Name : Akhil Eppa
SRN : PES1201802026

Regression Analysis to predict 'Burn Rate'

As a first step of the analysis, the various numerical columns in the training data are understood. Specifically, the range of the values in each column, mean, standard deviation and the various percentiles are understood. In the next step, the data types of various columns are checked and also the number of rows in the data is noted.
Same steps are followed for the test data as well.

Data Preprocessing

As the first step, the column 'Employee ID' is dropped from the test set as there is no column 'Employee ID' in the training data. Besides this, the employee ID attribute cannot be used for making predictions as it is just an alpha-numeric string used to uniquely identify each employee.

In the next step, the number of missing values in the training data are checked. Initially the number of missing values in each column are as follows:

```
Date of Joining            0
Gender                     0
Job Division               0
WFH Setup Available        0
Designation Grade          0
Resource Allocation     1381
Mental Fatigue Score    2117
Burn Rate               1124
dtype: int64
```

The rows with Burn Rate missing have to be dropped completely as these are the values that will be used for training and hence rows without burn rate values are meaningless. Thus, these rows are dropped. After dropping the rows, the missing values in other columns are as follows:

```
Date of Joining            0
Gender                     0
```

```
Job Division               0
WFH Setup Available        0
Designation Grade          0
Resource Allocation     1278
Mental Fatigue Score    1945
Burn Rate                  0
dtype: int64
```

These rows which have missing values are filled with the medians of the corresponding columns. This step ensures there are no more missing values in the training dataset.

The testing data when checked, has no missing values except the 'Burn Rate' column for which values have to be predicted. So, there is no necessity to fill up any missing values in the testing dataset.

Now, moving onto the categorical columns "Gender", "Job Division" and "WFH Setup Available". These are categorical values which cannot be used by a regression model. One hot encoding method is used to convert them to numerical representations. The column "Gender" is created with value 1 representing male and 0 representing female. The "WFH" column is created with 1 if work from home is available and 0 otherwise. A third column "Job Type" is created with value 1 representing service and 0 representing product. The original columns containing categorical data are dropped. These operations are performed both on the training and testing sets.

The last column that has to be pre-processed is the "Date of Joining" column. The "Date of Joining" column will not hold any meaning by itself unless compared with the current date. That is, by comparing the date of joining with the current day, we can obtain the duration for which an employee has been working at a company. This numerical value may be used while making predictions. So, a new column called "Duration" is created in both the datasets which contains the number of days that an employee has been working at a company. The "Date of Joining" column is no longer required and is hence dropped. These operations are performed on both the training and testing datasets. This concludes the pre-processing phase of the analysis.

Exploratory Data Analysis

As a first step, the correlation matrix for the training data is analysed.

	Duration	Gender	Job Type	WFH	Designation Grade	Resource Allocation	Mental Fatigue Score	Burn Rate
Duration	1.000000	0.001111	0.002081	0.005112	-0.009075	-0.007142	-0.001049	-0.001655
Gender	0.001111	1.000000	-0.011660	-0.073740	0.111794	0.136407	0.139104	0.154895
Job Type	0.002081	-0.011660	1.000000	0.003410	0.006597	0.007235	0.002631	0.004281
WFH	0.005112	-0.073740	0.003410	1.000000	-0.230274	-0.276793	-0.263712	-0.306266
Designation Grade	-0.009075	0.111794	0.006597	-0.230274	1.000000	0.851383	0.657882	0.737556
Resource Allocation	-0.007142	0.136407	0.007235	-0.276793	0.851383	1.000000	0.740061	0.829632
Mental Fatigue Score	-0.001049	0.139104	0.002631	-0.263712	0.657882	0.740061	1.000000	0.898926
Burn Rate	-0.001655	0.154895	0.004281	-0.306266	0.737556	0.829632	0.898926	1.000000

Inferences drawn from the correlation matrix:

