**CS-4331: Data Mining**

**Project II**

**Pritish Ayer**

**Nicholas Lovera**

**Jiayu yan**

**Contribution:**

**Setup Phase**: Pritish Ayer

**Modeling Phase:** Multiple Regression Modeling: Pritish Ayer

Neural Networs: Jiayu Yan

**Evaluation Phase**: Multiple Regression Modeling: Pritish Ayer

Neural Networs: Jiayu Yan

Report Created by: Pritish Ayer, Jiayu Yan

Link to the dataset: <https://data.ny.gov/Energy-Environment/Solar-Electric-Programs-Reported-by-NYSERDA-Beginn/3x8r-34rs>

**Preview:**

In the previous submission of this project we covered the problem understanding, data preparation and exploratory phase. To briefly recap:

**• Problem Understanding Phase:** Compare the cost and production of the solar energy and calculate the breakeven point.

**• Data Preparation Phase:**

* Got rid of many numbers of columns/variables to make the data more
* Changed the Annual production (‘Expected.KWh.Annual.Production’) to numeric
* Binning the data to set the labels as "Under 10000","10000 to 100000","Over 100000

• **Exploratory Data Analysis Phase**

* Partitioning the data
* Plotting multiple graphs

In this release the following steps were covered:

**Setup Phase:**

We took the processed data from the previous step and worked on it during the Setup phase. The three essential steps we did included:

1. **Partitioning the Data:** We used two-fold validation to partition the data. The data set was partitioned into training set and test set. It is recommended that for the simpler data set, the training data should be anywhere between 50-67 percent of the original data. However, for highly complex data sets, more training records would be recommended, such as 75-90% of the original data. Since, our dataset was fairly complex so we chose to do a 70/30 split in our training and test dataset respectively.

train\_ind <- runif(n) < 0.75

solar\_setup\_train <- solar\_setup[train\_ind,]

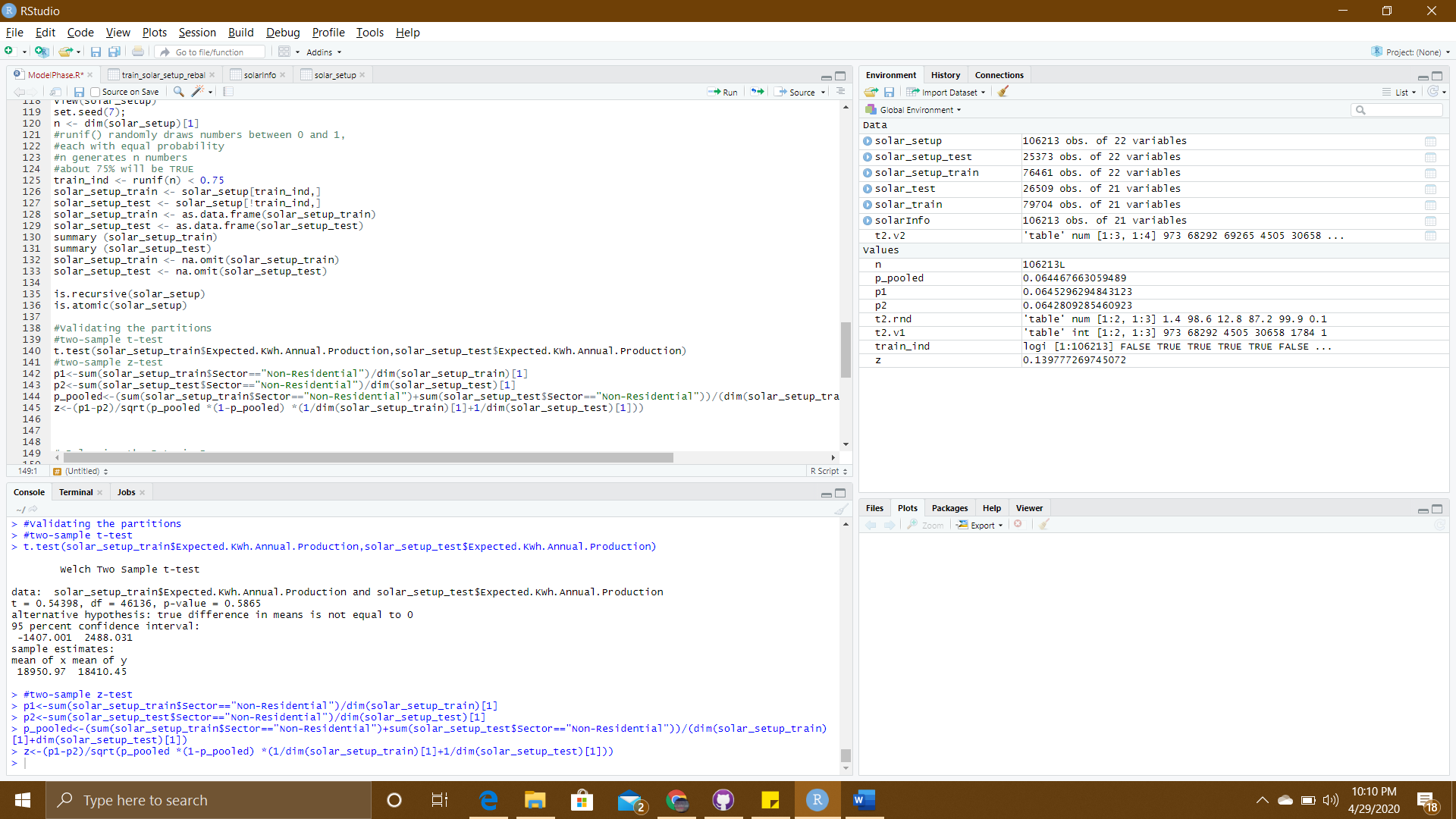
solar\_setup\_test <- solar\_setup[!train\_ind,]

