Crime/Murder in California Cities

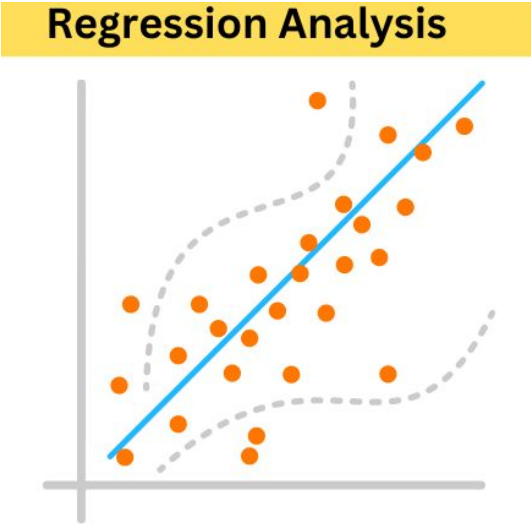
Tigran Manukyan, Aigerim Toleukhanova

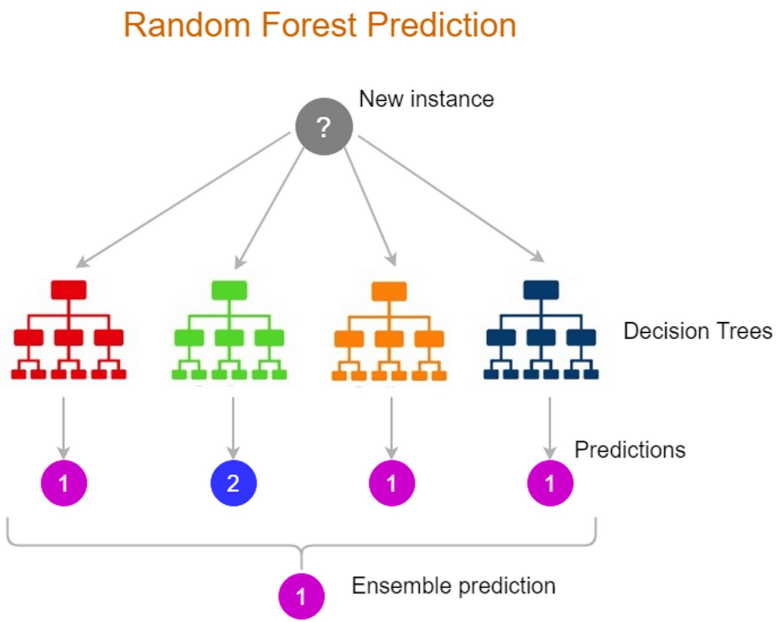
Problem

- Predicting the number of murders in California cities, based on the different crimes committed in those cities.
- FBI
- ~6000 tuples

Task Distribution

- Linear Regression by Aigerim
- Random Forest by Tigran





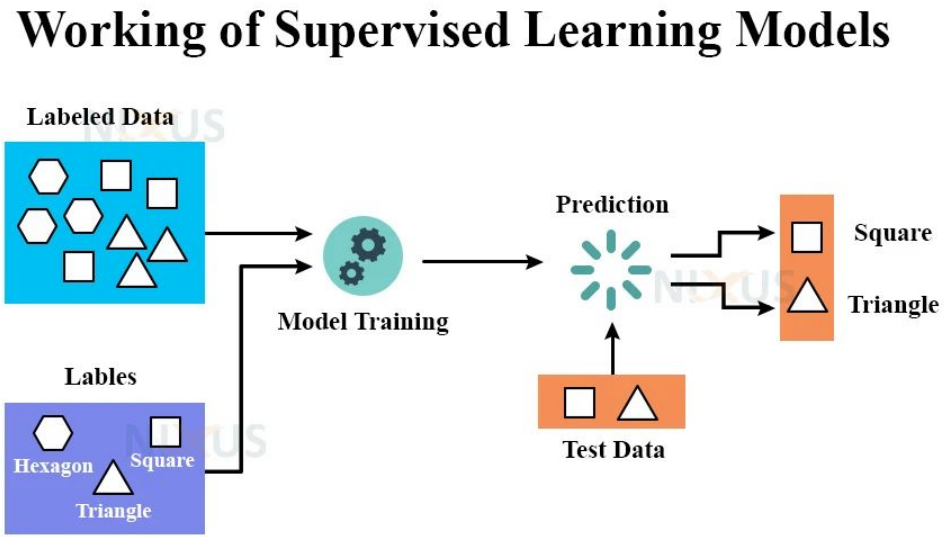
Supervised Learning

- Data is Labeled
- Target exists
- Regression :

