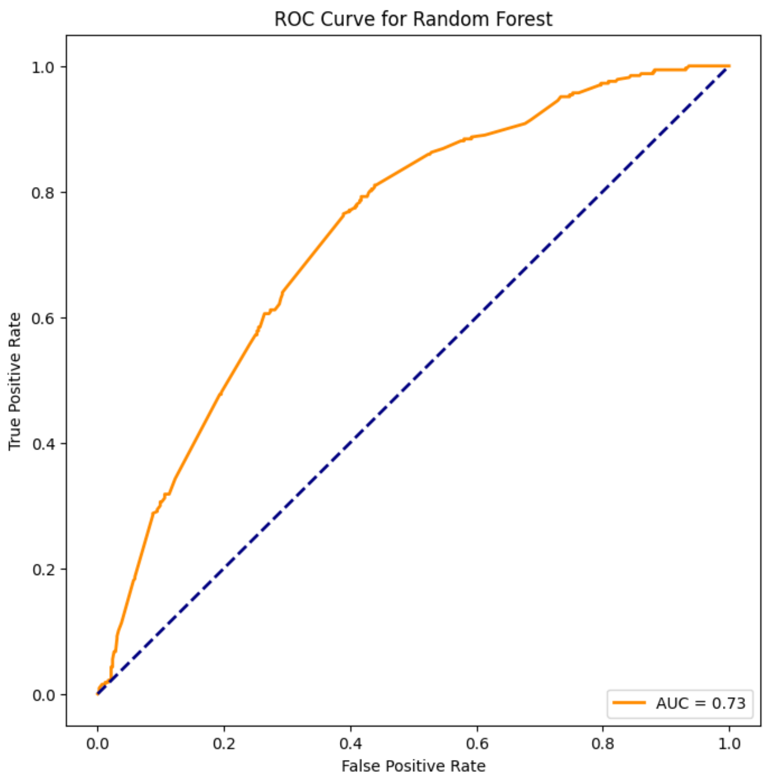




Lastly I wanted to use a very computationally costly model to see if we could reach an even higher score. I decided to try a Random Forest and I let it find the best parameters by creating a parameter grid and looking for all the elements to achieve the best score possible. With this we were able to achieve an accuracy of about 68% which is higher than the Decision Tree but taking a look at complexity we can see on the machine we tested this on, the tree took about 25 seconds to output meanwhile the Random Forest took about 10 minutes to output its results. While this does yield a better score we can still see that the score to computational complexity is not a good trade off.



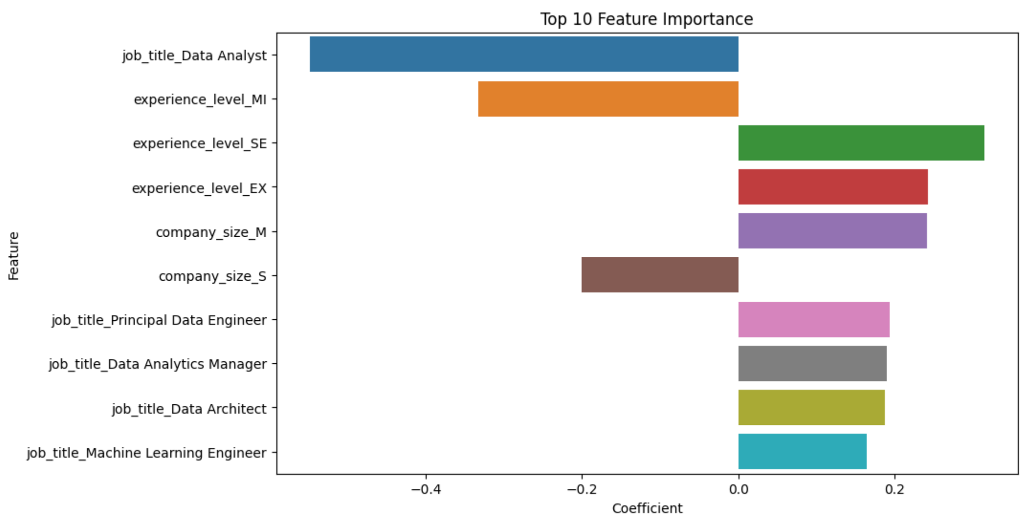
```
Best Parameters: {'max_depth': 30, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Accuracy: 0.68
Confusion Matrix:
[[255 169]
 [ 75 252]]
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.60      0.68       424
           1       0.60      0.77      0.67       327

    accuracy                           0.68       751
   macro avg       0.69      0.69      0.68       751
weighted avg       0.70      0.68      0.68       751
```



In conclusion when we take a look at the three different models that we attempted we can see that on this particular dataset, logistic regression was the best option and the model that yielded the best results and also was one of the least computationally complex. Taking the best Model, logistic regression into account we can see that our best accuracy was 69%.