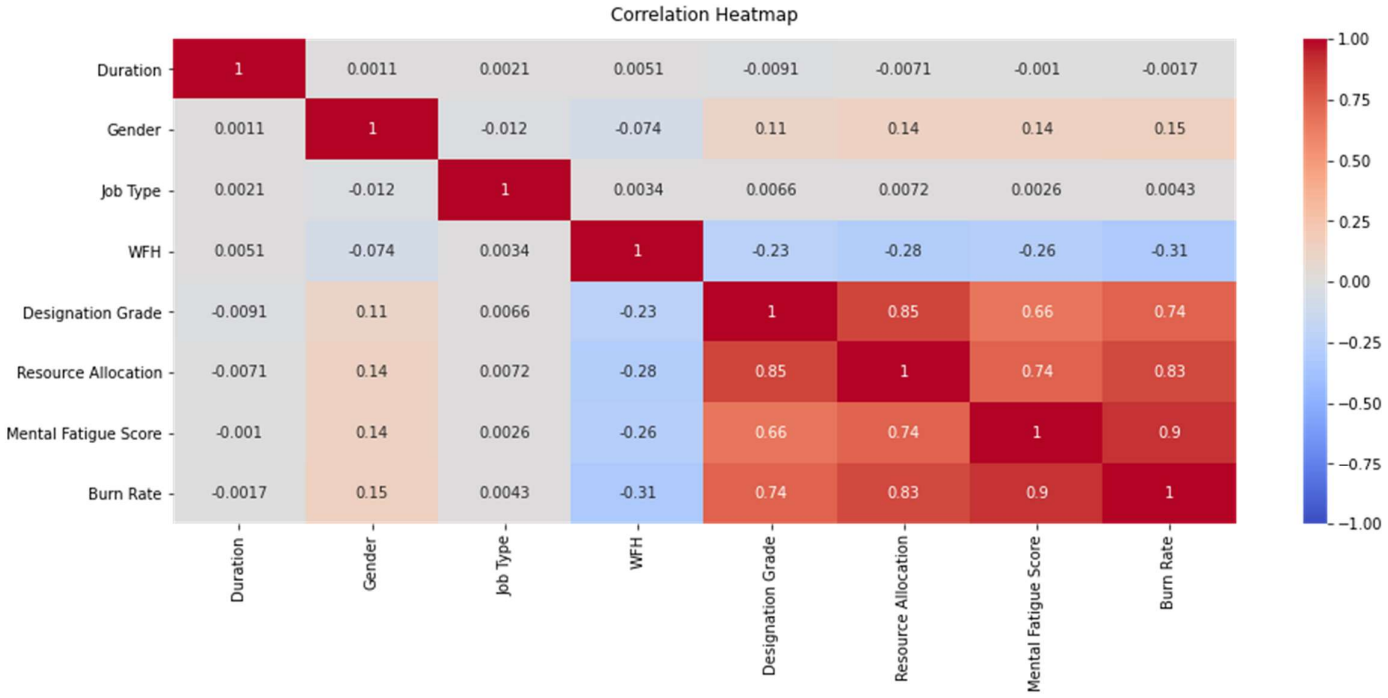
- Burn Rate and Mental Fatigue Score have a high correlation of 0.898926
- Burn Rate and Resource Allocation have a high correlation of 0.829632
- Burn Rate and Destination Grade have a correlation of 0.737556

Hence Mental Fatigue, Resource Allocation and Designation Grade have high influence on Burn Rate in the same mentioned order.