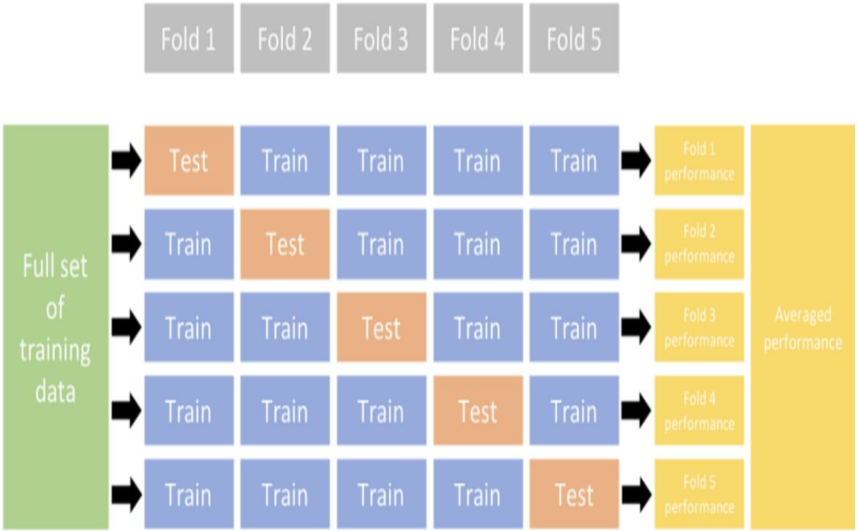
    Predict continuous numerical values

- Classification:

    Predict categorical target

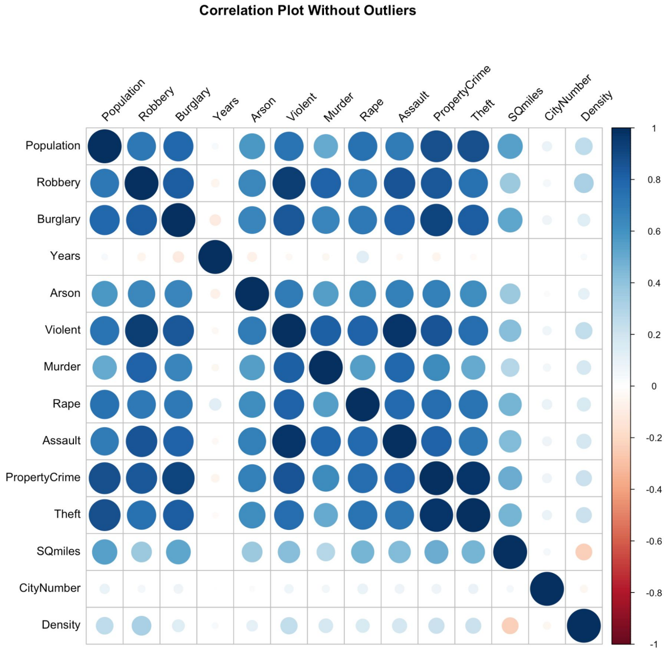


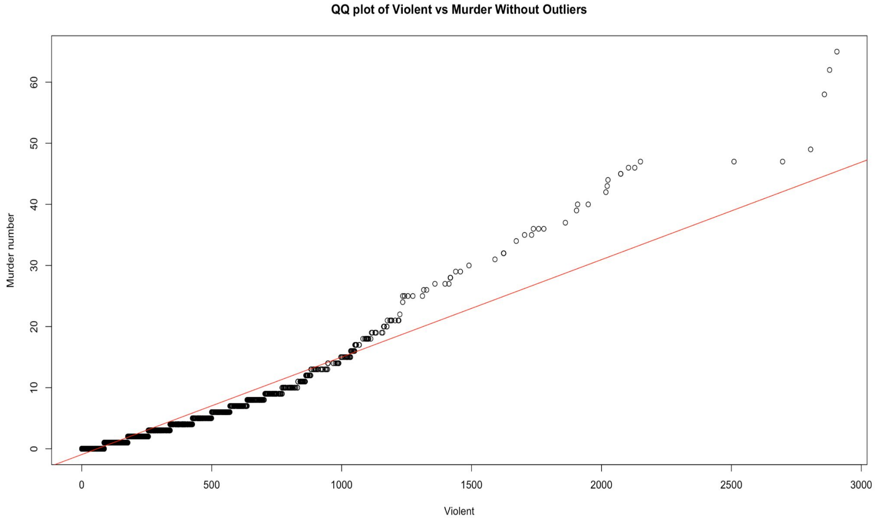
Training vs Testing: K-fold Cross Validation



- Rotation estimation
- Improve model
- Avoid overfitting and underfitting
- Train and predict all objects

