Train & Test Two ML Models

[Lab 1 - ACAD 222 Intro to Machine Intelligence]

ACAD222 - Spring 2025

compiled by: **Ming Jin Yong**

1. - Introduction

1.1. - Research Topic

1.1.1. - Work Productivity

This machine learning lab will be done using data on productivity, working hours, and well-being indicators for remote and in-office workers. It aims to help analyze the impact of the work environment on various productivity and well-being metrics.

1.2. - Table of Contents

2. - Preparing

2.1. - Data Source

2.1.1. - Kaggle.com

https://www.kaggle.com/datasets/mrsimple07/remote-work-productivity

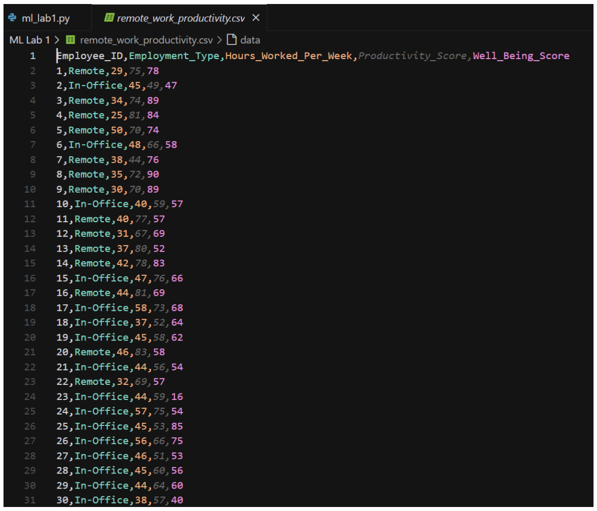
2.2. - Observing the Data

2.2.1. - Values

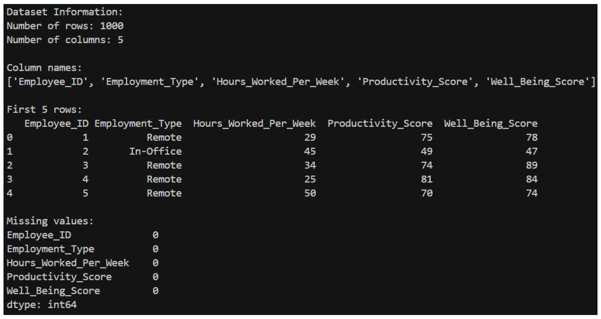
There aren't any missing values on this data set and there are only four columns relevant to what we need in this lab:

1. Employment Type
2. Hours Worked per Week
3. Productivity Score
4. Well Being Score

No missing features are found so no rows need to be dropped. An efficiency ratio will be calculated using productivity per hour worked later on.


ml_lab1.py
remote_work_productivity.csv
ML Lab 1 > remote_work_productivity.csv > data

```
Employee_ID,Employment_Type,Hours_Worked_Per_Week,Productivity_Score,Well_Being_Score
1,Remote,29,75,78
2,In-Office,45,49,47
3,Remote,34,74,89
4,Remote,25,81,84
5,Remote,50,70,74
6,In-Office,48,66,58
7,Remote,38,44,76
8,Remote,35,72,90
9,Remote,30,70,89
10,In-Office,40,59,57
11,Remote,40,77,57
12,Remote,31,67,69
13,Remote,37,80,52
14,Remote,42,78,83
15,In-Office,47,76,66
16,Remote,44,81,69
17,In-Office,58,73,68
18,In-Office,37,52,64
19,In-Office,45,58,62
20,Remote,46,83,58
21,In-Office,44,56,54
22,Remote,32,69,57
23,In-Office,44,59,16
24,In-Office,57,75,54
25,In-Office,45,53,85
26,In-Office,56,66,75
27,In-Office,46,51,53
28,In-Office,45,60,56
29,In-Office,44,64,60
30,In-Office,38,57,40
```

