## Training and Testing after finding best parameters using Cross Validation

```
method convert_time_to_quarter(time):
    hours, mins <- split the time
    return int(hours/6)
method map(row)
    new_features <- []
    vendor_id, pickup_datetime, num_passengers, pickup_latitude,
    pickup_longitude, dropoff_latitude, dropoff_longitude, save_information,
    trip_duration <- split the row
    new_features.append(vendor_id)
    date, time <- split pickup_datetime
    year, month, date <- split date
    day <- weekday(year, month, date)
    new_features.append(day as either 0 or 1 for all days of week)
    quarter <- convert_time_to_quarter(time)
    new_features.append(quarter as either 0 or 1 for all quarters of the day)
    new_features.append(month as either 0 or 1 for all months in the dataset)
    latitude_distance <- pickup_latitude - dropoff_latitude
    new_features.append(latitude_distance)
    longitude_distance <- pickup_longitude - dropoff_longitude
    new_features.append(longitude_distance)
    manhattahn_distance(|latitude_distance| + |longitude_distance|)
    new_features.append(manhattan_distance)
    if save_information == ‘N’ do
```

```
            new_features.append(0)
        **else do**
            new_features.append(1)
        new_features.append(trip_duration)
        **return** new_features


**method** parse_data(line)
        features, result <- split the line
        **return** LabeledPoint(result, features)
**method** main()
        read text file as RDD
        RDD.map(split row into raw_features)
        RDD.filter(remove first column)
        RDD.map(map)
        RDD.map(parse_data)
        Split RDD into TrainRDD and TestRDD
        Shuffle TrainRDD
        train a Linear Regression Model on TrainRDD with best parameters
        RDD <- predict on TestRDD using the trained model
        SquareError <- RDD.map(square(true_value – predicted_value))
        MeanSquareError <- RDD.reduce(sum all squares)/RDD.count()
        RootMeanSquareError <- SquareRoot(MeanSquareError)
        Print(RootMeanSquareError)
        AbsoluteError <- RDD.map(absolute(true_value – predicted_value))
        MeanAbsoluteError <- RDD.reduce(sum all absolutes)/RDD.count()
        Print(MeanAbsoluteError)
```