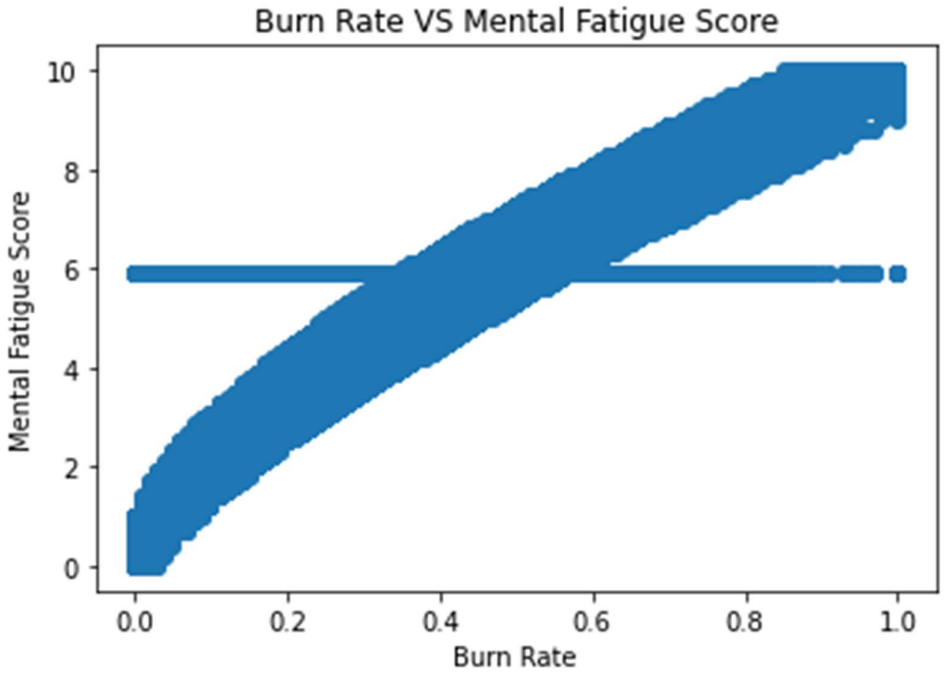
Besides this, the other variables that have high correlations among themselves are:

- Resource Allocation and Designation Grade have correlation of 0.851383
- Mental Fatigue Score and Resource Allocation have correlation of 0.740061

Correlation matrix represented as a heat map for easier understanding:



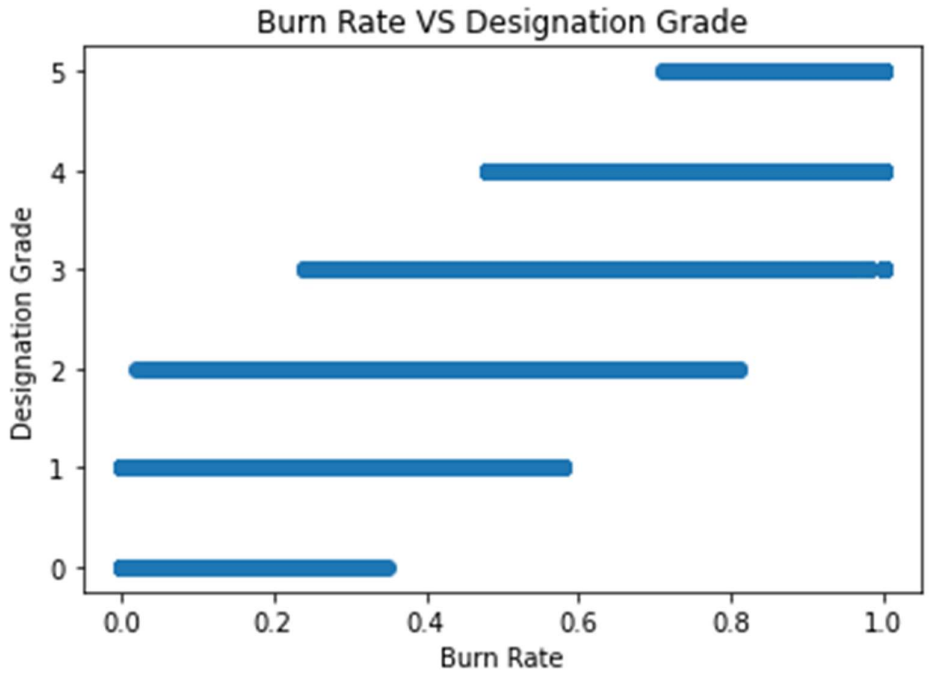
To see how strong the correlation between Mental Fatigue and Burn Rate is, a scatter plot is obtained.