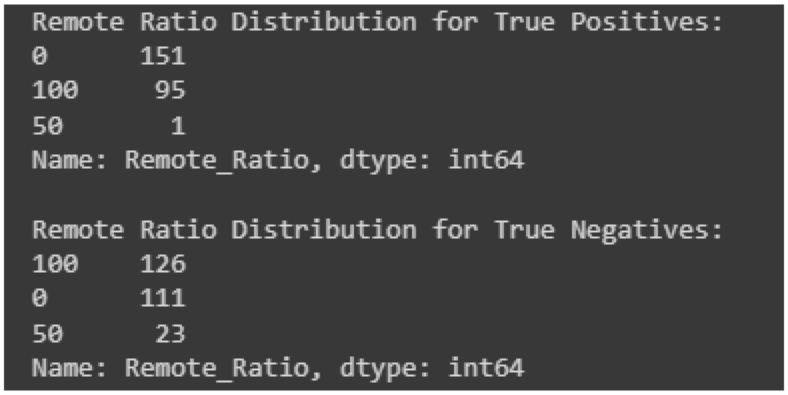
## Understanding the ML Models significance

Having a working model that will sort people in my data set to either above or below the 50th percentile is a good thing, but understanding what we can get out of this is what the most important factor is here. When taking another look at the feature importance we can see once the parameter tuning was run it gave us back the most influential factors in our data set.

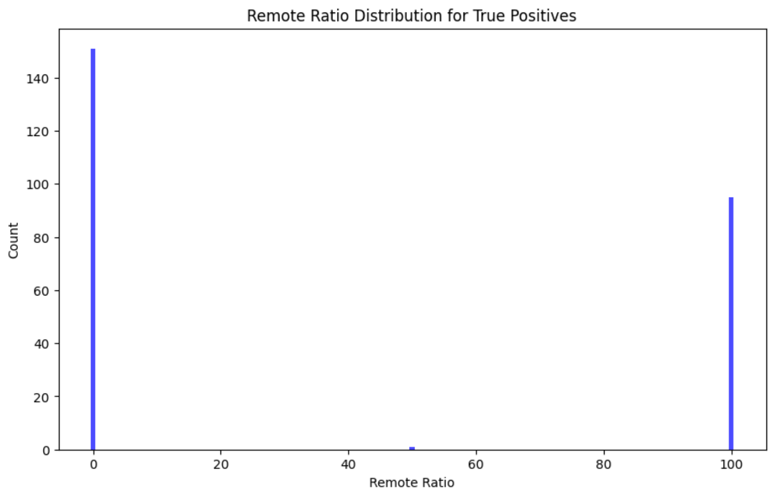


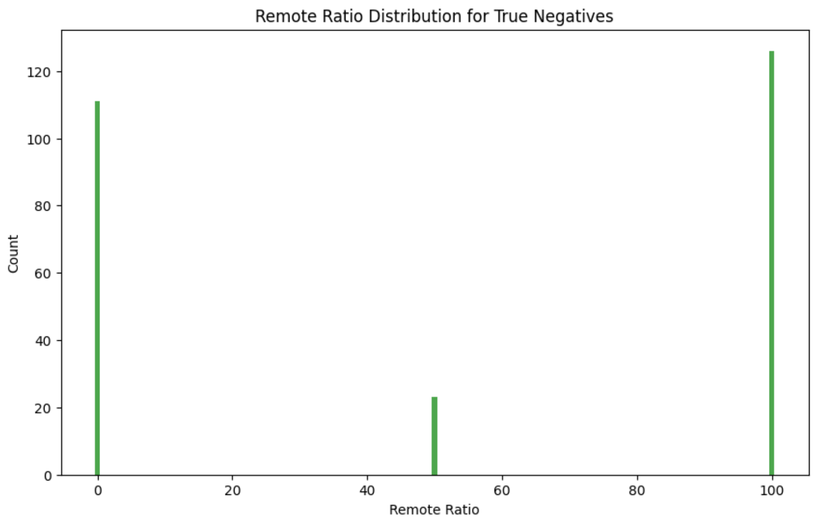
We can see here that the main driving factor in the linear regression model that was taken into account were the different levels of experience. Naturally with more experience people expect to be paid more, those who are starting out will be paid much less than those who are senior levels in their field. Looking at another factor, we can see that company size also was a factor that appeared a few times, it took into account smaller companies and medium sized companies when trying to run the regression model. Lastly, when you take a first glance at this you notice that the job title is the one that appears most often, when we recall the visualization where it was shown the top 10 salaries, we see the variance there is amongst them allowing this model to take into account that variance and be able to classify the results.