Linear Regression: Analysis Motivation





Linear Regression: ?lm or help(lm)

Dataset: $\{y_i,\, x_{i1},\ldots,x_{ip}\}_{i=1}^{n}$

Model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta} + \varepsilon_i$$

i = 1, 2, ..., n,
where $^\mathsf{T}$ denotes the transpose, so that $x_i^\mathsf{T}\beta$ is the inner product between vectors x_i and β

- Statistical model
- Estimates the linear relationship between one or more variables
- Dependent or independent variables
- β is regression coefficient
- ?lm or help(lm)

Linear Regression: Code in R

```
library(tidyverse)

modelLR<-train(formula, data = mergedMurder, method = "lm",

          trControl =train_control)

predictions<-predict(modelLR, mergedMurder)
```

Linear Regression: Formula

- formula <- Murder ~ .
- formula <- Murder ~ Population+ Violent+ Robbery+ Burglary+Rape+ Assault+ PropertyCrime
- formula <- Murder ~ Violent+ Robbery+ Burglary+Arson+Rape+ Assault+ PropertyCrime+ Theft

Linear Regression: (modelLR)

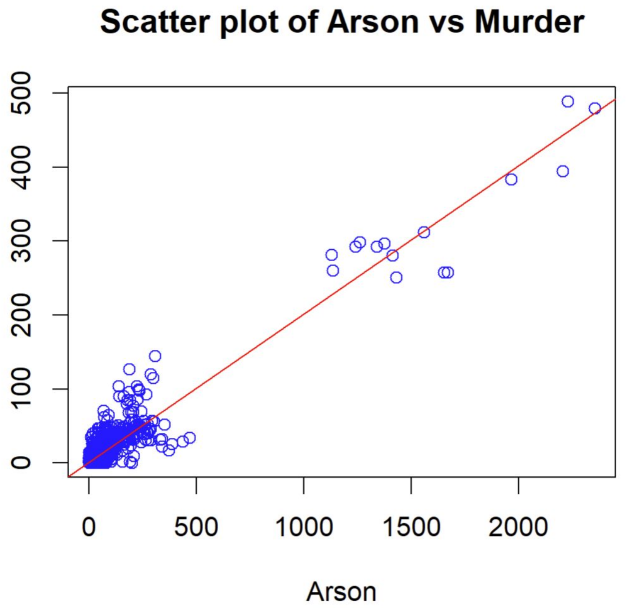
Linear Regression

6288 samples
8 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 5030, 5030, 5031, 5030, 5031
Resampling results:

RMSE	Rsquared	MAE
9.5823687112e-13	1	6.6798655415e-13

Tuning parameter 'intercept' was held constant at a value of TRUE

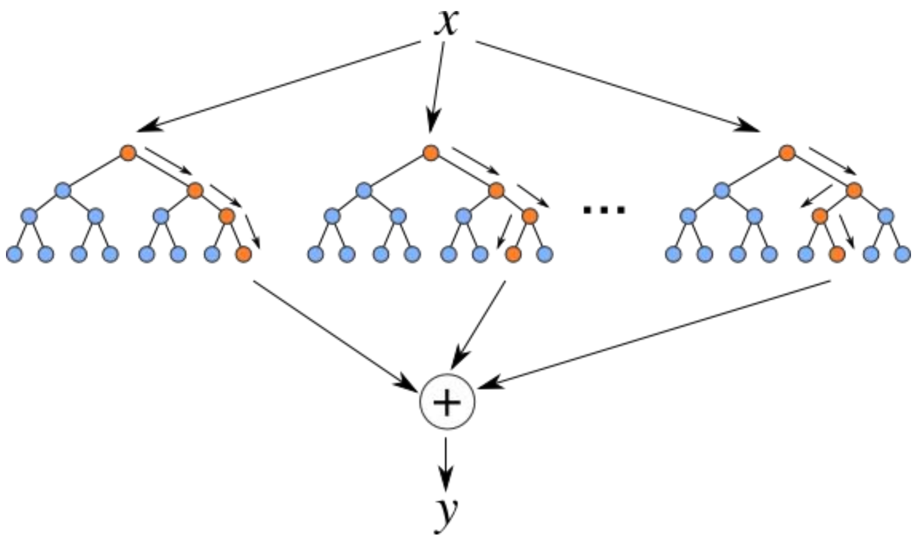


Random Forest: Motivation

I like trees so it Decision Trees for me.

