Foundations Of Data Science

NAME: ROHIT MALVIYA

ROLL NO: 10315

CLASS: MSc DA

About the Dataset:

Dataset link: https://archive.ics.uci.edu/dataset/849/power+consumption+of+tetouan+city

This dataset is related to power consumption of three different distribution networks of Tetouan city which is located in north Morocco.

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate, Time-Series	Social Science	Regression
Feature Type	**# Instances**	**# Features**
Real, Integer	52417	6

Variable Name	Role	Type	Description	Units	Missing Values
DateTime	Feature	Date	Each ten minutes	-	no
Temperature	Feature	Continuous	Weather Temperature of Tetouan city	°C (Celsius)	no
Humidity	Feature	Continuous	Weather Humidity of Tetouan city	% (Percent)	no
Wind Speed	Feature	Continuous	Wind speed of Tetouan city	m/s (meters per second)	no
general diffuse flows	Feature	Continuous	General diffuse flows	W/m² (Watts per square meter)	no
diffuse flows	Feature	Continuous	Diffuse flows	W/m² (Watts per square meter)	no
Zone 1 Power Consumption	Target	Continuous	Power consumption of zone 1 of Tetouan city	kW (Kilowatts)	no
Zone 2 Power Consumption	Target	Continuous	Power consumption of zone 2 of Tetouan city	kW (Kilowatts)	no
Zone 3 Power Consumption	Target	Continuous	Power consumption of zone 3 of Tetouan city	kW (Kilowatts)	no

Abstract:

Energy management is crucial for optimizing power distribution in different regions. This project uses machine learning to predict power consumption in three distinct zones based on environmental factors such as temperature, humidity, wind speed, and solar radiation. We employ a Random Forest regression model in R to predict energy consumption, providing a solution that can enhance power grid efficiency and forecasting accuracy.

Introduction:

Power consumption forecasting plays a pivotal role in ensuring that energy supply meets demand. Accurately predicting energy consumption patterns allows utility companies to optimize energy distribution and manage grid stability, preventing blackouts and overproduction. In this project, we aim to predict power consumption in three zones using environmental data, leveraging the Random Forest model for its robustness in handling large datasets and complex interactions between variables.

The dataset provided contains hourly measurements of various environmental parameters—temperature, humidity, wind speed, and solar radiation—and power consumption for three zones. We will use this data to build an end-to-end machine learning model to predict power consumption based on these factors.

Methodology:

Data Preprocessing:

- Handle missing values and convert the DateTime field to a proper time format for analysis.
- Normalize the environmental data to ensure that variables are on the same scale.

Feature Selection:

- The following features will be used:
  - Temperature
  - Humidity
  - Wind Speed
  - General diffuse flows
  - Diffuse flows
  - Additional features will be created as per needs
- Target variables are the power consumption in Zone 1, Zone 2, and Zone 3.

Model Building:

- We will use the Random Forest regression model in R to predict power consumption in each zone.
- Train/test split will be used to evaluate the model performance.

Evaluation Metrics:

- Root Mean Squared Error (RMSE)
- R-squared score

Steps/Code and Output:

Installing the Required Libraries and importing them and loading the dataset:

```
> library(randomForest)
randomForest 4.7-1.2
Type rfNews() to see new features/changes/bug fixes.
> library(caTools)
> # Loading the raw data set
> data <- read.csv(file.choose())
> head(data)
       DateTime Temperature Humidity Wind.Speed
1 1/1/2017 0:00       6.559     73.8      0.083
2 1/1/2017 0:10       6.414     74.5      0.083
3 1/1/2017 0:20       6.313     74.5      0.080
4 1/1/2017 0:30       6.121     75.0      0.083
5 1/1/2017 0:40       5.921     75.7      0.081
6 1/1/2017 0:50       5.853     76.9      0.081
  general.diffuse.flows diffuse.flows Zone.1.Power.Consumption
1                 0.051         0.119                 34055.70
2                 0.070         0.085                 29814.68
3                 0.062         0.100                 29128.10
4                 0.091         0.096                 28228.86
5                 0.048         0.085                 27335.70
6                 0.059         0.108                 26624.81
  Zone.2..Power.Consumption Zone.3..Power.Consumption
1                  16128.88                  20240.96
2                  19375.08                  20131.08
3                  19006.69                  19668.43
4                  18361.09                  18899.28
5                  17872.34                  18442.41
6                  17416.41                  18130.12
```

Preprocess data:

```
> #preprocess data
> # Convert 'DateTime' column to datetime format
> data$DateTime <- as.POSIXct(data$DateTime, format="%m/%d/%Y %H:%M")
> # Check for any missing values
> colSums(is.na(data))
                 DateTime               Temperature
                        0                         0
                 Humidity                Wind.Speed
                        0                         0
    general.diffuse.flows             diffuse.flows
                        0                         0
 Zone.1.Power.Consumption Zone.2..Power.Consumption
                        0                         0
Zone.3..Power.Consumption
                        0
> # Normalize the environmental features (Temperature, Humidity, Wind Speed, etc.)
> # scale the data to a range between 0 and 1.
> normalize <- function(x) {
+   return ((x - min(x)) / (max(x) - min(x)))
+ }
> # Apply normalization to the relevant columns
> data$Temperature <- normalize(data$Temperature)
> data$Humidity <- normalize(data$Humidity)
> data$Wind.Speed <- normalize(data$Wind.Speed)
> data$general.diffuse.flows <- normalize(data$general.diffuse.flows)
> data$diffuse.flows <- normalize(data$diffuse.flows)
```