The next thing I wanted to look at was the remote ratio amongst our true positives and true negatives, in our case a true positives would be someone that does earn above the 50th percentile and the models identified them y and a true negative would be someone that makes below the 50th percentile and was identified correctly as well. correctly. While Remote Ratio was not an element that the parameter hypertuning picked up as significant, the Trees both chose them and achieved a very close score. Taking a look at our results below we can see that the higher earners had a much higher number of people who did not have any remote work at all and as we know from previous visualizations, those who did not have any were on average the highest of all remote ratios. Next taking a look at the 50% remote ratio we can see that lower earners had a significantly higher number of these and again as we know from above that those who are 50% remote had the lowest salaries by far.



```
Remote Ratio Distribution for True Positives:
0      151
100     95
50       1
Name: Remote_Ratio, dtype: int64

Remote Ratio Distribution for True Negatives:
100    126
0      111
50      23
Name: Remote_Ratio, dtype: int64
```



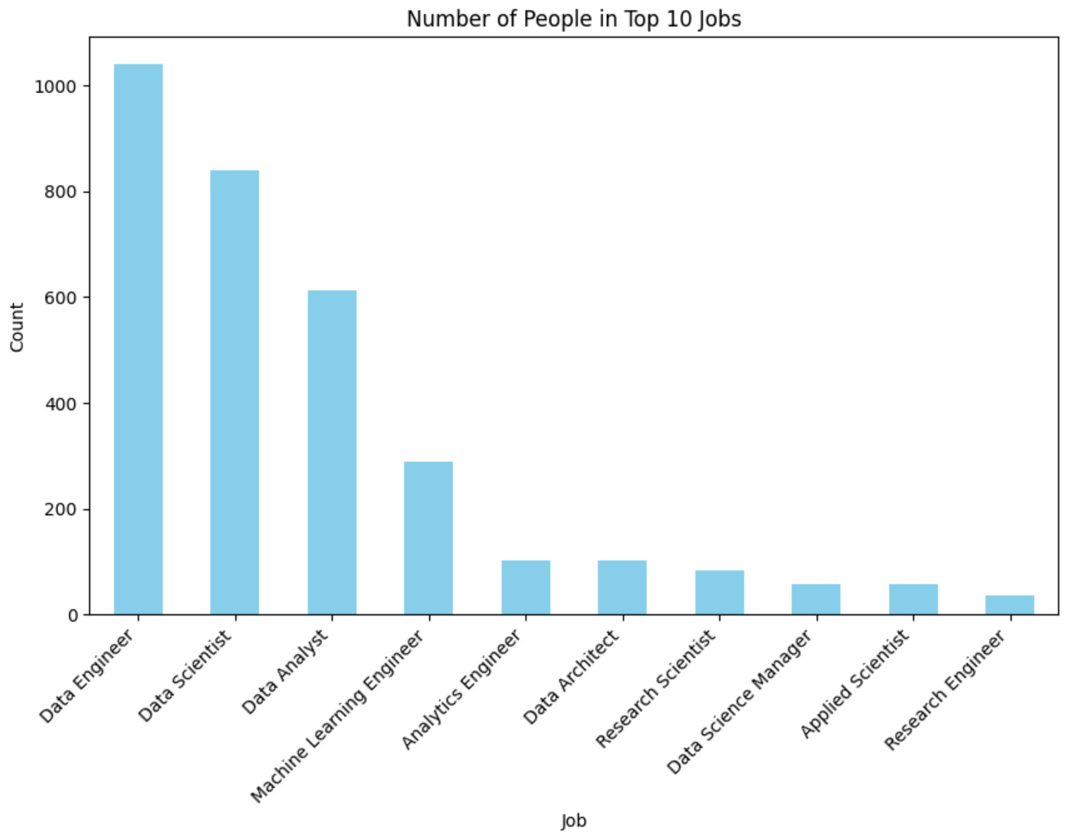


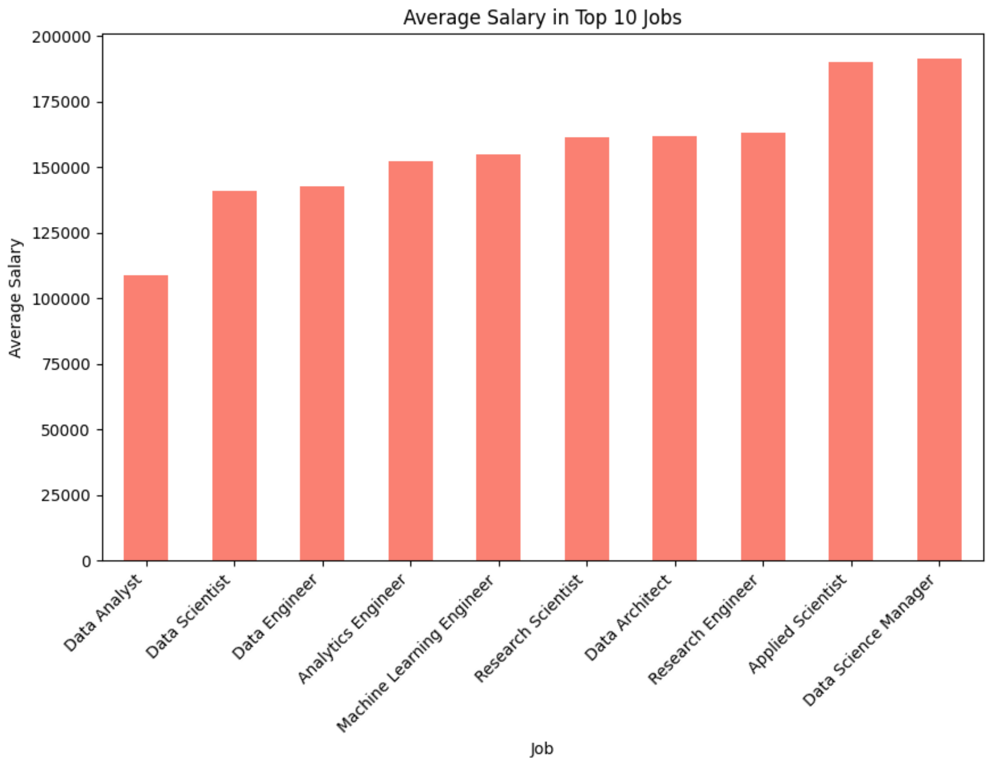
## Scalability/Reason

With all these things over the significance of the Models and how the features are important, I want to tie it all together and let you see how these factors can be scaled into a model to understand growth and change of all salaries across all fields and why things are changing. As we saw above, the remote ratio had a massive effect on salaries and most people that were in person made a substantial amount of money more then those who are either remote or hybrid. The job market is a very difficult thing to understand and get right, but if there is a model such as the ones we have drafter were the persons qualities can be put in and sorted in a way such that they are given a range where they should be paid, i feel like we will have a more accurate system to both assinge salaries and evaluate change on salaries as time has passed.