1. **Validating the partition**: We had to absolutely ensure that the training and test sets are indeed independent. In order to achieve this validating the partition was necessary. Some of the famous ways to validate the data are:

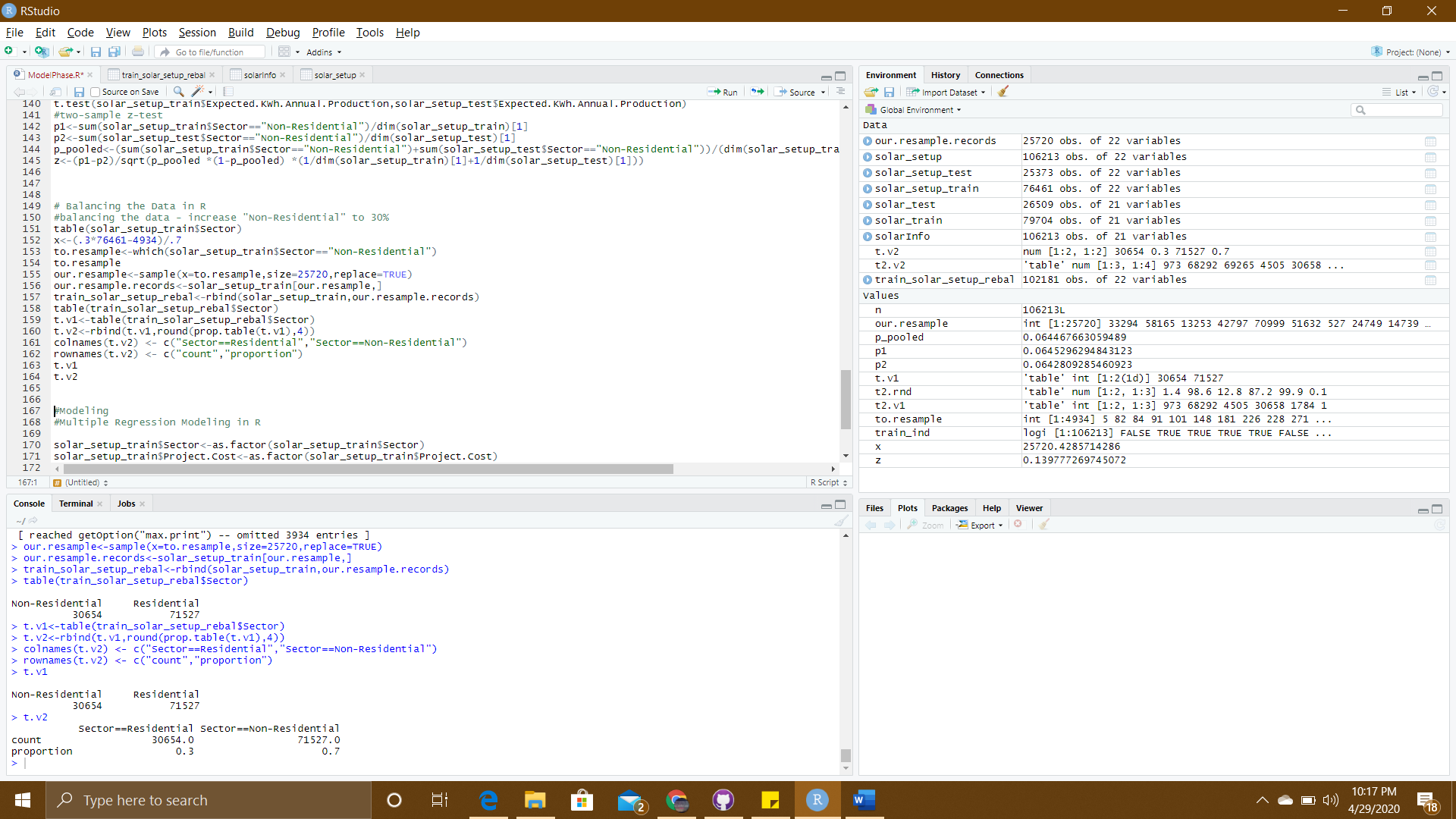
Numeric – two-sample t-test for the difference in means