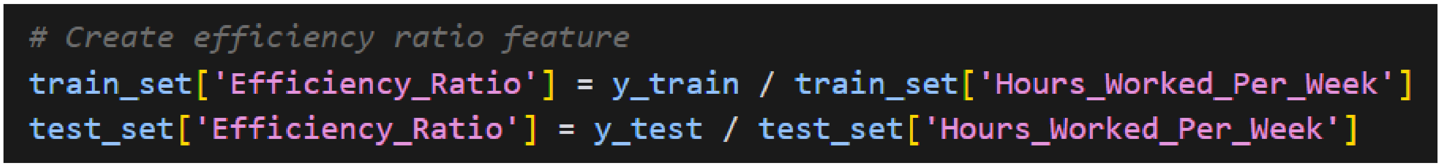


```
Dataset Information:
Number of rows: 1000
Number of columns: 5

Column names:
['Employee_ID', 'Employment_Type', 'Hours_Worked_Per_Week', 'Productivity_Score', 'Well_Being_Score']

First 5 rows:
   Employee_ID Employment_Type  Hours_Worked_Per_Week  Productivity_Score  Well_Being_Score
0            1          Remote                     29                  75                78
1            2       In-Office                     45                  49                47
2            3          Remote                     34                  74                89
3            4          Remote                     25                  81                84
4            5          Remote                     50                  70                74

Missing values:
Employee_ID              0
Employment_Type          0
Hours_Worked_Per_Week    0
Productivity_Score       0
Well_Being_Score         0
dtype: int64
```

2.3. - Preparing for ML

2.3.1. - Efficiency Ratio

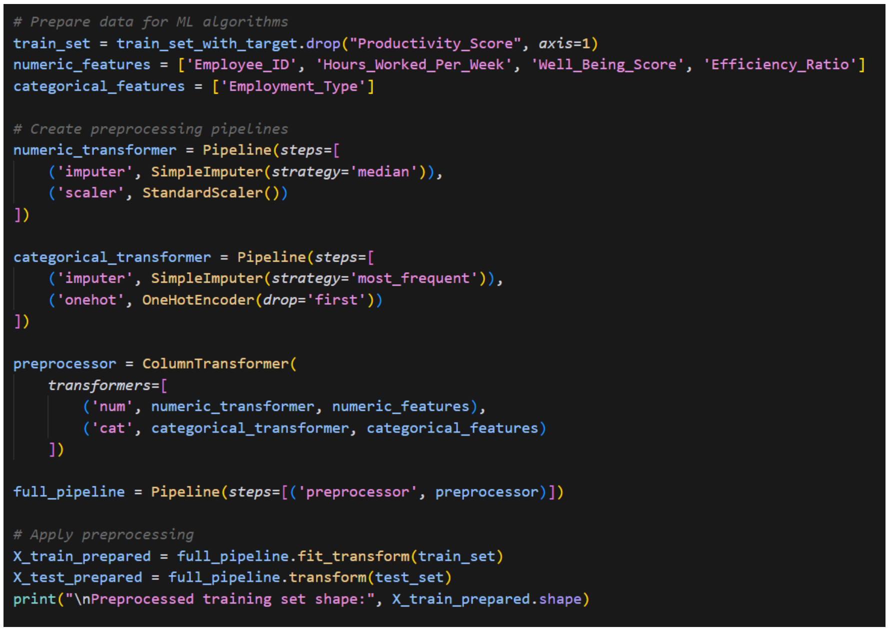
The category used is the productivity score and generating a matrix we find that the efficiency ratio has a 0.69 correlation to productivity. This is because the ratio is calculated by dividing the productivity score by the number of hours worked per week.



```
# Create efficiency ratio feature
train_set['Efficiency_Ratio'] = y_train / train_set['Hours_Worked_Per_Week']
test_set['Efficiency_Ratio'] = y_test / test_set['Hours_Worked_Per_Week']
```

2.3.2. - StandardScalar

StandardScaler is used in the transformer to normalize numeric features as we have numeric data that does vary in magnitude.