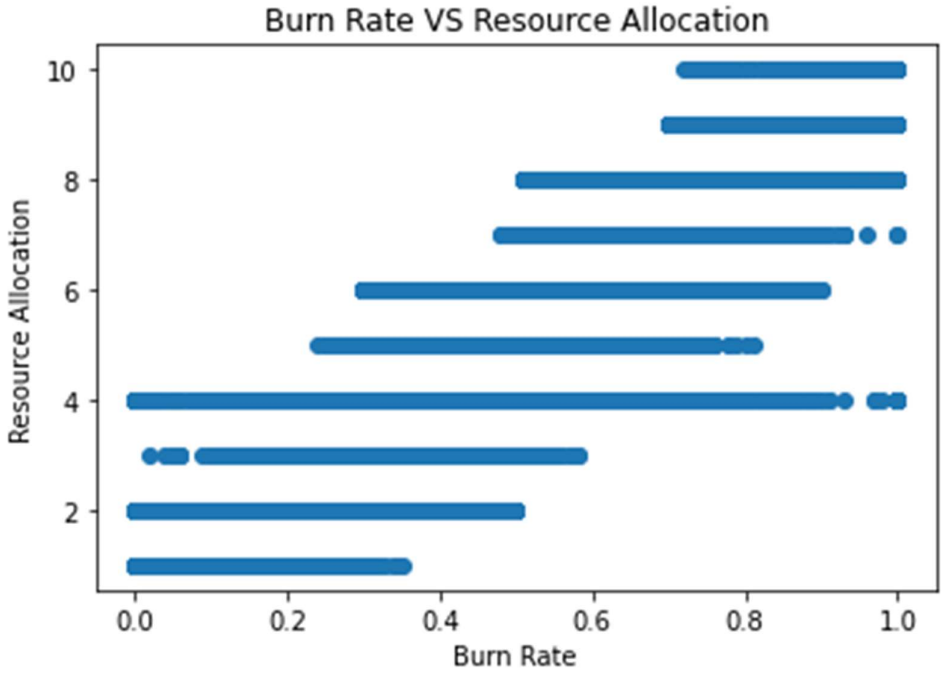
As seen in the scatter plot, most of the points are surrounded around an imaginary straight line showing strong linear relation between the two variables.

The horizontal line that runs in the middle of the plot is formed because the missing values for mental fatigue score are filled with the same value which is the median of all values in mental fatigue score. This shows that dropping rows with mental fatigue scores may be better but that would reduce the amount of training data available. Another option is to build a secondary model that predicts missing mental fatigue scores from other variables like Designation Grade and Resource Allocation. This is another path that can be explored in the future to improve the final model's performance.

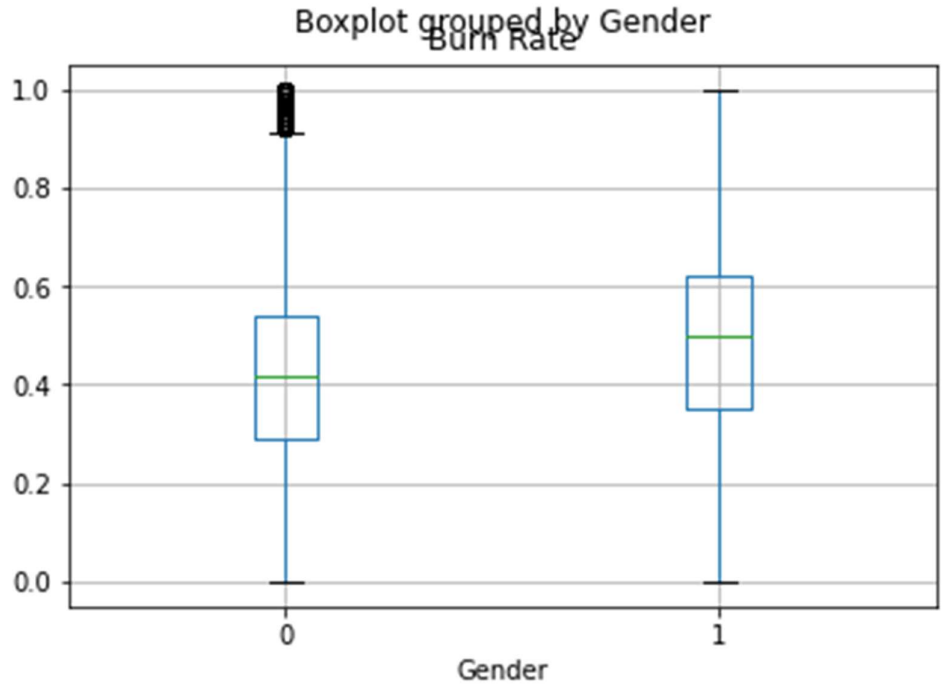
The plot below shows the scatter plot for burn rate along X axis and Designation grade along Y axis. There is a general trend where as the designation grade rises, the burn rate increases signifying that a higher designation grade tends to result in higher burn rate.