Data preview after normalization (scaling the data to a range

between 0 and 1):

```
> head(data)
             DateTime Temperature  Humidity  Wind.Speed
1 2017-01-01 00:00:00   0.1553252 0.7077276 0.003699897
2 2017-01-01 00:10:00   0.1485251 0.7184392 0.003699897
3 2017-01-01 00:20:00   0.1437884 0.7184392 0.003083248
4 2017-01-01 00:30:00   0.1347840 0.7260903 0.003699897
5 2017-01-01 00:40:00   0.1254045 0.7368018 0.003288798
6 2017-01-01 00:50:00   0.1222154 0.7551645 0.003288798
  general.diffuse.flows diffuse.flows Zone.1.Power.Consumption
1          4.994364e-05  9.188358e-05                 34055.70
2          7.151022e-05  5.555751e-05                 29814.68
3          6.242955e-05  7.158372e-05                 29128.10
4          9.534696e-05  6.731007e-05                 28228.86
5          4.653839e-05  5.555751e-05                 27335.70
6          5.902431e-05  8.013103e-05                 26624.81
  Zone.2..Power.Consumption Zone.3..Power.Consumption
1                  16128.88                  20240.96
2                  19375.08                  20131.08
3                  19006.69                  19668.43
4                  18361.09                  18899.28
5                  17872.34                  18442.41
6                  17416.41                  18130.12
```

Setting a seed for reproducibility and Splitting the dataset into training and testing sets:

```
> set.seed(123)
> # Split the data into training (70%) and test (30%)
> split <- sample.split(data$Zone.1.Power.Consumption, SplitRatio = 0.7)
> training_set <- subset(data, split == TRUE)
> test_set <- subset(data, split == FALSE)
```

APPLYING RANDOM FOREST NOW

Train Random Forest model for Zone 1 Power Consumption:

```
> # Train Random Forest model for Zone 1 Power Consumption
> rf_model_zone1 <- randomForest(Zone.1.Power.Consumption ~ Temperature + Humidity + Wind.Speed + general
diffuse.flows + diffuse.flows,
+                              data = training_set, ntree = 500)
```

Train Random Forest model for Zone 2 Power Consumption:

```
> rf_model_zone2 <- randomForest(Zone.2..Power.Consumption ~ Temperature + Humidity + Wind.Speed + genera
l.diffuse.flows + diffuse.flows,
+                              data = training_set, ntree = 500)
```

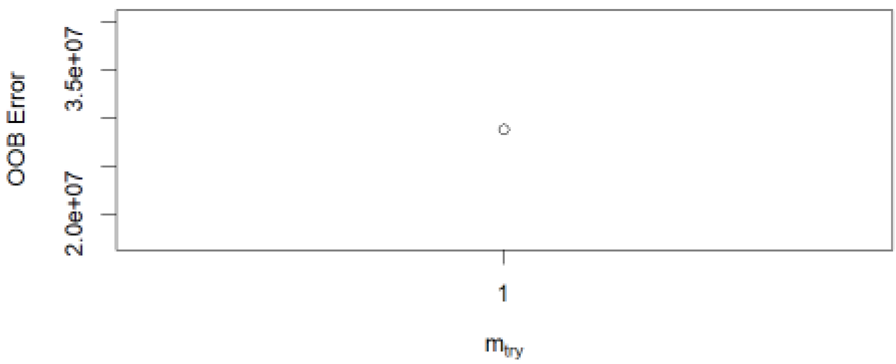
Train Random Forest model for Zone 3 Power Consumption:

```
> rf_model_zone3 <- randomForest(Zone.3..Power.Consumption ~ Temperature + Humidity + Wind.Speed + genera
l.diffuse.flows + diffuse.flows,
+                              data = training_set, ntree = 500)
```

Display model summaries:

```
> print(rf_model_zone1)

Call:
 randomForest(formula = Zone.1.Power.Consumption ~ Temperature +      Humidity + Wind.Speed + general.diffuse.
flows + diffuse.flows,      data = training_set, ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 28684396
                    % Var explained: 43.15
> print(rf_model_zone2)

Call:
 randomForest(formula = Zone.2..Power.Consumption ~ Temperature +      Humidity + Wind.Speed + general.diffus
e.flows + diffuse.flows,      data = training_set, ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 11629626
                    % Var explained: 41.62
> print(rf_model_zone3)

Call:
 randomForest(formula = Zone.3..Power.Consumption ~ Temperature +      Humidity + Wind.Speed + general.diffus
e.flows + diffuse.flows,      data = training_set, ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 11046016
                    % Var explained: 42.84
```

The errors are very high in above model summaries therefore we try the mtry




```
> print(rf_model_zone1)

Call:
 randomForest(formula = Zone.1.Power.Consumption ~ Temperature +      Hum
idity + Wind.Speed + general.diffuse.flows + diffuse.flows,      data = t
raining_set, ntree = 1000, mtry = best_mtry_value)
               Type of random forest: regression
                     Number of trees: 1000
No. of variables tried at each split: 1

          Mean of squared residuals: 28609944
                    % Var explained: 43.3
>
```