## Data Visualization

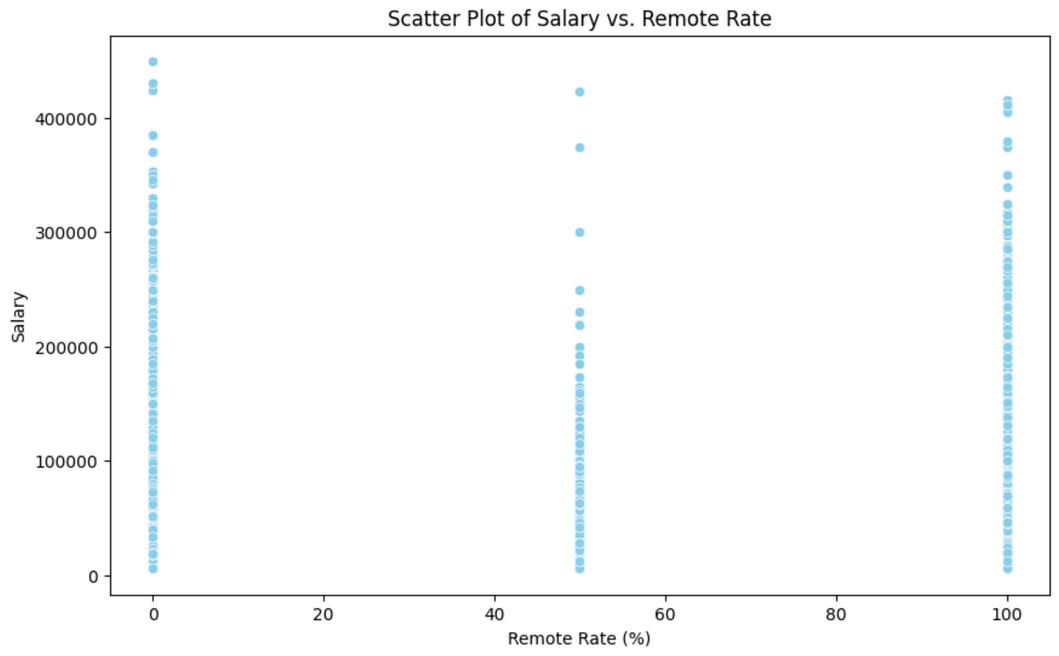
In the subject of visualization there were many things that i could visualize but i had to limit myself to relevant things that will make it clear there is a major trend in the data engineering salaries. Starting off I wanted to go over the fields, with the visualizations below we can see the top 10 jobs by count and in the following one are able to see how much the top 10 jobs are paid.



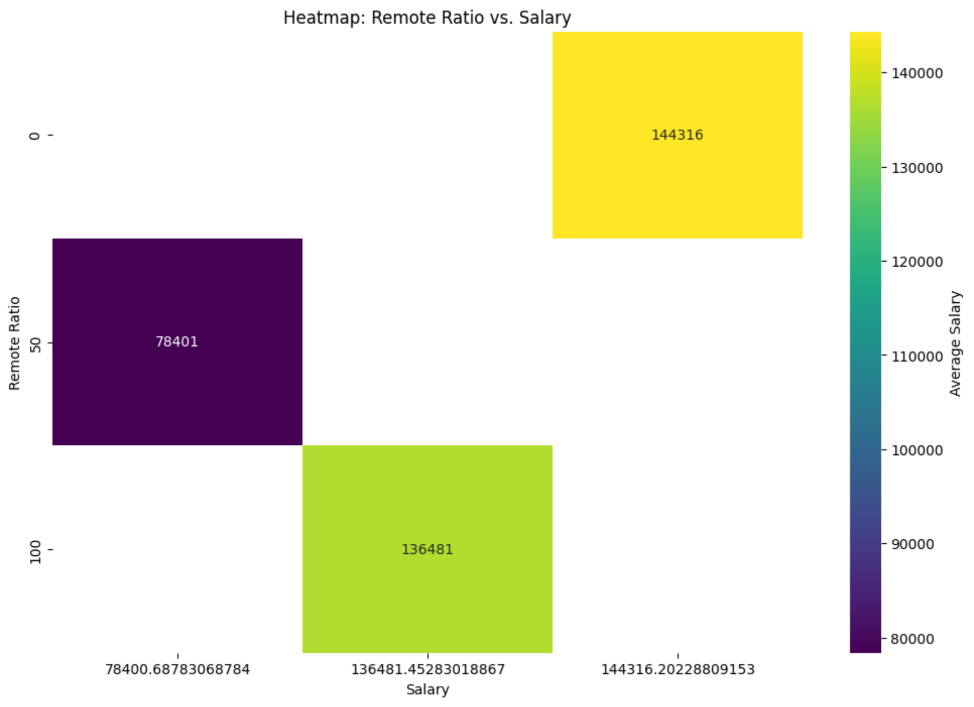


From this photo we can see that even though Data engineers are the most prevalent field they are not the most paid, the most paid would be Data Science Managers that are among one of the lower counts in the top 10 jobs by count in our first visualization.

Moving forward during the year 2020 to 2021 the world was impacted by the COVID-19 pandemic and a lot of people were forced to work from home, while this was nice for most people, people did experience a slight pay drop on average. This trend not only continued, but proved itself for those that are working a hybrid role, as you can see below, those who work in person on average are paid higher and have a larger density around the top of the pay scale while those who are hybrid are significantly lower on the scale.



