Salary Prediction Portfolio

Problem Definition:

The purpose of this project is to make accurate salary predictions that are based on existing known salaries, so the company can recruit and retain top talent. This model will help the company for offering competitive pay to existing and future employees. Data transformation and machine learning will be used to create a model that will predict a salary when given job description category, contract type, and contract time: The data for this model consists of a training dataset with the features listed below and their corresponding salaries. Twenty five percent of this training dataset was split into a test dataset with corresponding salaries so accuracy and error of the model can be determined.

Dataset:

train_rev1.csv: Each row represents an observation for each individual job posting. The "Category" column is unique to each job posting and the other columns are the different features of the job postings. The file has twelve columns.

Features Description:

- **Job Code: Primary Key - unique identifier for each job posting**
- **Title: - The job posting title**
- **Job Description: Job level description**
- **Region: Job posting regional area**
- **Location: Job posting location**
- **ContractType: Contract Posting (e.g Part-time, Full-Time)**
- **ContractTime: Job Posting Time (e.g Part-time, Permanent, Contract)**
- **Company: Hiring Company**
- **Category: Job level Category (e.g IT-Jobs, Engineering)**
- **Salary_Range: salary range for the job, in thousands UK dollars.**
- **Salary: Salary paid, in thousands UK dollars, target variable.**
- **•Source_Name: Source of Job posting**

Data Wrangling:

Cleaning

```python
def scrub_data(data, columns, axis, vals, names):
    #drop useless columns
    data = data.drop(columns, axis = 1)
    #detect the missing data
    missingcounts = data.isnull().sum()
    missingcols = [col for col in missingcounts.keys() if missingcounts[col] != 0]
    #fill the missing data with specific values
    values = dict(zip(missingcols, vals))
    data = data.fillna(value = values)
    #rename the columns
    data.columns = names
    return data
```

Checked for null values

```python
for column in train_data.columns:
  null_count = len(train_data[train_data[column].isna()])
  print("{} : {}".format(column, null_count))
```

```
Id : 0
Title : 1
FullDescription : 0
LocationRaw : 0
LocationNormalized : 0
ContractType : 179326
ContractTime : 63905
Company : 32430
Category : 0
SalaryRaw : 0
SalaryNormalized : 0
SourceName : 1
```

Checked for duplicate values

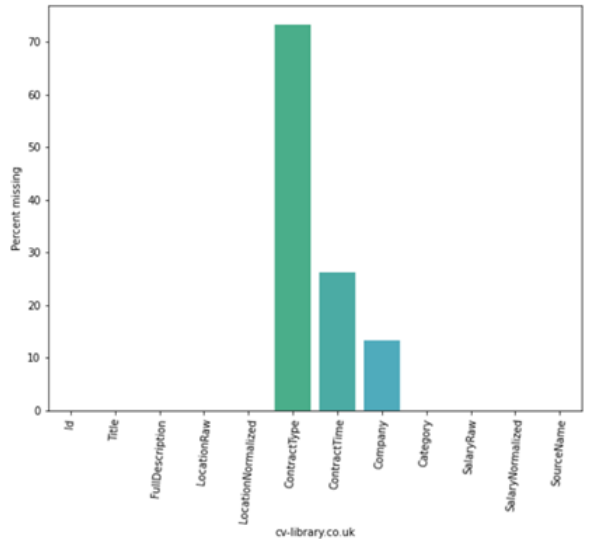
```python
train_data[train_data.duplicated(['Id', 'Title'])]
```

Id	Title	FullDescription	LocationRaw	LocationNormalized	ContractType	ContractTime	Company	Category	SalaryRaw	SalaryNormalized	SourceName

Number Of Missing Values Visualized

```python
null_vals = train_data.isnull().sum()/len(train_data)*100
null_vals = pd.DataFrame(null_vals)
null_vals.reset_index(inplace = True)
null_vals.columns = ["cv-library.co.uk","Percent missing"]
plt.figure(figsize = (9,7))
plt.xticks(rotation=85)
sns.barplot(x ="cv-library.co.uk",y ="Percent missing",data = null_vals)
```

```
<AxesSubplot:xlabel='cv-library.co.uk', ylabel='Percent missing'>
```



Explanatory Data Analysis:

Check for outliers Deleted the rows with salary < 0. Calculated the quantile value for 75% and 25%, the value above 75% - 200,000 tend to be outliers. Verified and validated the data as those rows are legitimate and those salary belongs to IT executive type jobs or from highly paid industry like oil and finance.