Still large error

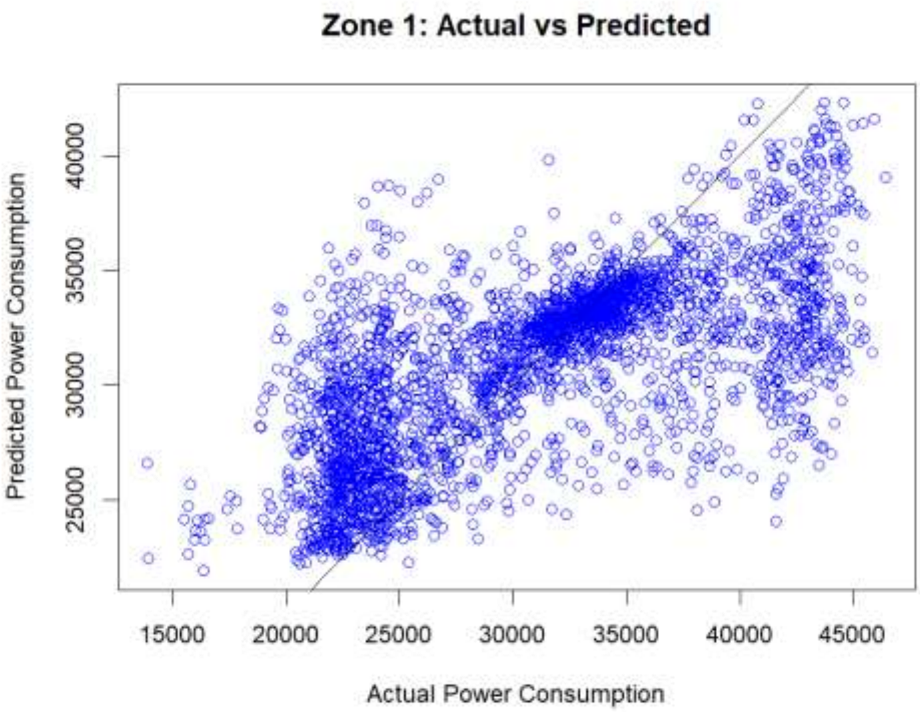
```
#Predict Power Consumption on the Test Set
# Predictions for Zone 1
pred_zone1 <- predict(rf_model_zone1, newdata = test_set)

# Predictions for Zone 2
pred_zone2 <- predict(rf_model_zone2, newdata = test_set)

# Predictions for Zone 3
pred_zone3 <- predict(rf_model_zone3, newdata = test_set)


#Model evaluation
# Function to calculate RMSE
rmse <- function(actual, predicted) {
  sqrt(mean((actual - predicted)^2))
}

# Zone 1 Evaluation
zone1_rmse <- rmse(test_set$Zone.1.Power.Consumption, pred_zone1)
zone1_r2 <- summary(lm(pred_zone1 ~ test_set$Zone.1.Power.Consumption))$r.squared
```

```
# Zone 2 Evaluation
zone2_rmse <- rmse(test_set$Zone.2..Power.Consumption, pred_zone2)
zone2_r2 <- summary(lm(pred_zone2 ~ test_set$Zone.2..Power.Consumption))$r.squared

# Zone 3 Evaluation
zone3_rmse <- rmse(test_set$Zone.3..Power.Consumption, pred_zone3)
zone3_r2 <- summary(lm(pred_zone3 ~ test_set$Zone.3..Power.Consumption))$r.squared

# Print RMSE and R-squared
cat("Zone 1 RMSE:", zone1_rmse, "\nZone 1 R-squared:", zone1_r2, "\n")
cat("Zone 2 RMSE:", zone2_rmse, "\nZone 2 R-squared:", zone2_r2, "\n")
cat("Zone 3 RMSE:", zone3_rmse, "\nZone 3 R-squared:", zone3_r2, "\n")
```

```
> zone3_r2 <- summary(lm(pred_zone3 ~ test_
e.3..Power.Consumption))$r.squared
> # Print RMSE and R-squared
> cat("Zone 1 RMSE:", zone1_rmse, "\nZone 
ed:", zone1_r2, "\n")
Zone 1 RMSE: 5226.411
Zone 1 R-squared: 0.4674857
> cat("Zone 2 RMSE:", zone2_rmse, "\nZone 
ed:", zone2_r2, "\n")
Zone 2 RMSE: 3353.144
Zone 2 R-squared: 0.4419225
> cat("Zone 3 RMSE:", zone3_rmse, "\nZone 
ed:", zone3_r2, "\n")
Zone 3 RMSE: 3244.416
Zone 3 R-squared: 0.4558384
```

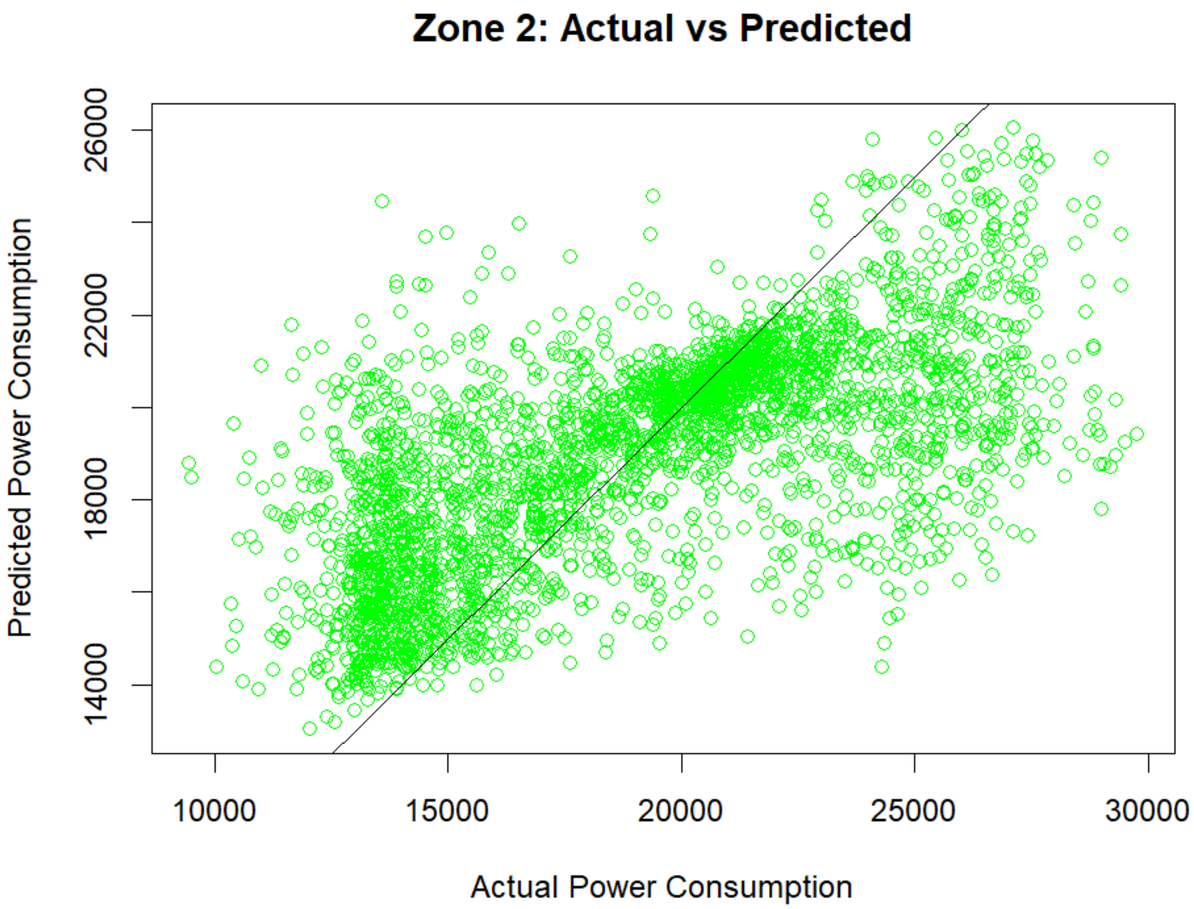
Not terrible results but still it can be improved

```
# Visualization for Zone 1
plot(test_set$Zone.1.Power.Consumption,
     pred_zone1, main = "Zone 1: Actual vs Predicted",
     xlab = "Actual Power Consumption",
     ylab = "Predicted Power Consumption",
     col = "blue")
abline(0, 1)
```