The plot below shows the scatter plot for burn rate along X axis and Resource Allocation along Y axis. This plot also shows that an increase in resource allocation increases burn rate but the relation is not very straight forward.

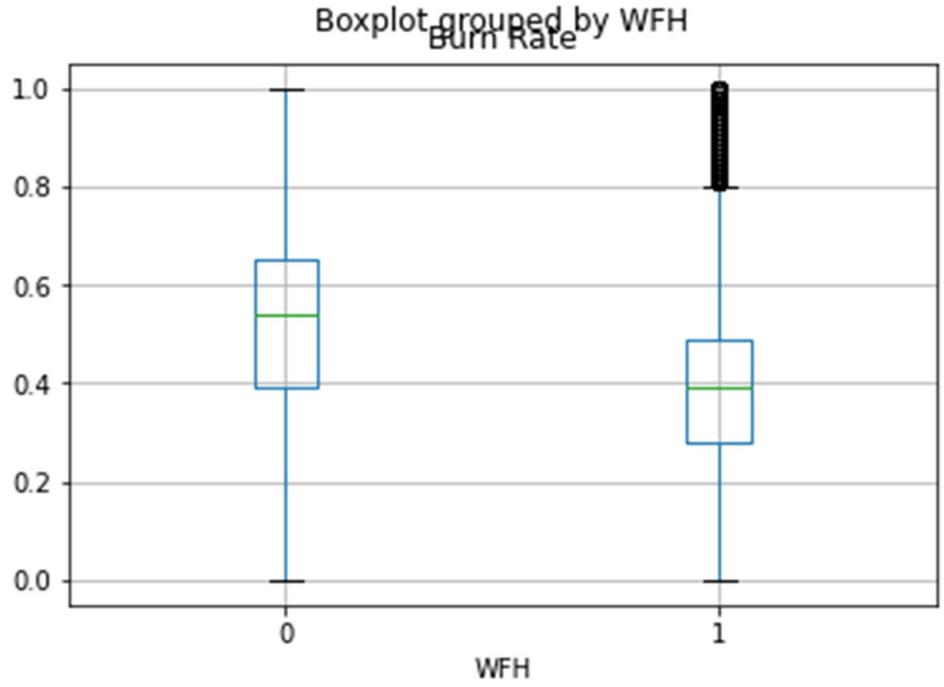


The box plot below shows that the general consensus is that males have a higher mean burn rate compared to females.



Additional information is that the number of males in the pre-processed data is 10277 and number of females is 11349.

The boxplot below shows that the general consensus is that employees who do not have a work from option tend have a higher mean burn rate compared to those who have a work from home option.



Additional information is that in the pre-processed data, the number of employees having the work from home option is 11685 and those not having work from home is 9941.

Model Preparation and Training

A part of the training set is assigned as a validation set. Specifically, 20% of the training set is kept aside as the validation set. The sizes of the training and validation sets are specified below.

```
Training Set: X_train size =  (17300, 7)
Training Set: y_train size =  (17300,)
Validation Set: X_val size =  (4326, 7)
Validation Set: y_val size =  (4326,)
```

X_test is also created which contains all pre-processed columns except burn rate which has to be predicted.

Once this is done, all the values in X_train, X_val and X_test and normalized in the range of 0 to 1 so that the model which is built does not give a higher preference to columns with relatively larger magnitudes. Two functions are also written to print R-squared values and Error values in a neat and understandable format.