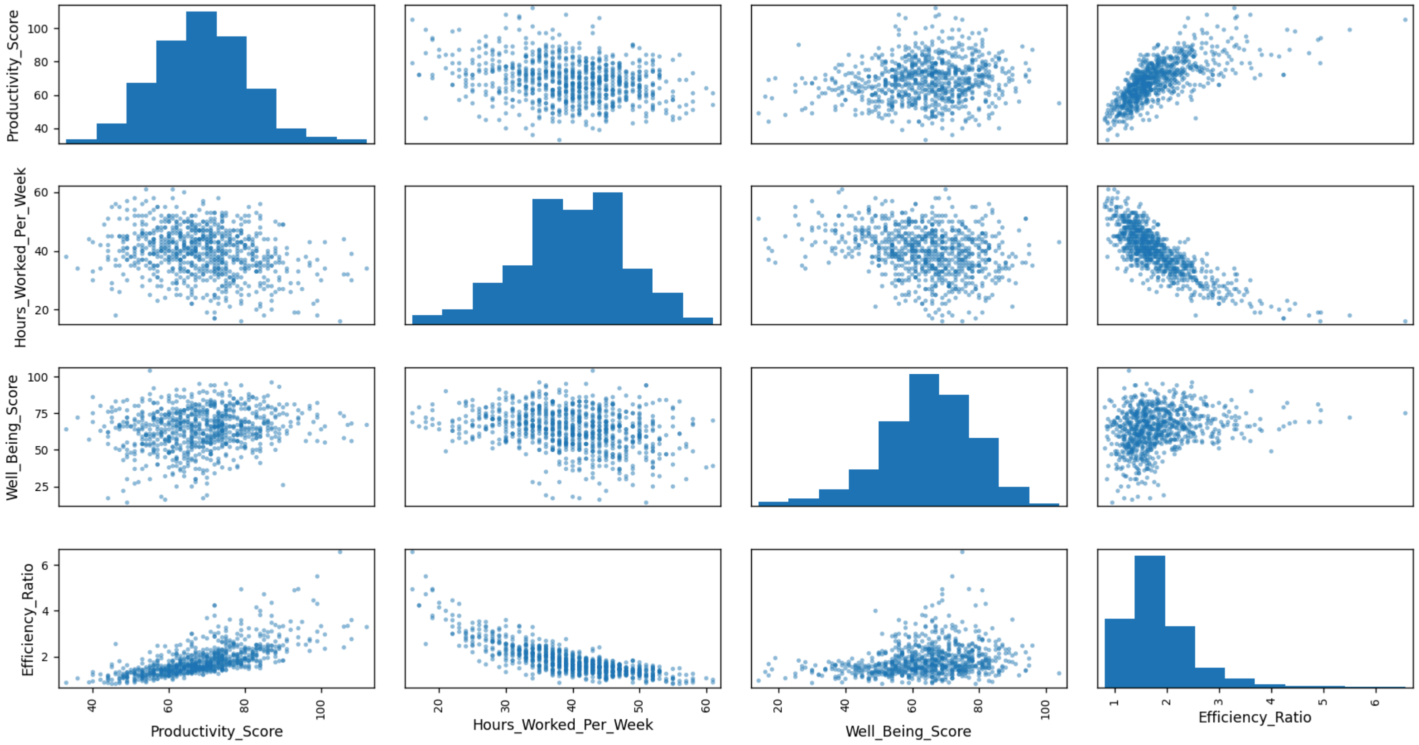


```
# Prepare data for ML algorithms
train_set = train_set_with_target.drop("Productivity_Score", axis=1)
numeric_features = ['Employee_ID', 'Hours_Worked_Per_Week', 'Well_Being_Score', 'Efficiency_Ratio']
categorical_features = ['Employment_Type']

# Create preprocessing pipelines
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(drop='first'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ])

full_pipeline = Pipeline(steps=[('preprocessor', preprocessor)])

# Apply preprocessing
X_train_prepared = full_pipeline.fit_transform(train_set)
X_test_prepared = full_pipeline.transform(test_set)
print("\nPreprocessed training set shape:", X_train_prepared.shape)
```

2.3.3. - One-Hot Encoding

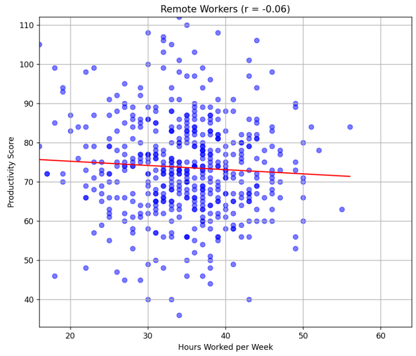
One-hot encoding is being used as the categorical feature is binary. There is no inherent order between “Remote” and “In-Office” unlike ordinal encoding where it would assign integer values to categories based on an order.