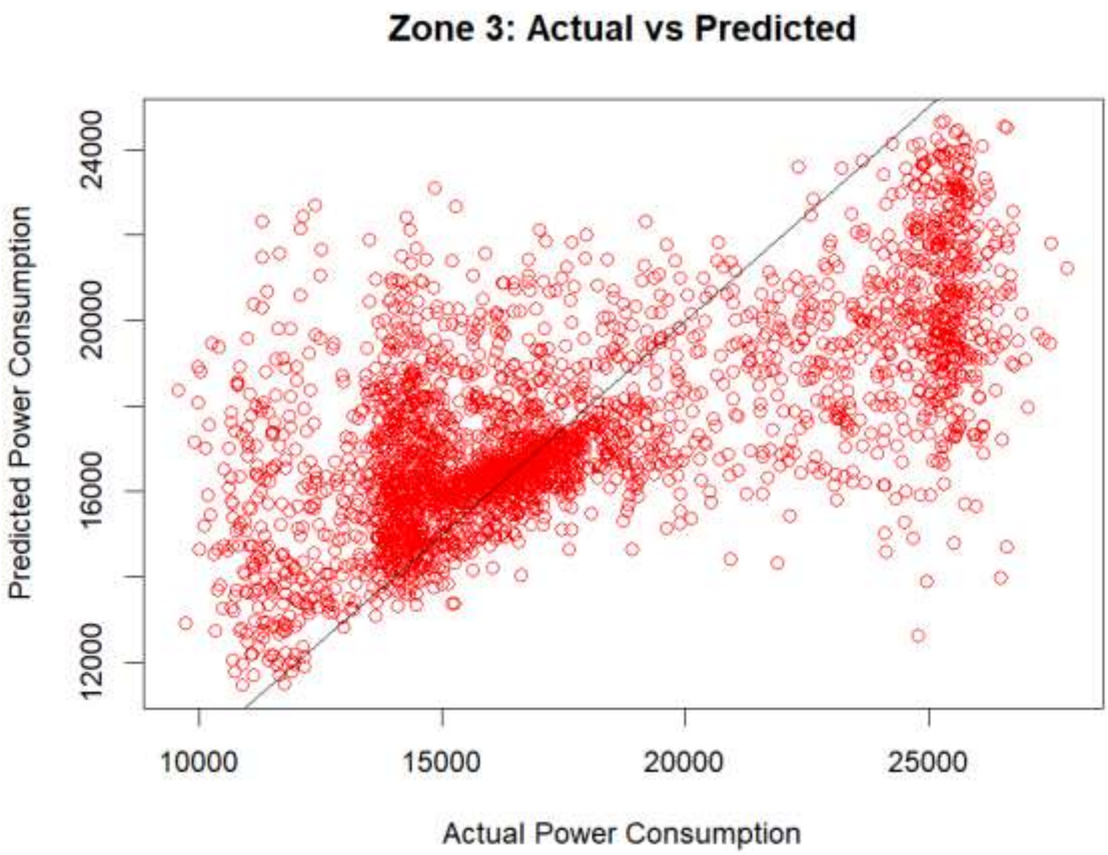


Scattered data points in this visualization (it can be improved)

Visualization for Zone 2



Visualization for Zone 3



Incorporate Hour and Calendar Data for better model performance:

```
# Convert DateTime to hour and day of the week
training_set$Hour <- as.numeric(format(training_set$DateTime, "%H"))
training_set$DayOfWeek <- as.factor(weekdays(as.Date(training_set$DateTime)))
```

Significant improvement after incorporating Hour and Calendar Data:

```
> rf_model_zone11 <- randomForest(Zone.1.Power.Consumption ~ Temperature
+                               + Humidity + Wind.Speed + general.diffuse.flows
+                               + diffuse.flows + Hour + DayOfWeek,
+                               data = training_set,
+                               ntree = 500)
> rf_model_zone11

Call:
 randomForest(formula = Zone.1.Power.Consumption ~ Temperature +      Humidity + Wind.Speed + general.diffuse.
flows + diffuse.flows +      Hour + DayOfWeek, data = training_set, ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 1354708
                    % Var explained: 97.32
```