Random Forest > Regular Decision Tree

- Reduces Overfitting
- No Pruning (although can help)
- Better with noise

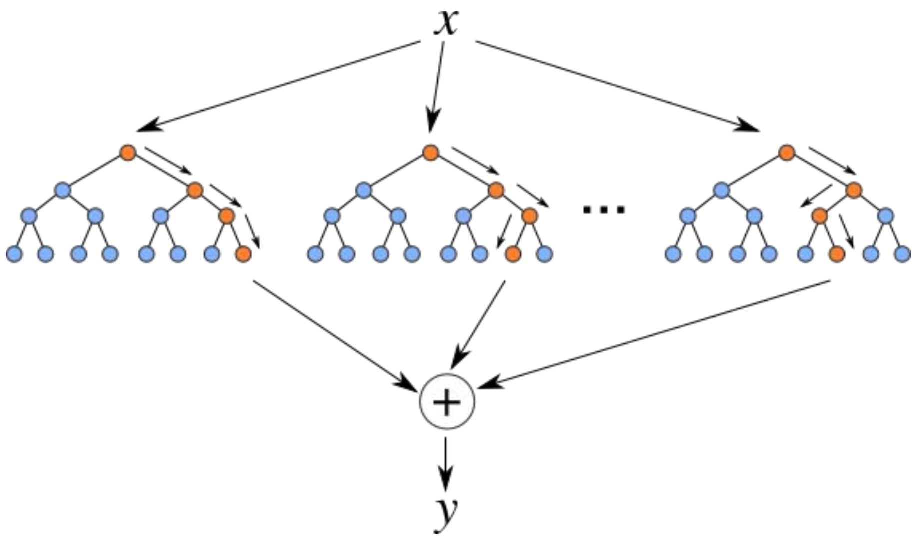


Random Forest: How It Works

Builds user defined number of Decision Trees (N):

For each split point of a decision tree

- Randomly selects a number of features (m) from data, and chooses the best attribute from the subset
- M is a number between 1 and number of features



Random Forest: How It Works

Evaluation

- Classification:
  - Chooses the output that occurs most frequently
- Regression
  - Takes the average out of all the outputs
  - Formula:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

  - Where y_i is the output of the ith decision tree

Random Forest: Code

```r
# Define the number of folds for cross-validation
num_folds <- 5

# Set up the cross-validation scheme
ctrl <- trainControl(method = "cv", number = num_folds)
```

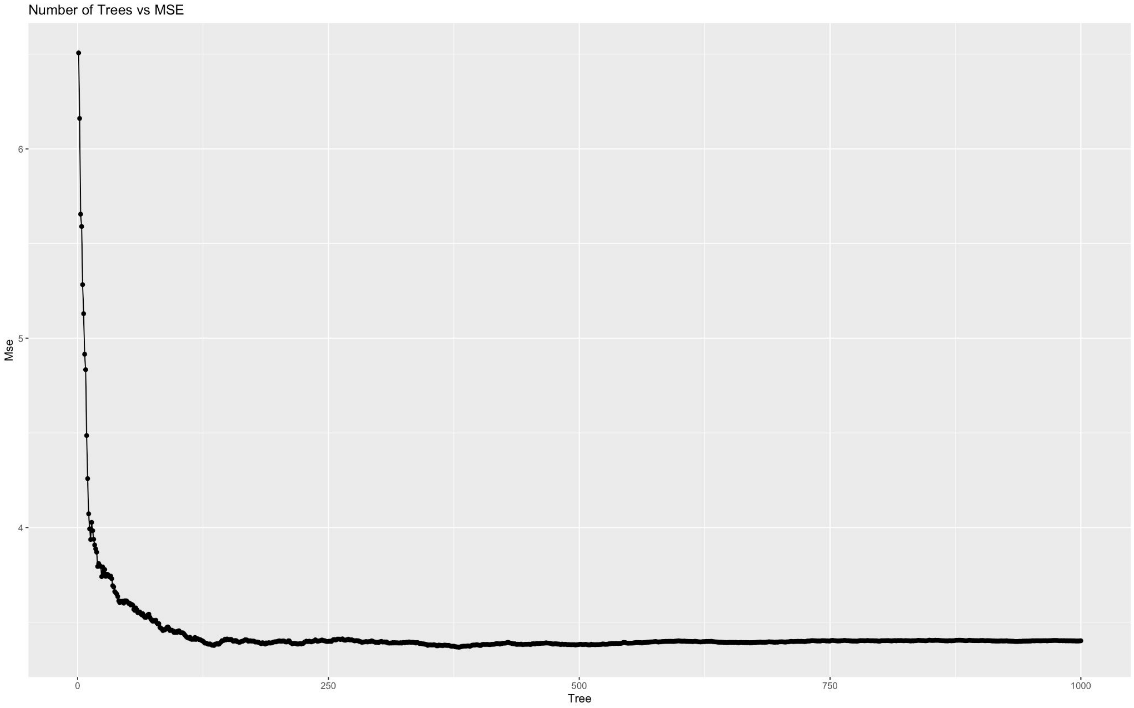
```r
# Train Random Forest model using k-fold cross-validation
rf_model <- train(
  x = murderData[, -which(names(murderData) == "Murder")],  # Predictors
  y = murderData$Murder,  # Target variable
  method = "rf",
  trControl = ctrl,
  tuneGrid = expand.grid(mtry = 1:10),  # Grid of mtry values to try
  ntree = 500
  )
```

library(randomForest)