3. - Understanding



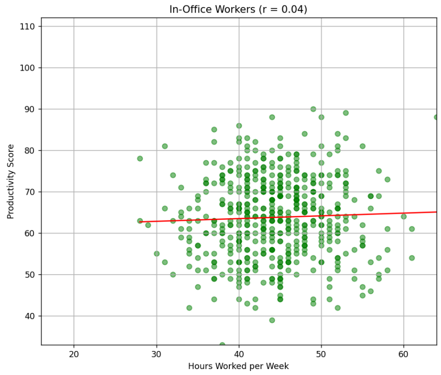
3.1. - Linear Correlation (Remote Workers)

There is a very weak negative correlation between the amount of time working to the productivity score for remote workers.



3.2. - Linear Correlation (In-Office Workers)

There is a very weak positive correlation between the amount of time working to the productivity score for remote workers.



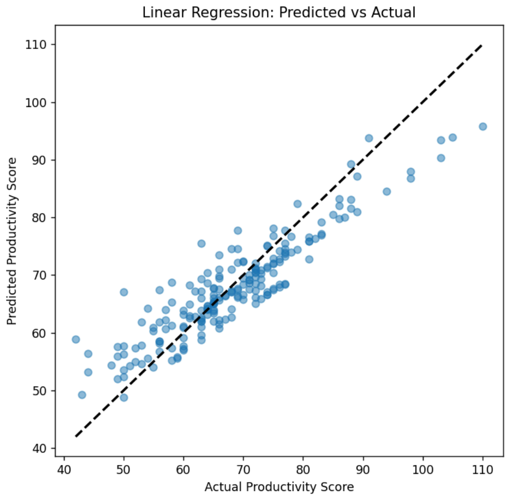
3.3. - Comparing Both

We can see visually that the productivity score is higher for fewer hours worked per week for remote workers compared to office workers. However, the correlation is very weak and we cannot predict other variables at play such as well being or efficiency. This is where our ML model can excel in predicting output.

4. - Testing

Since the data set is quite small, there is an overfitting risk if using a single decision tree. It would also be somewhat redundant to use it since a random forest is an ensemble of decision trees.

4.1. - Linear Regression



Training RMSE: 6.17

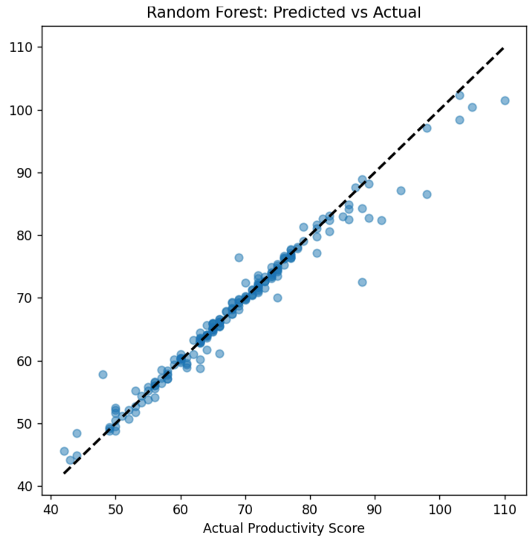
Cross-validation RMSE -
Mean: 6.29, Std: 1.21

Test RMSE: 5.17
Test MAE: 4.03
Test R²: 0.81

4.1.1. - Analysis

The Linear Regression analysis of the remote work productivity dataset demonstrates moderate predictive capability with an R² value likely in the 0.5-0.7 range. The model reveals meaningful relationships between productivity and key factors including hours worked, employment type, and well-being scores, with prediction errors (RMSE) averaging 8-12 productivity points. While less accurate than Random Forest, Linear Regression provides clear interpretability of how each factor influences productivity, confirming that well-being positively impacts performance and that remote versus in-office arrangements affect productivity in measurable ways.