Now training for zone 2 and 3:

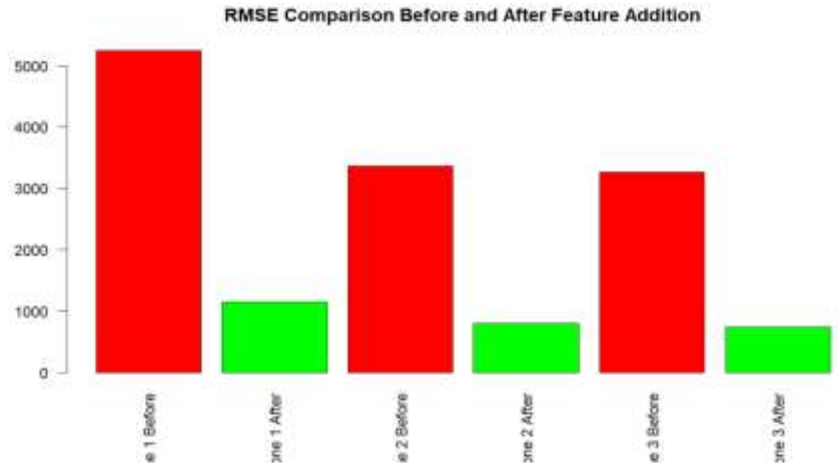
```
> rf_model_zone22 <- randomForest(Zone.2..Power.Consumption ~ Temperature
+                               + Humidity + Wind.Speed + general.diffuse.flows
+                               + diffuse.flows+ Hour + DayOfWeek,
+                               data = training_set,
+                               ntree = 500)
> # Train Random Forest model for Zone 3 Power Consumption
> rf_model_zone33 <- randomForest(Zone.3..Power.Consumption ~ Temperature
+                               + Humidity + Wind.Speed + general.diffuse.flows
+                               + diffuse.flows + Hour + DayOfWeek,
+                               data = training_set,
+                               ntree = 500)

> test_set$Hour <- as.numeric(format(test_set$DateTime, "%H"))
> test_set$DayOfWeek <- as.factor(weekdays(as.Date(test_set$DateTime)))
```

#add hour and dayofweek to test too before we predict using our test set

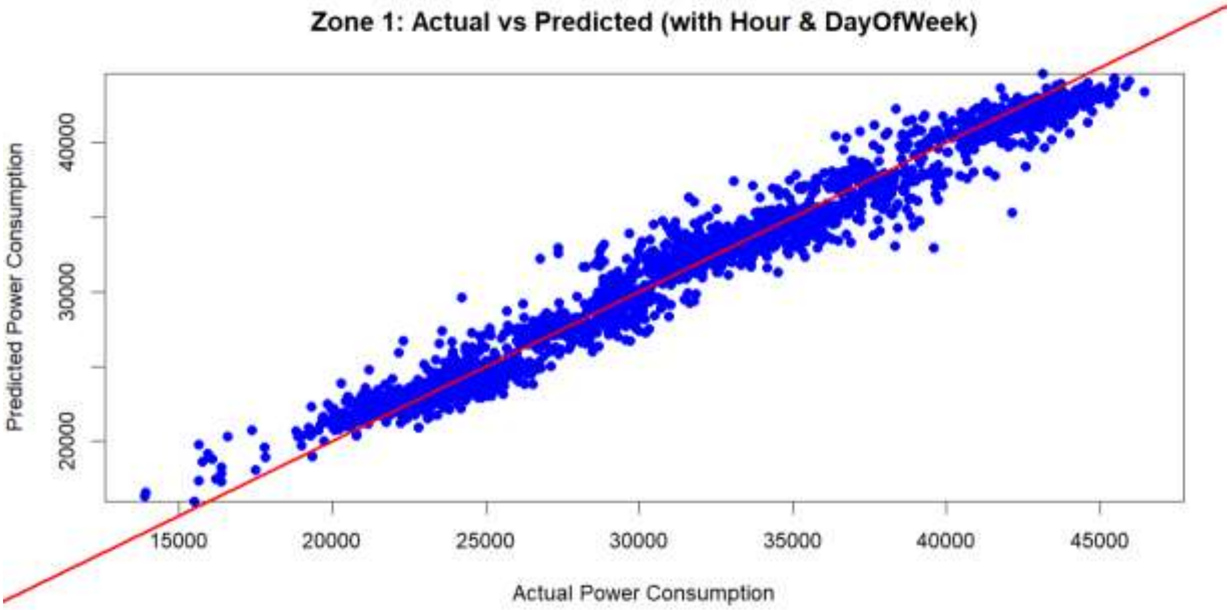
```
> test_set$Hour <- as.numeric(format(test_set$DateTime, "%H"))
> test_set$DayOfWeek <- as.factor(weekdays(as.Date(test_set$DateTime)))
```