Random Forest: Tuning

Number of Trees:

- Once number of trees gets big enough it has now effect on the accuracy
- We will use 500



Random Forest: Tuning

Variables tried at each split:

- mtry is variables tried at each split
- If we compare the RMSE the best one is 5
- So 5 variables are randomly selected at each split point

	mtry	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	1	1.915505	0.7758536	1.026560	0.2233059	0.04396384	0.05561283
2	2	1.860022	0.7852879	1.017678	0.1769859	0.03293316	0.05169909
3	3	1.851209	0.7869588	1.019853	0.1597497	0.02979185	0.05042968
4	4	1.847615	0.7878483	1.020679	0.1453453	0.02732317	0.04734661
5	5	1.847082	0.7882261	1.022831	0.1409833	0.02474072	0.04695352
6	6	1.851030	0.7876245	1.024002	0.1344879	0.02297482	0.04550555
7	7	1.852954	0.7874130	1.026063	0.1274865	0.02185773	0.04650042
8	8	1.859817	0.7859601	1.028236	0.1279782	0.02070695	0.04456948
9	9	1.858056	0.7867935	1.026075	0.1232412	0.01875030	0.04472329
10	10	1.854645	0.7870214	1.027237	0.1263919	0.01838821	0.04249963

```
"Best No. of variables tried at each split: 5"
```

Random Forest: Final Forest

```
finalForest <- train(
  x = murderData[, -which(names(murderData) == "Murder")],  # Predictors
  y = murderData$Murder,  # Target variable
  method = "rf",
  trControl = ctrl,
  tuneGrid = expand.grid(mtry = 5),  # best mtry model
  ntree = 500
)
```

Random Forest: MISC

- Random Forest training also gives the importance of each variable as a bonus.
- For us Violent Crime was the most important
- Robbery and Assault are significantly important
- Rest not as important

rf variable importance

	Overall
Violent	100.0000
Robbery	59.2716
Assault	30.0079
Burglary	7.8878
PropertyCrime	5.6624
Population	4.5514
Theft	3.5147
Arson	3.4622
SQmiles	2.9727
Years	0.1473
Rape	0.0000

Evaluation Metrics

1. Root Mean Square Error
2. Mean Square Error
3. Mean Absolute Error
4. Mean Absolute Percentage Error
5. Confusion Matrix:
    - Sensitivity
    - Specificity
    - Balanced Accuracy

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

$$\mathrm{MAPE} = 100\frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Linear Regression: Result by summary(modelLR)

```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
           Min             1Q         Median
-4.1663170e-11 -3.8838000e-15  3.3667000e-15
            3Q            Max
 1.0014800e-14  4.2943003e-11
```

⋮

```
Residual standard error: 8.0068054e-13 on 6279 degrees
of freedom
Multiple R-squared:          1, Adjusted R-squared:
1
F-statistic: 2.1827389e+28 on 8 and 6279 DF,  p-value:
< 2.22045e-16
```

- Residual (differences between observed and predicted) is symmetric
- Residual Standard Error (average distance that the observed values fall from the regression line) is low
- F-statistic (measure of how well the model fits as a whole)
- Small p-value (< 0.05) is good

Linear Regression: Result 5-fold Cross-Validation

formula <- Murder ~ **Violent+ Robbery+** Burglary+ Arson+**Rape+ Assault+** PropertyCrime+ Theft+ SQmiles