Since this may be difficult to get the true meaning of, I also created a heat map that will allow you to better see the values. As we can see the hybrid roll was significantly lower.



After looking at these visualizations we can come to the conclusion that the online period took a toll on salaries and caused the average to be below those who are in person during their entire work day. Even as we have shifted away from this idea of being remote after the pandemic there are still those who stay online and who split their time between online and in person. Once we took a look at the heat map and scatter plot we can see just how much of a difference there actually is.

## Code

With around 400 lines it would span around 10 pages if i talked through all of it so i'm going to just go over a few of the things that we did and came up with.

### Data Preprocessing

- **Pyspark ingestion of data**

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("SalariesAnalysis").getOrCreate()
df = spark.read.csv("/content/ds_salaries.csv", header=True, inferSchema=True)
df.printSchema()
df.show()
print("Number of records: {}".format(df.count()))
df.describe().show()
df.write.csv("/content/SparkDf", header=True, mode="overwrite")
```

We started out by taking the dataset that we had that had around 3755 rows of data, with knowing this we knew that we would not have to use HDFS and could just go straight to using Pyspark. We saved the newly created session into a folder on the google Colab, if you want to see proof of this you will be able to see the file saved if ran on IPBYN File. From this point we were able to use things such as pandas to manipulate the data and continue without data preprocessing.

- **Checking for NA values and Quality of data**

```
missing_values = df.isna()
missing_values_count = missing_values.sum()
print("Missing values count in each column:")
print(missing_values_count)
if missing_values.any().any():
   print("There are missing values in the DataFrame.")
else:
   print("No missing values in the DataFrame.")
```

Checking for NA values was very crucial here as if there was any it would ruin the integrity of the data, if there was any NA values found we would take them and drop them from the dataset. There might be a question of why we wouldn't try and fill them in with either the average or the other values? We wanted to see the true changes of what things made these salarries be the way they are so we did not want the chance of the results being skewed because of information that we filled in.

```
locations = df['company_location'].unique()
print("Unique locations:")
print(locations)
num_locations = len(locations)
print("Number of unique locations:", num_locations)
```

Thus our data validation process began, we wanted to check first how many unique locations there were for company location to be able to understand how diverse our data truly was.

```
salary_column = df['salary_in_usd']

salary_range = salary_column.min(), salary_column.max()
print("Salary range:", salary_range)


mean_salary = salary_column.mean()
median_salary = salary_column.median()
mode_salary = salary_column.mode().iloc[0]

print("Mean salary:", mean_salary)
print("Median salary:", median_salary)
print("Mode salary:", mode_salary)
```

Similarly to above we wanted to see how much the salaries ranged to further reassure us that our data set was one that was diverse and going to yield us a good platform for investigating the effects of salaries based on features in the data set.

- **Machine learning and Modeling**

We had three Models that we used, a Decision Tree, Random Forest, and a logistic regression. Since the logistic regression yielded us the highest results I will be covering that one in the code review.

```
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}
model = LogisticRegression()

grid_search = GridSearchCV(model, param_grid, cv=5, scoring='accuracy')
```

```
grid_search.fit(X_train_scaled, y_train)