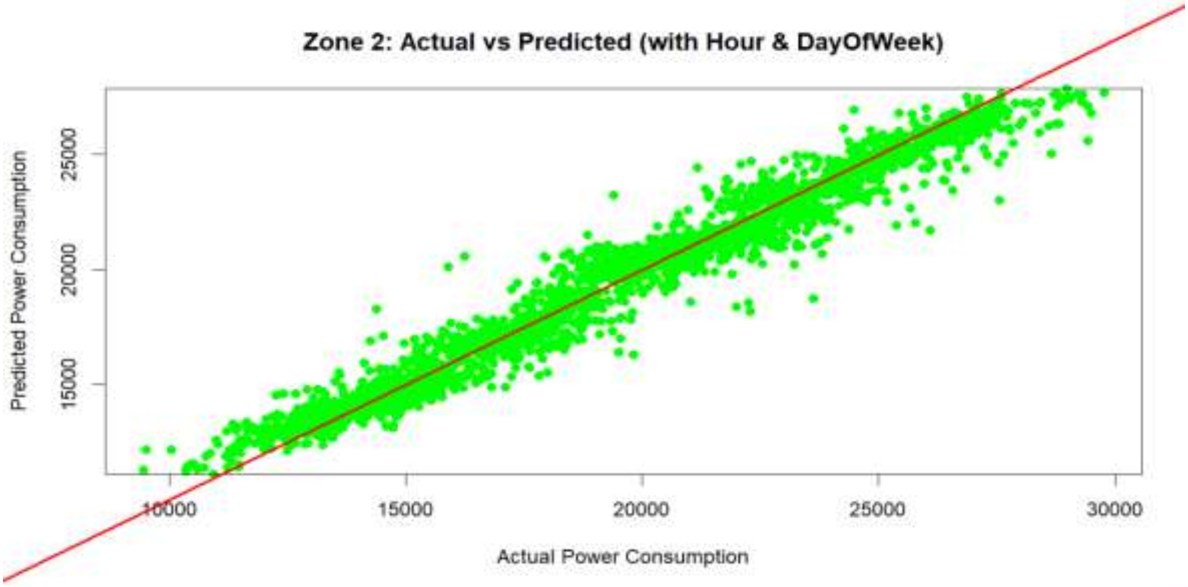
Improved Models:

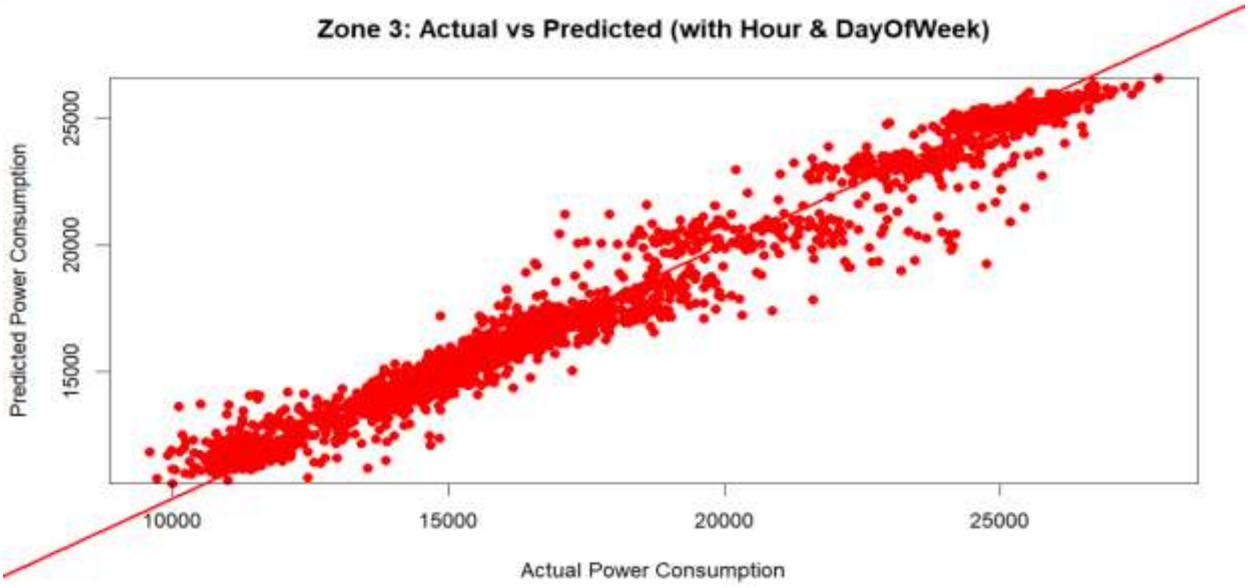
```
> cat("Zone 1 (with Hour & DayOfWeek) RMSE:", zone11_rmse, "\nZone 1 R-squared:", zone11_r2, "\n")
Zone 1 (with Hour & DayOfWeek) RMSE: 1154.46
Zone 1 R-squared: 0.9741607
> cat("Zone 2 (with Hour & DayOfWeek) RMSE:", zone22_rmse, "\nZone 2 R-squared:", zone22_r2, "\n")
Zone 2 (with Hour & DayOfWeek) RMSE: 801.8158
Zone 2 R-squared: 0.9682869
> cat("Zone 3 (with Hour & DayOfWeek) RMSE:", zone33_rmse, "\nZone 3 R-squared:", zone33_r2, "\n")
Zone 3 (with Hour & DayOfWeek) RMSE: 740.9489
Zone 3 R-squared: 0.9717222
```



Actual vs Predicted Power Consumption after improvement in model:







Creating 2 new features that are "TwoHourInterval" and "Weekend":

```
# Create 2-hour intervals
data$TwoHourInterval <- as.numeric(format(data$DateTime, "%H")) %/% 2

# Create a weekend/weekday feature
# ifelse(..., "Weekend", "Weekday"):
# SYNTAX: ifelse(condition, true_value, false_value)
data$Weekend <- ifelse(weekdays(as.Date(data$DateTime)) %in% c("Saturday", "Sunday"), "Weekend", "Weekday")
data$Weekend <- as.factor(data$Weekend)
```

Training model on new features:

```
# Train the Random Forest Model for Zone 1, Zone 2, and Zone 3 Power Consumption
rf_model_zone111 <- randomForest(Zone.1.Power.Consumption ~ Temperature + Humidity + Wind.Speed
                                 + general.diffuse.flows + diffuse.flows +
                                   TwoHourInterval + Weekend,
                                data = training_set, ntree = 500)

rf_model_zone222 <- randomForest(Zone.2..Power.Consumption ~ Temperature + Humidity + Wind.Speed
                                 + general.diffuse.flows + diffuse.flows +
                                   TwoHourInterval + Weekend,
                                data = training_set, ntree = 500)

rf_model_zone333 <- randomForest(Zone.3..Power.Consumption ~ Temperature + Humidity + Wind.Speed
                                 + general.diffuse.flows + diffuse.flows +
                                   TwoHourInterval + Weekend,
                                data = training_set, ntree = 500)
```