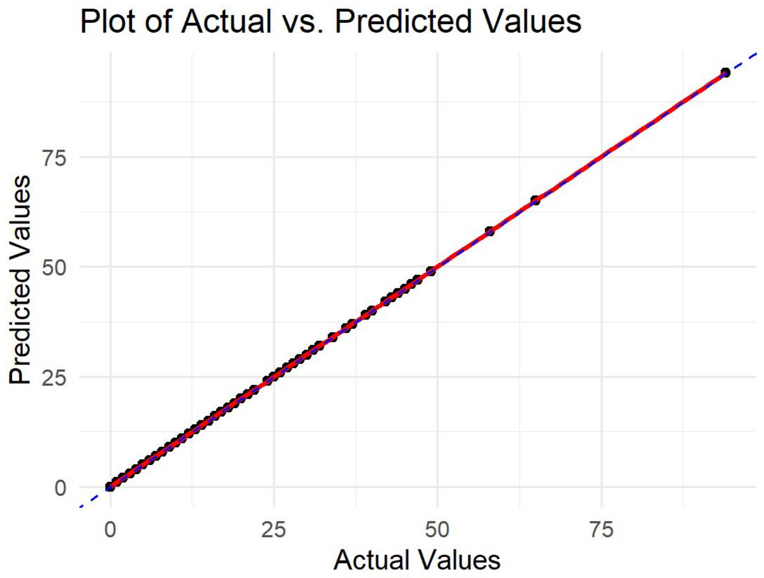
MAE: 0.00000000000067703531147

MSE: 6.3740991771e-25

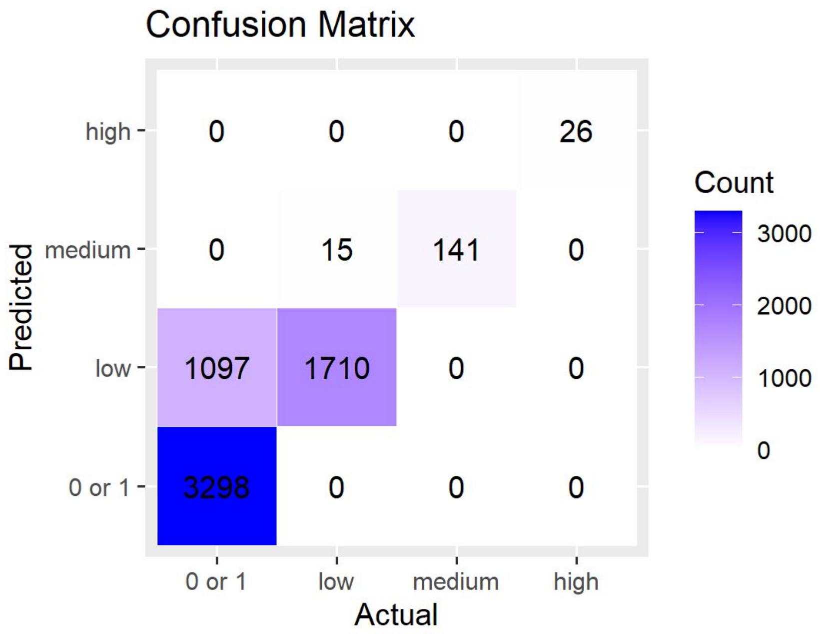
RMSE: 0.00000000000079837955742

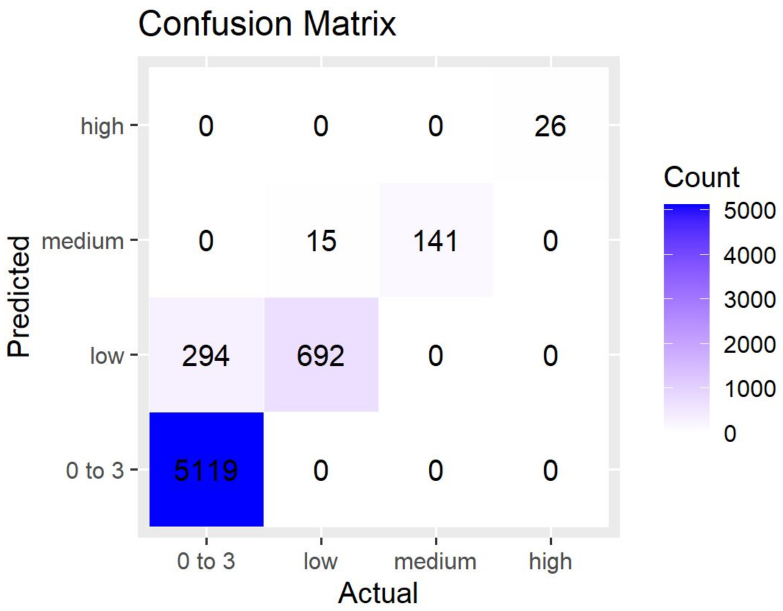
MAPE: 0.000035696965786 %

Accuracy: **99.999964303 %**



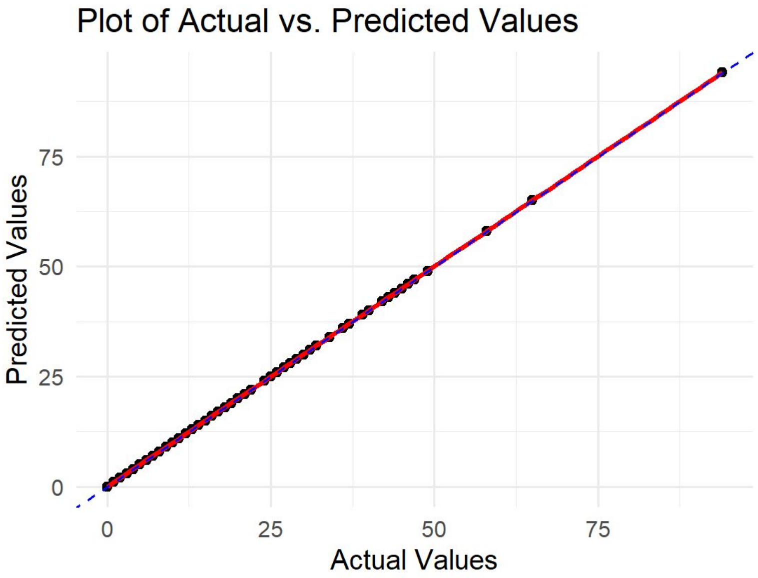
Linear Regression: Result of Assigning Range