- **Linear Regression Model**
  The linear regression model is taken as the baseline model. The model is tested on the validation set. The results obtained are:

```
----------------------------
R2 for training =  0.8710987261690938
R2 for validation =  0.8681063562565299

-------------------------
Errors on Validation Set:
Mean Absolute Error =  0.05376375218853183
Mean Squared Error =  0.005115328990475156
Root Mean Squared Error =  0.07152152816093317
```

- **Decision Tree Regressor**
  The decision tree regressor model is created with the maximum depth parameter as 5. The model is tested on the validation set. The results obtained are:

```
----------------------------
R2 for training =  0.880013949901232
R2 for validation =  0.8775682371857245

-------------------------
Errors on Validation Set:
Mean Absolute Error =  0.05381126726315801
Mean Squared Error =  0.004748361845980525
Root Mean Squared Error =  0.06890835831726456
```

- **Decision Tree Regressor with Parameter Tuning**
  Since there is scope to increase the performance of the decision tree regressor model to an R-Squared score of 0.9 or above, hyperparameter tuning is done. The Grid Search Technique is used. The optimal maximum depth is found to be 8 in a range of 5 to 10. The tuned model is tested on the validation data. The R-Squared value crosses the 0.9 mark. The results obtained are as below:

```
----------------------------
R2 for training =  0.9056479480159318
R2 for validation =  0.9000773441022165

-------------------------
Errors on Validation Set:
Mean Absolute Error =  0.04867275587937491
Mean Squared Error =  0.003875374460905444
Root Mean Squared Error =  0.06225250565965553
```

- **Random Forest Regressor**
  The random forest regressor model is built with the parameters set as 200 for n_estimators and 10 for min_sample_leaf. The model is tested on the validation set. The results obtained are as follows:

```
----------------------------
R2 for training =  0.9251691637615632
R2 for validation =  0.9030733435078913

-------------------------
Errors on Validation Set:
Mean Absolute Error =  0.048047587358581945
Mean Squared Error =  0.0037591783942845067
Root Mean Squared Error =  0.06131213904509047
```

- **Regression using XGBoost**
  The XGBoost regressor model is built with parameters as 350 for n_estimators, 5 for max_depth and 0.1 for the learning rate. The model is tested on the validation set. The results obtained are:

```
----------------------------
R2 for training =  0.9252296927470371
R2 for validation =  0.9046412356654384

-------------------------
Errors on Validation Set:
Mean Absolute Error =  0.04794107120840625
Mean Squared Error =  0.0036983696700745748
Root Mean Squared Error =  0.06081422259697623
```

- **AdaBoost Model**
  The adaboost model is built with the decision tree regressor as a base model. The base model has maximum depth of 8. The adaboost regressor has values 350 for n_estimators and learning_rate as 0.1. The model is tested on the validation set. The results obtained are as follows:

```
----------------------------
R2 for training =  0.9145022075476883
```

```
R2 for validation =  0.900177168387153

-------------------------
Errors on Validation Set:
Mean Absolute Error =  0.0496121145897408
Mean Squared Error =  0.0038715029016384746
Root Mean Squared Error =  0.06222140227958925
```

Conclusion

Model	R-Squared on Training Data	R-Squared on Validation Data
Linear Regression Model	0.8710987261690938	0.8681063562565299
Decision Tree Regressor	0.880013949901232	0.8775682371857245
Decision Tree Regressor with Parameter Tuning	0.9056479480159318	0.9000773441022165
Random Forest Regressor	0.9251691637615632	0.9030733435078913
Regression using XGBoost	0.9252296927470371	**0.9046412356654384**
AdaBoost Model	0.9145022075476883	0.900177168387153

The XGBoost Regressor Model marginally outperforms the Random Forest Regressor Model on the Validation Dataset and is the model with the highest R-Squared value on the Validation Dataset. Hence the XGBoost Regressor is used to predict the burn rate on the test set. The Burn Rate for all rows in the test set is predicted and return to a file called test_results.csv along with all the other original columns.