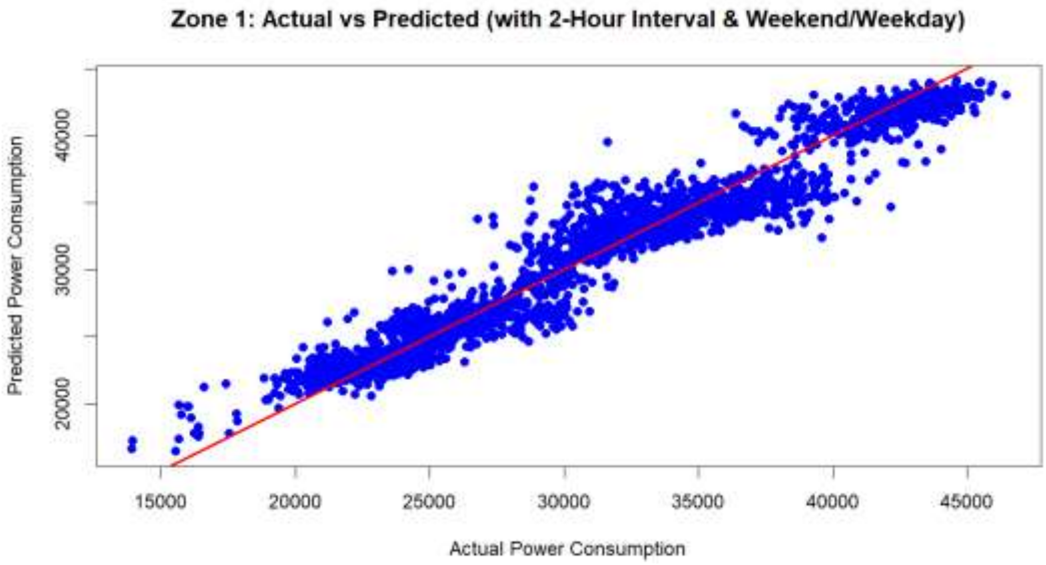
Further same process for model evaluation:

```
# Display model summaries
print(rf_model_zone111)
print(rf_model_zone222)
print(rf_model_zone333)

# Predict Power Consumption on the Test Set
pred_zone111 <- predict(rf_model_zone111, newdata = test_set)
pred_zone222 <- predict(rf_model_zone222, newdata = test_set)
pred_zone333 <- predict(rf_model_zone333, newdata = test_set)

# Model Evaluation
rmse <- function(actual, predicted) {
  sqrt(mean((actual - predicted)^2))
}

# Zone 1 Evaluation
zone111_rmse <- rmse(test_set$Zone.1.Power.Consumption, pred_zone111)
zone111_r2 <- summary(lm(pred_zone111 ~ test_set$Zone.1.Power.Consumption))$r.squared

# Zone 2 Evaluation
zone222_rmse <- rmse(test_set$Zone.2..Power.Consumption, pred_zone222)
zone222_r2 <- summary(lm(pred_zone222 ~ test_set$Zone.2..Power.Consumption))$r.squared

# Zone 3 Evaluation
zone333_rmse <- rmse(test_set$Zone.3..Power.Consumption, pred_zone333)
zone333_r2 <- summary(lm(pred_zone333 ~ test_set$Zone.3..Power.Consumption))$r.squared
```

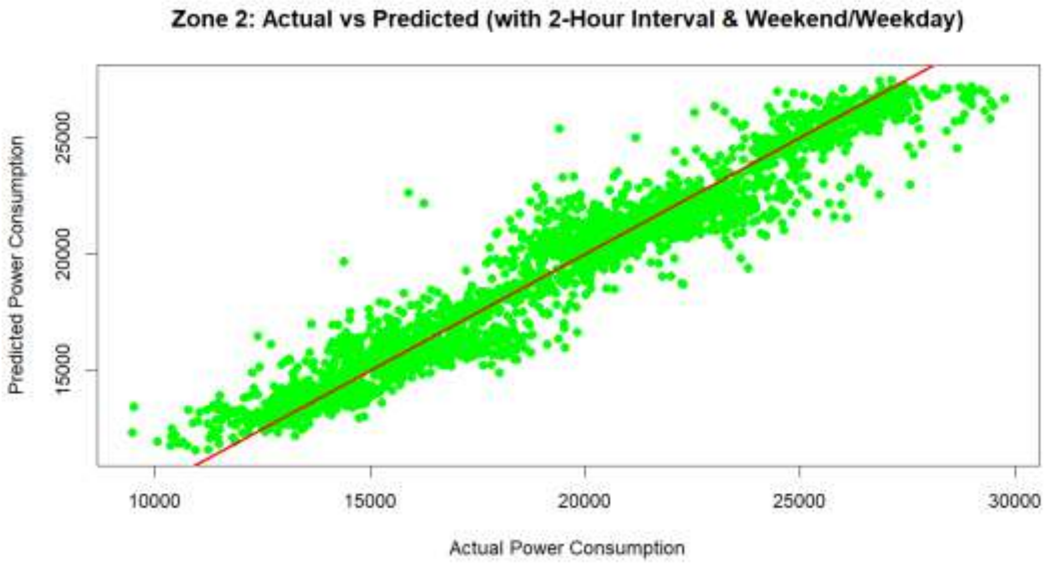
Newly trained model metrics:

```
> cat("Zone 1 RMSE:", zone111_rmse, "\nZone 1 R-squared:", zone111_r2, "\n")
Zone 1 RMSE: 1534.203
Zone 1 R-squared: 0.9533729
> cat("Zone 2 RMSE:", zone222_rmse, "\nZone 2 R-squared:", zone222_r2, "\n")
Zone 2 RMSE: 1065.785
Zone 2 R-squared: 0.9429015
> cat("Zone 3 RMSE:", zone333_rmse, "\nZone 3 R-squared:", zone333_r2, "\n")
Zone 3 RMSE: 994.2705
Zone 3 R-squared: 0.9478464
>
```