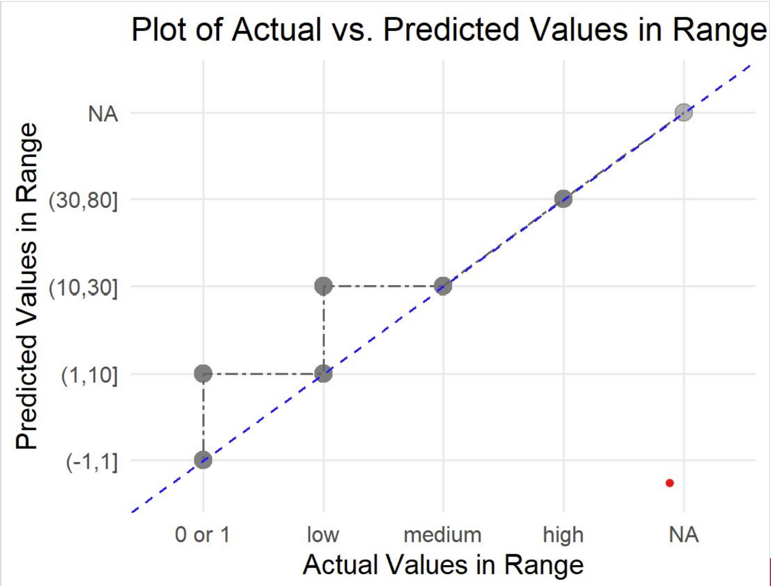




Accuracy : **0.82312709** vs **0.95085096**

Linear Regression: Result





Random Forest: Result

MSE: 0.602320576613915

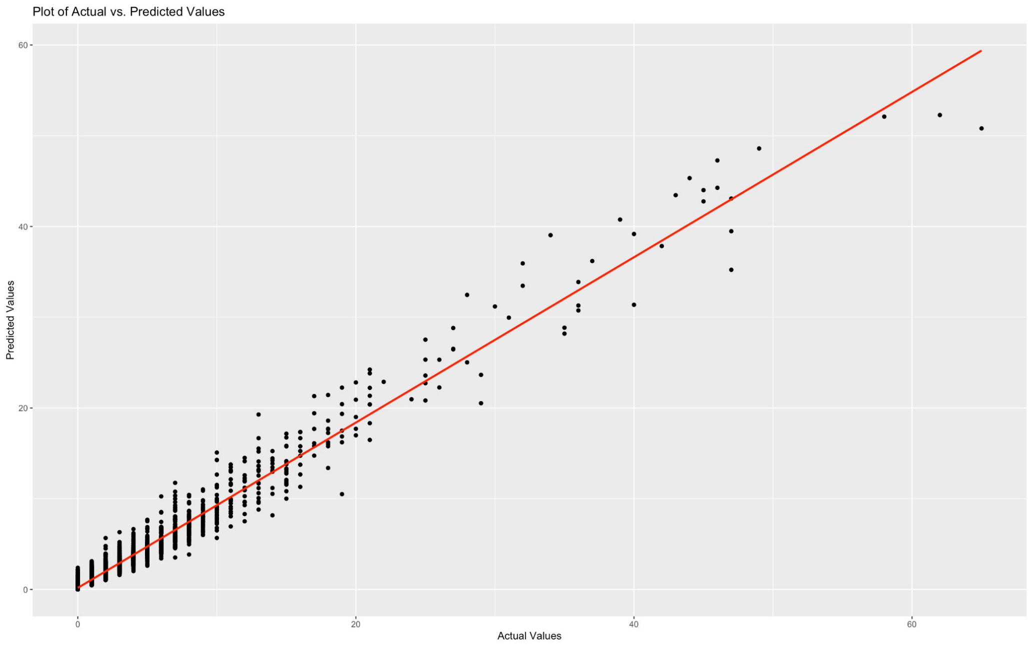
RMSE: 0.776093149444005

MAE: 0.430247166891318

R-squared: 0.965121456446335

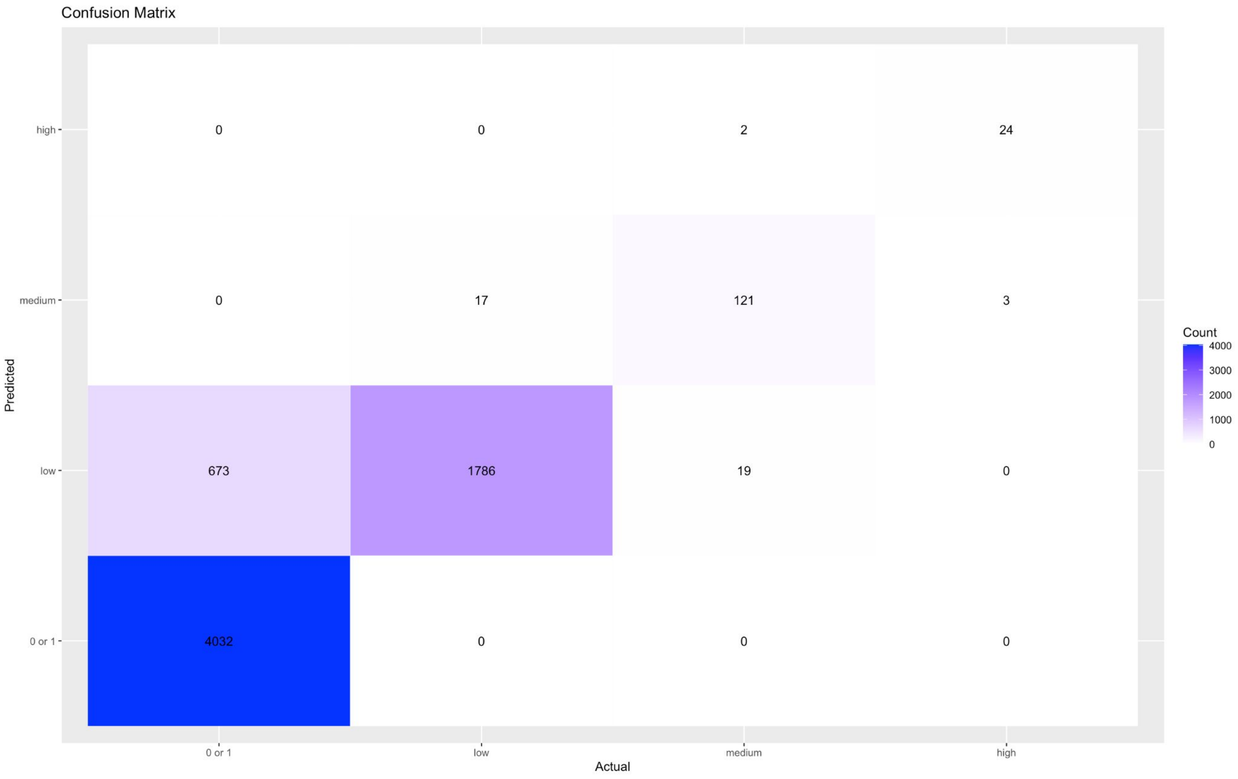
MAPE: 19.1058273502418%

Accuracy: **80.8941726498%**



Random Forest: Result

- Accuracy : 89.31%
- Results better then Linear Regression
- Better for classification models not regression



Comparing the Result

Linear Regression:

MAE: 0.00000000000067703531147

MSE: 6.3740991771e-25

RMSE: 0.00000000000079837955742

MAPE: 0.000035696965786 %

Accuracy: **99.999964303 %**

Random Forest:

MAE: 0.430247166891318

MSE: 0.602320576613915

RMSE: 0.776093149444005

MAPE: 19.1058273502418%

Accuracy: **80.8941726498%**

Conclusion

- Linear Regression gave better results
- Our data was very correlated so it makes sense
- However, random forest still gave decent and could prove useful for future tests

Linear Regression

Random Forest

Future work

- Add more features such demographic characteristics:
    - Race/nationality of population
    - Income of household/person
    - Dissect Cities into Neighborhoods
    - Consider safety Index of Neighborhoods
    - School rating
- Adapt Neural Network
    - Assign weight/threshold
    - Gave priority for some attributes

Thank you!
Questions?