Project Report

Project Title: Salaries for San Francisco Employee

1. Project Overview

This project predicts employees salary using compensation data across various departments. The goal is to uncover insights into high performing jobs and employees in organization based on these data for strategic business decisions.

2. Dataset Summary

- Rows: 312882- Columns: 12

Key Features: - 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',

'Benefits' , 'TotalPayBenefits', 'Year'.

Missing Data: No column has missing values but unnamed columns with “Nan” values are removed.

3. Exploratory Data Analysis using Python

We began with data preparation and cleaning in Python:

- **Data Loading:** Imported the dataset using pandas.
- **Initial Exploration:** Used df.info() to check structure and .describe() for summary statistics.

```
1 df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 312882 entries, 0 to 312881
Data columns (total 8 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   EmployeeName      312882 non-null  object
 1   JobTitle          312882 non-null  object
 2   BasePay           312882 non-null  object
 3   OvertimePay       312882 non-null  object
 4   OtherPay          312882 non-null  object
 5   Benefits          312882 non-null  object
 6   Year              312882 non-null  int64
 7   TotalPayBenefits  312882 non-null  float64
dtypes: float64(1), int64(1), object(6)
memory usage: 19.1+ MB
```

```
In [17]: 1 df2.describe()
```

Out[17]:

	BasePay	OvertimePay	OtherPay	Benefits	Year	TotalPayBenefits	JobTitle_Numeric
count	312882.000000	312882.000000	312882.000000	312882.000000	312882.000000	312882.000000	312882.000000
mean	69672.670978	5668.670240	3460.390329	22125.262623	2014.625303	100928.339777	1323.033377
std	45436.770132	12745.525827	7387.169347	16289.100115	2.290899	66485.186495	599.313038
min	-474.000000	-292.000000	-7058.000000	-13939.000000	2011.000000	-3628.780000	0.000000
25%	35341.250000	0.000000	0.000000	2079.000000	2013.000000	48955.072500	865.000000
50%	67645.500000	0.000000	728.000000	26771.000000	2015.000000	100011.290000	1485.000000
75%	99235.500000	5223.000000	3958.000000	34288.000000	2017.000000	142376.300000	1875.000000
max	592394.000000	309481.000000	400184.000000	125891.000000	2018.000000	712802.360000	2285.000000

Missing Data Handling: Checked for null values in each columns.

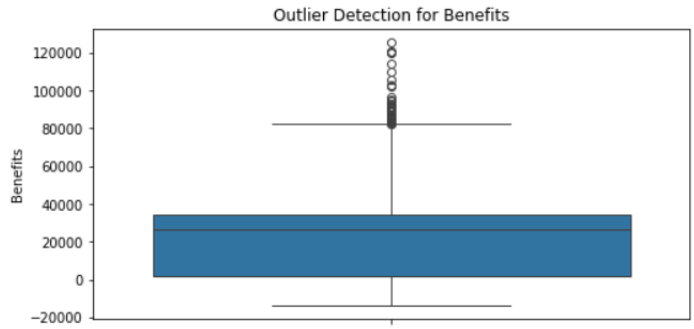
Column Standardization: Renamed columns to **snake case** for better readability and documentation and removed unwanted columns.

Feature Engineering: No need to do feature engineering.

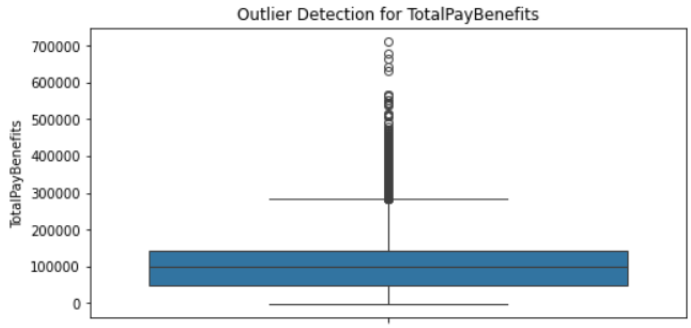
Data Consistency Check: Total pay and TotalPayBenefits contains same values so dropped TotalPay column.

Outlier handling: Detects outlier using boxplot method and removed outliers using IQR method.

OtherPay



Benefits

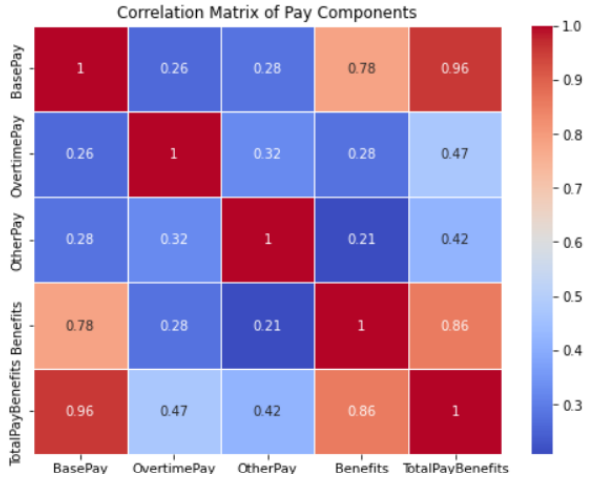


Data Visualization:

We determined correlation between independent and dependent variables using heatmap .

Considered the features which are highly correlated.

```
# Correlation matrix
plt.figure(figsize=(8,6))
corr_matrix = df[['BasePay', 'OvertimePay', 'OtherPay',
'Benefits', 'TotalPayBenefits']].corr()
# Plotting heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm',
linewidths=0.5)
plt.title('Correlation Matrix of Pay Components')
plt.show()
```



4.Model Selection & training :

we trained our dataset on two models i.e. linear regression and random forest regressor as data was continuous.

5.Model Evaluation:

After evaluation we got good result on linear regression as compare to random forest regressor.

6. Business Recommendations

The average base pay is highly correlated with total compensation.

Job titles like "Chief Executive Officer" have the highest salaries.