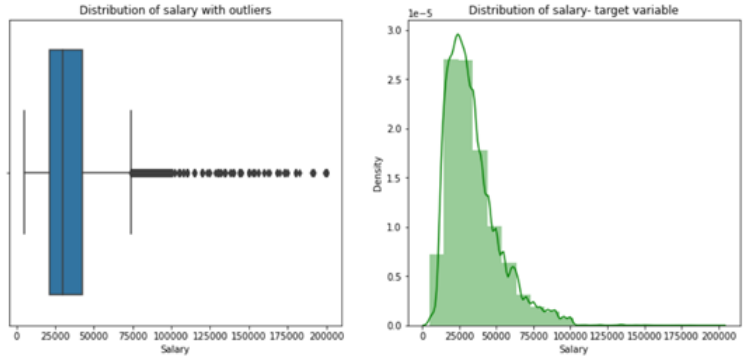


Fig X. Salary – target variable

In Fig X. we see, the target variable salary is somewhat right-skewed distribution, which is due to outliers and verified. Performed explanatory data analysis to learn more about the relation between each feature and the target variable

Salary versus Job Industry

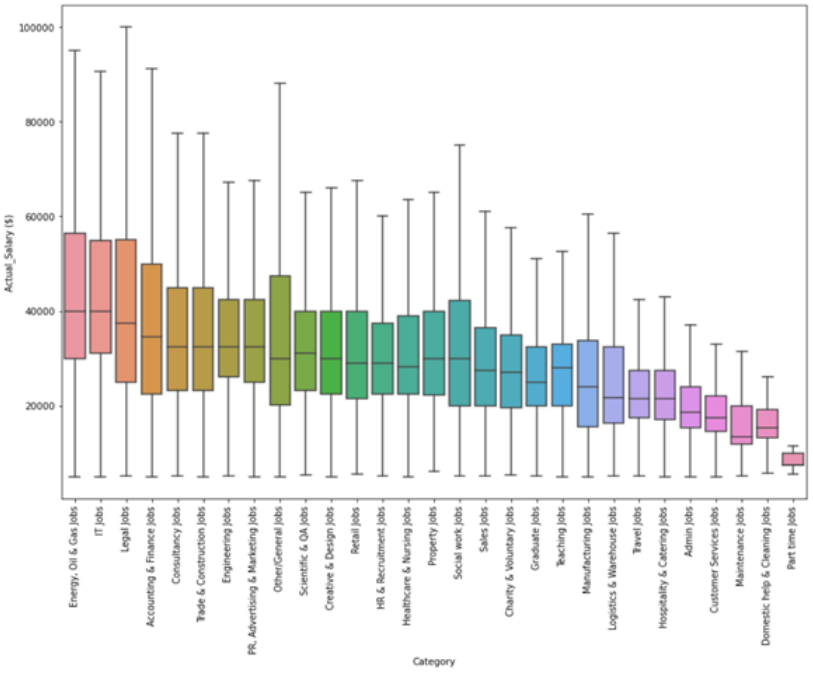


Fig Y. Salary by Job Industry category

A clear difference is observed in the salary of different jobholders' categories. In Fig Y. we see, unsurprisingly, a relatively strong relationship between the industry someone works in and the salary they make. Job Industry category and salary with the top 5 categories being Oil & Gas, IT jobs, Legal jobs, finance, and consulting jobs. Again unsurprisingly, those working part time made significantly less.

Salary versus Contract Agreement

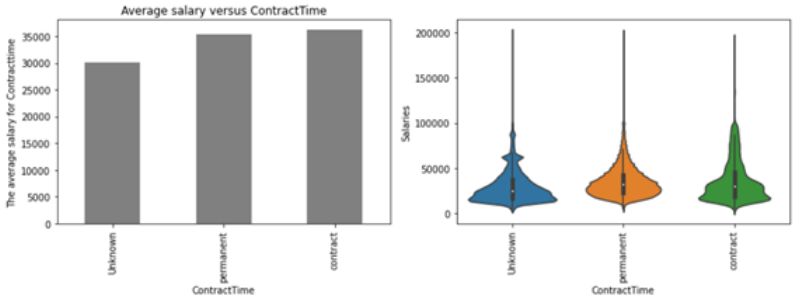


Fig Z. Salary by Job Contract

Permanent contract agreement seems to have higher salary

Created - One-hot encoding for nominal variables

```
# One-hot encoding for nominal variables
one_hot_var = ['Category', 'ContractTime','Company']
train_data_dum = pd.get_dummies(train_data[one_hot_var], drop_first=True)
train_data_dum.head()
```

	Category_Domestic help & Cleaning Jobs	Category_Maintenance Jobs	Category_Customer Services Jobs	Category_Admin Jobs	Category_Hospitality & Catering Jobs	Category_Travel Jobs	Category_Logistics & Warehouse Jobs	Category_Ma
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	