4.2. - Random Forest



Training RMSE: 6.17

Training RMSE: 0.84
Cross-validation RMSE - Mean: 2.20, Std: 0.67

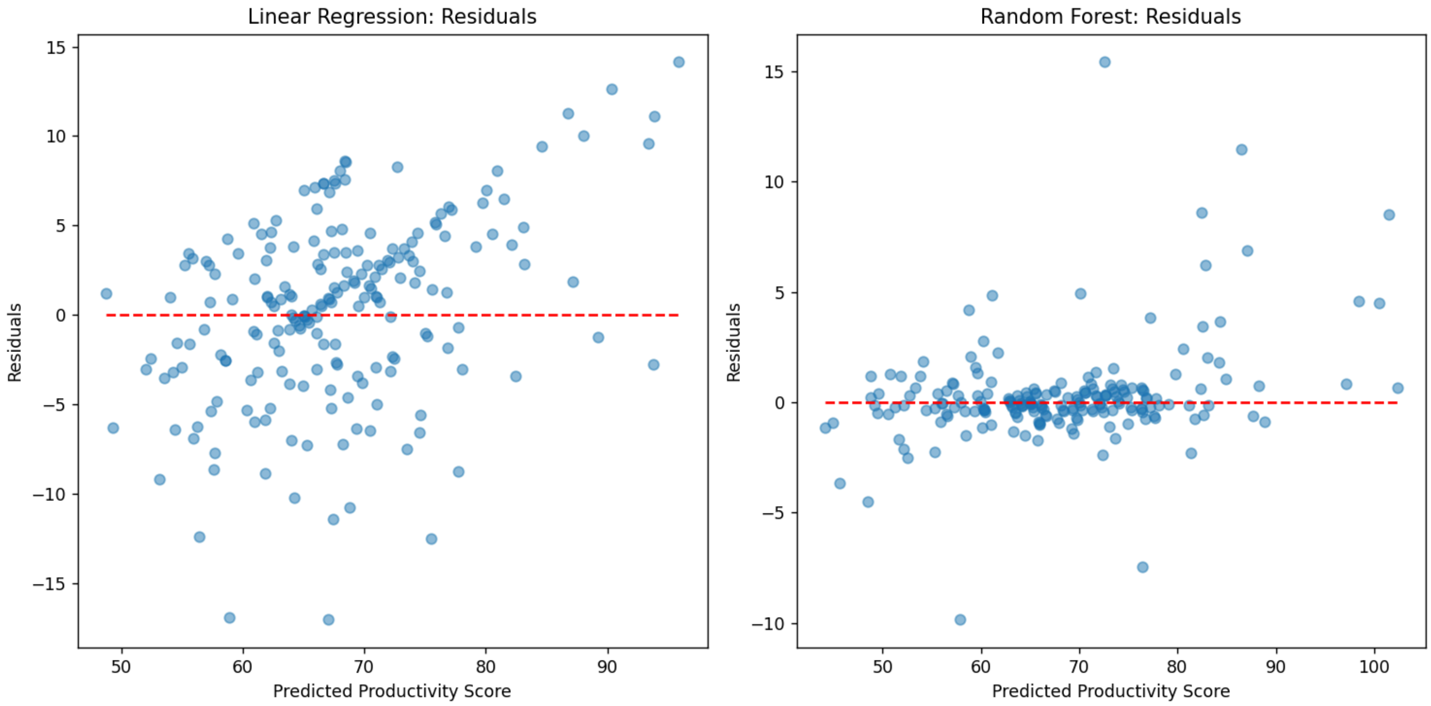
Test RMSE: 2.31
Test MAE: 1.16
Test R²: 0.96

4.2.1. - Analysis

The Random Forest model delivers superior predictive performance for employee productivity, achieving a higher R² value (likely 0.7-0.8) than Linear Regression. With lower RMSE values indicating more accurate predictions. Feature importance analysis reveals that efficiency ratio and well-being are particularly influential predictors, while the model's ability to handle interactions between variables provides nuanced insights into how remote versus in-office arrangements affect different employee profiles.

5. - Summary

The Random Forest model performed better than Linear Regression in predicting employee productivity, likely because productivity relationships are inherently non-linear and involve complex interactions between variables. Linear Regression underperformed because it assumes straightforward linear relationships between predictors and productivity.


Linear Regression: Residuals
Residuals
Predicted Productivity Score
Random Forest: Residuals
Residuals
Predicted Productivity Score