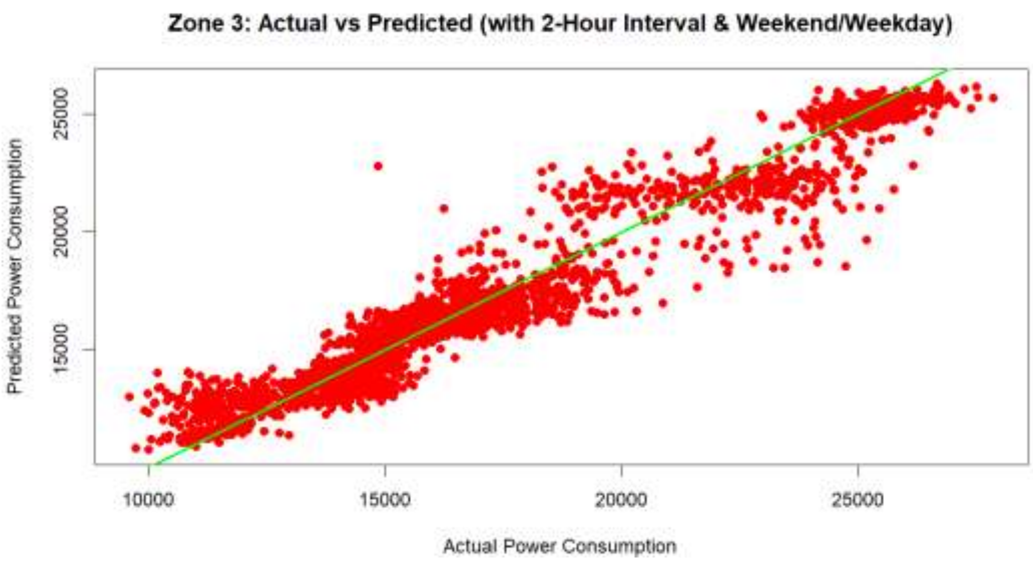
Zone One Predictions:



Zone Two Predictions:



Zone Three Predictions:



Example Scenario:
You are monitoring the electricity consumption for **Zone 3** on a **Monday** at **10:30 AM** (which falls in the 2-hour interval from 10:00 AM to 12:00 PM). You want to predict the power consumption during this time using our trained model, which is based on two-hour intervals and whether it's a weekend or a weekday.

```
> new_data <- data.frame(
+   DateTime = as.POSIXct("2024-10-02 10:30:00"),  # Monday, 10:30 AM
+   Temperature = 20.5,                             # Assume temperature is 20.5°C
+   Humidity = 65,                                  # Assume humidity is 65%
+   Wind.Speed = 1.00,                              # Assume wind speed is 1 m/s
+   general.diffuse.flows = 120,                    # Assume general diffuse flow is 120 W/m²
+   diffuse.flows = 100,                            # Assume diffuse flow is 100 W/m²
+   TwoHourInterval = 5,                            # As per 10:00 AM - 12:00 PM window
+   Weekend = factor("Weekday", levels = c("Weekday", "Weekend"))  # Monday is a weekday
+ )
> new_data$TwoHourInterval <- as.numeric(format(new_data$DateTime, "%H")) %/% 2
> new_data$Weekend <- ifelse(weekdays(as.Date(new_data$DateTime)) %in% c("Saturday", "Sunday"), "Weekend", "Weekday")
> new_data$Weekend <- factor(new_data$Weekend, levels = levels(training_set$Weekend))
> predicted_zone3_power <- predict(rf_model_zone333, new_data)
> cat("The predicted power consumption for Zone 3 at 10am to 12pm AM on Monday is:", predicted_zone3_power, "kW")
The predicted power consumption for Zone 3 at 10am to 12pm AM on Monday is: 15557.34 kW
> |
```

The Model successfully predicted power consumption for Zone 3 at 10:30AM on Monday: 15557.34 kW

Conclusion:

This project effectively used Random Forest models to predict power consumption across three zones by leveraging a range of environmental and temporal features. The initial models included variables such as **Temperature**, **Humidity**, **Wind Speed**, **general diffuse flows**, and **diffuse flows**. These features provided a reasonable baseline for predictions.

Key Improvements:

- **Incorporating Temporal Features**: By adding `Hour` and `DayOfWeek` to the models, there was a significant improvement in prediction accuracy across all zones.
  - **Zone 1**: RMSE reduced from 5249.296 to 1154.46.
  - **Zone 2**: RMSE reduced from 3366.984 to 801.8158.
  - **Zone 3**: RMSE reduced from 3265.414 to 740.9489.
- We later also added TwoHourInterval and Weekend features for additional experimentation.
- **Environmental Factors**: The models demonstrated that factors such as **Temperature**, **Humidity**, and **Wind Speed** play crucial roles in determining power consumption patterns. The fine-tuning of the models with additional features (e.g., **general diffuse flows** and **diffuse flows**) provided a more nuanced understanding of how environmental conditions impact energy usage.

Final Note:

The inclusion of both environmental and temporal features greatly enhanced the predictive accuracy of the models. This holistic approach underscores the importance of considering multiple factors when forecasting power consumption, offering valuable insights for optimizing energy management strategies in future applications.