

```
# Import necessary libraries
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
```

```
# Load the dataset
data = pd.read_csv('train.csv')
```

```
numeric_cols = ['Designation', 'Resource Allocation', 'Mental Fatigue Score']
```

```
for col in numeric_cols:
    data[col] = pd.to_numeric(data[col], errors='coerce')
data = data.apply(lambda x: x.fillna(x.mean()) if x.name in numeric_cols else x)
data.dropna(inplace=True)
```

```
data.isnull()
```



	Employee ID	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...
22745	False	False	False	False	False	False	False	False	False
22746	False	False	False	False	False	False	False	False	False
22747	False	False	False	False	False	False	False	False	False
22748	False	False	False	False	False	False	False	False	False
22749	False	False	False	False	False	False	False	False	False

21626 rows × 9 columns

```
data['Gender'] = data['Gender'].map({'Male': 0, 'Female': 1})
data['Company Type'] = data['Company Type'].map({'Service': 0, 'Product': 1})
data['WFH Setup Available'] = data['WFH Setup Available'].map({'No': 0, 'Yes': 1})
```

```
X = data.drop(['Employee ID', 'Date of Joining', 'Burn Rate'], axis=1)
y = data['Burn Rate']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='linear'))
```

```
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
model.fit(X_train, y_train, epochs=500, batch_size=32, validation_data=(X_test, y_test))
```



```
-----
NameError                                Traceback (most recent call last)
<ipython-input-1-9f848cf5838b> in <cell line: 1>()
----> 1 model.fit(X_train, y_train, epochs=500, batch_size=32, validation_data=(X_test,
y_test))

NameError: name 'model' is not defined
```

```
loss = model.evaluate(X_test, y_test)
print(f'Mean Squared Error on Test Data: {loss}')
```



```
136/136 [=====] - 0s 1ms/step - loss: 0.0044
Mean Squared Error on Test Data: 0.0043907794170081615
```

```
predictions = model.predict(X_test)
```



```
136/136 [=====] - 0s 826us/step
```