5 rows × 20841 columns

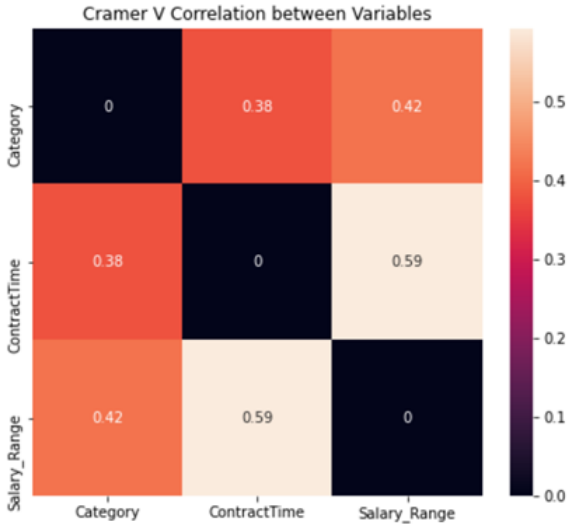
Split the X and y into 75/25 training and testing data subsets

```
#train test split

X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.25, random_state=42)

```

Feature Engineering:

Performed one-hot encoding for nominal categorical variable like category, contract-time and company.

Checked for correlation:



Most of the features are positively correlated with the target variable salary and there is no evidence of multicollinearity of correlation > 0.6.

Regression Models:

Used three models to train the model,

- **Linear Regression**
- **Random Forest**
- **XGBoost**

Trained the model with manual parameter and same model with k-fold(k=5) cross validation

	Linear Regression	Random Forest	XGBoost
Without Cross_validation	1.435527e+04	1.585477e+04	1.401334e+04
R^2 Values	3.300000e-01	0.000000e+00	0.000000e+00
With Cross_validation	1.934695e+24	2.503950e+08	1.996497e+08

From the above table, it shows XGBoost seems have lower root mean squared error without cross validation compared to the other two models(Linear regression and Random Forest), so performed hyperparameter tunning for XGBoost.

Hyperparameter Tunning:

The mean squared error of hyperparameter tunned model is

	XGBoost_ Hyperparameter
RMSE:	14667.57

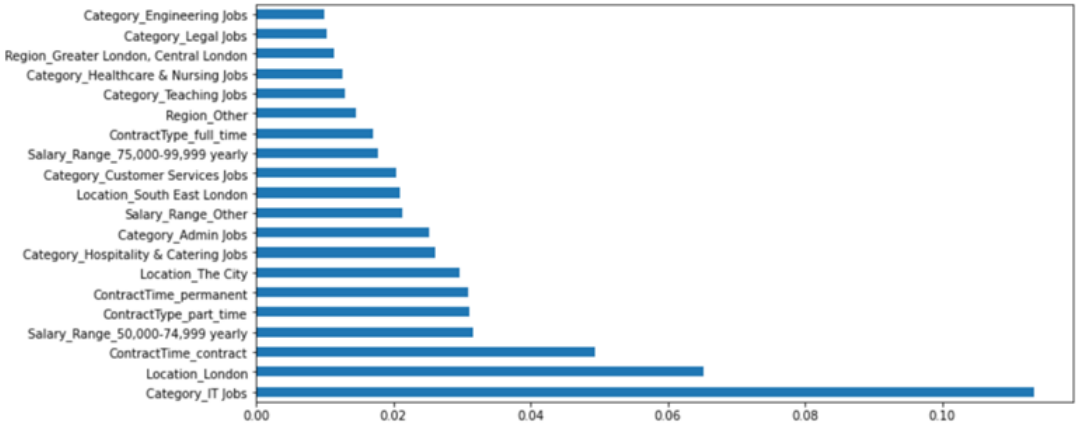
Predicted Result:

The salary is predicted after learning the data from the trained dataset.

	Salary Prediction
0	33645.218750
1	33645.218750
2	33825.910156
3	35094.058594
4	35094.058594

Feature importance:

Below is the feature importance for feature selection for further tunning the model.



Conclusion

Since job description and position strongly influence salary, a company's recruit team might want to consider these factors when negotiating salary for new hires. On the other hand, if a company has fixed budget on new hires, they can decide what kind of position to hire for based on how much they have. For instance, if a company has 100k budget, then it can afford to hire a financial executive management position, instead of a construction manager.

Salary varies according to the following:

• Job type Industry: Oil/Energy> IT jobs > Managers > Healthcare > Part-time
• Salary increases linearly with Job Type
• Salary decreases linearly with temporary and part-time jobs
• Oil and finance industries are the highest paying, I.T, healthcare, and social workers are average paying, while part-time work is the lowest paying.

Based on this information, candidates can also decide the type of industry, location, and position to target to achieve the desired salary. For example, if an entry-level candidate is looking to earn a lot of money, he or she should work in the oil or finance industry. However, living in different locations might be a deciding factor for the best and worst jobs depending on opportunity.

Note:

The type of job opportunities and description titles are important factors to consider when predicting salary, but they are not included in the model. While it is important to consider the position and industry, we should also keep in mind other variables that are not included in the model.

**Please check the notebook for details: https://github.com/kxnarcisse/Springboard/blob/master/Git/Notebooks/Capstone-2/Feature%20Engineering%20Machine%20Learning-%20Salary%20Prediction%20Capstone%202.ipynb