best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

print(f'Best Parameters: {best_params}')
y_pred = best_model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{classification_rep}')
```

The first step that we took is we let the logistic regression model find the best fit for the model and it would run through all different parameters in order to see what would be the best fit, once this fit was found we were able to run the model and this gave us the results of a accuracy of a 69%. While this might not seem like an overall high score you must keep in mind that our data size was not a large one to start out with. We only had around 3755 rows to work with and with more data that number could be boosted to a much greater score. Going over the time complexity a bit, compared to the other models that we ran this one was by far the most efficient. The Random Forest took almost 15 minutes to find the best fit and gave us the results of the test, and in the end it only yielded a 1% higher score then that of the DT. With all of these results we generated all of their ROC curves and visualized their results as you have seen in the report.

## Analytical Outcomes

1. Data Exploration:
    - Explored and visualized the "Data Science Salaries 2023" dataset, showcasing the diversity of variables such as work year, experience level, employment type, job title, salary, salary currency, remote ratio, company location, and company size.
2. Trend Analysis:
    - Investigated the average salary trends over time, revealing a significant growth in data engineering salaries from 2020 to 2023, even considering external factors like the COVID-19 pandemic.
3. Effect of Remote Work on Salaries:

    - Visualized and analyzed the impact of remote work on salaries, indicating that in-person roles tend to have higher average salaries compared to hybrid or fully remote roles.
4. Job Title Influence:
    - Identified job titles with the highest count and corresponding average salaries. Revealed that while Data Engineers are the most prevalent, Data Science Managers have the highest average salaries.
5. Machine Learning Models:
    - Explored and implemented three machine learning models (Decision Tree, Logistic Regression, Random Forest) to predict whether an individual's salary is above or below the 50th percentile.
    - Logistic Regression demonstrated the highest accuracy (69%) and was chosen as the most suitable model.
6. Feature Importance:
    - Analyzed feature importance in the Logistic Regression model, highlighting the significance of experience level, company size, and job title in predicting salary ranges.
7. Remote Ratio Impact:
    - Investigated the impact of remote ratio on true positives and true negatives, indicating that higher earners are more likely to have no remote work, while lower earners often have a 50% remote ratio.
8. Scalability and Significance:
    - Explored how models and features' significance can be scaled to understand salary growth and changes across various fields in data engineering.
    - Emphasized the significance of models in providing accurate salary range predictions.
9. Data Visualization:
    - Created visualizations showcasing the top 10 job titles by count and their corresponding average salaries.
    - Visualized the effect of the COVID-19 pandemic on salaries, highlighting the lower average salaries for hybrid roles compared to in-person roles.

## Test Cases

Normally test cases are tests that you run your code to see if it is running correctly and efficiently, in our case here we have a dataset that we did transformations on and ended up making a model to be able to better understand what parameters followed either above or below the 50th percentile. In our models we let the model run parameter tuning and to see what factors would be the best to run and would verify these things based on the scores they get on the fitness model. Then these factors that were checked already were placed into our regression and tree models to give us the accuracy. The test case also is the score that was given to the models performance, in our cases we were able to get a high of a 69% accuracy with logistic regression.

**Random Forest**

```
X = pd.get_dummies(df[['remote_ratio', 'company_size', 'job_title',
'experience_level']], columns=['job_title', 'company_size', 'experience_level'],
prefix=['job_title', 'company_size', 'experience_level'])
df['Above 138k'] = df['salary_in_usd'].apply(lambda x: 1 if x > 138000 else 0)
X_train, X_test, y_train, y_test = train_test_split(X, df['Above 138k'],
test_size=0.2, random_state=42)

param_grid = {
   'n_estimators': [50, 100, 150],
   'max_depth': [None, 10, 20, 30],
   'min_samples_split': [2, 5, 10],
   'min_samples_leaf': [1, 2, 4],
   'max_features': ['auto', 'sqrt', 'log2']
}


rf_model = RandomForestClassifier(random_state=42)
grid_search_rf = GridSearchCV(rf_model, param_grid, cv=5, scoring='accuracy')
grid_search_rf.fit(X_train, y_train)

best_params_rf = grid_search_rf.best_params_
best_model_rf = grid_search_rf.best_estimator_
print(f'Best Parameters: {best_params_rf}')
y_pred_rf = best_model_rf.predict(X_test)

accuracy_rf = accuracy_score(y_test, y_pred_rf)
conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)
classification_rep_rf = classification_report(y_test, y_pred_rf)

print(f'Accuracy: {accuracy_rf:.2f}')
print(f'Confusion Matrix:\n{conf_matrix_rf}')
print(f'Classification Report:\n{classification_rep_rf}')
```

This is one of the test cases that we did, we selected the rows that we wanted to run the tests on and then let the random forest find the best fit for all of the trees. Once it found the best fit for the trees, it would run that fit for a few different iterations in order to find the majority vote to see what formation of a tree would be the best fit. With this, it was a very time consuming process, it took around 15 minutes to run and provide results with this test. The results were an accuracy of 68%, only 1% higher than that of the regular decision tree.

## References

- **Kaggle. (2023). Data Science Salaries 2023. Kaggle. https://www.kaggle.com/datasets/arnabchaki/data-science-salaries-2023/data**
- **Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, 281-305.**
- **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.**
- **Pandas documentation -** `pandas.DataFrame.plot` **pandas.DataFrame.plot documentation**
- **Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. Link to Paper**
- **House, A. (n.d.). *Data scientist salary | US News Best Jobs*. Data Scientist Overview. https://money.usnews.com/careers/best-jobs/data-scientist/salary**