Categorical variable with 2 classes – two-sample Z-test for the difference in proportions Categorical variable with more than 2 classes – test for the homogeneity of proportions

We used two-sample t-test for Expected.KWh.Annual.Production as it is numeric value. We used z-test to validate the Sector attribute for our dataset as Sector is two categorical attribute.



1. **Rebalancing the Training Data:** In our dataset, we found out the huge imbalance among the categorical data of Sector attribute. We had 4934 Non-Residential data and 71527 Residential data. This means that Non-Residential data only constitute 6% of the total data, so, we rebalanced the data to 30%.



**Modeling Phase:**

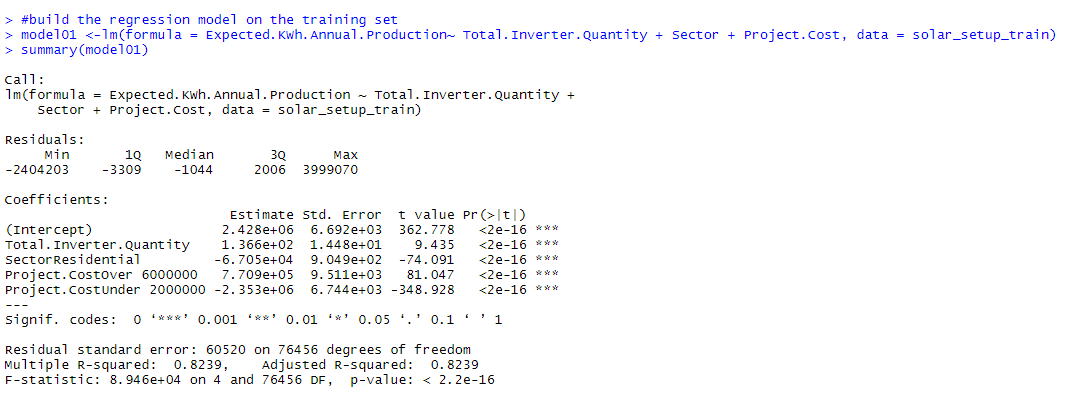
1. Multiple Regression Modelling: One of the model that we implemented was Multiple Regression Modeling. Different variables that were used for the modelling were:
2. Predictor Variable: Total.Inverter.Quantity: Numeric

Sector: Categorical

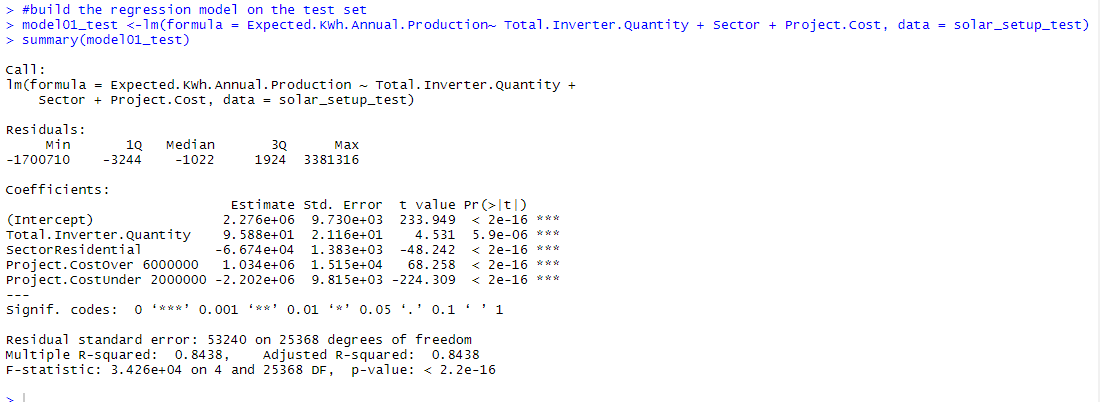
Project.Cost: Categorical

1. Target Variable: Expected.KWh.Annual.Production

The model built on training set:

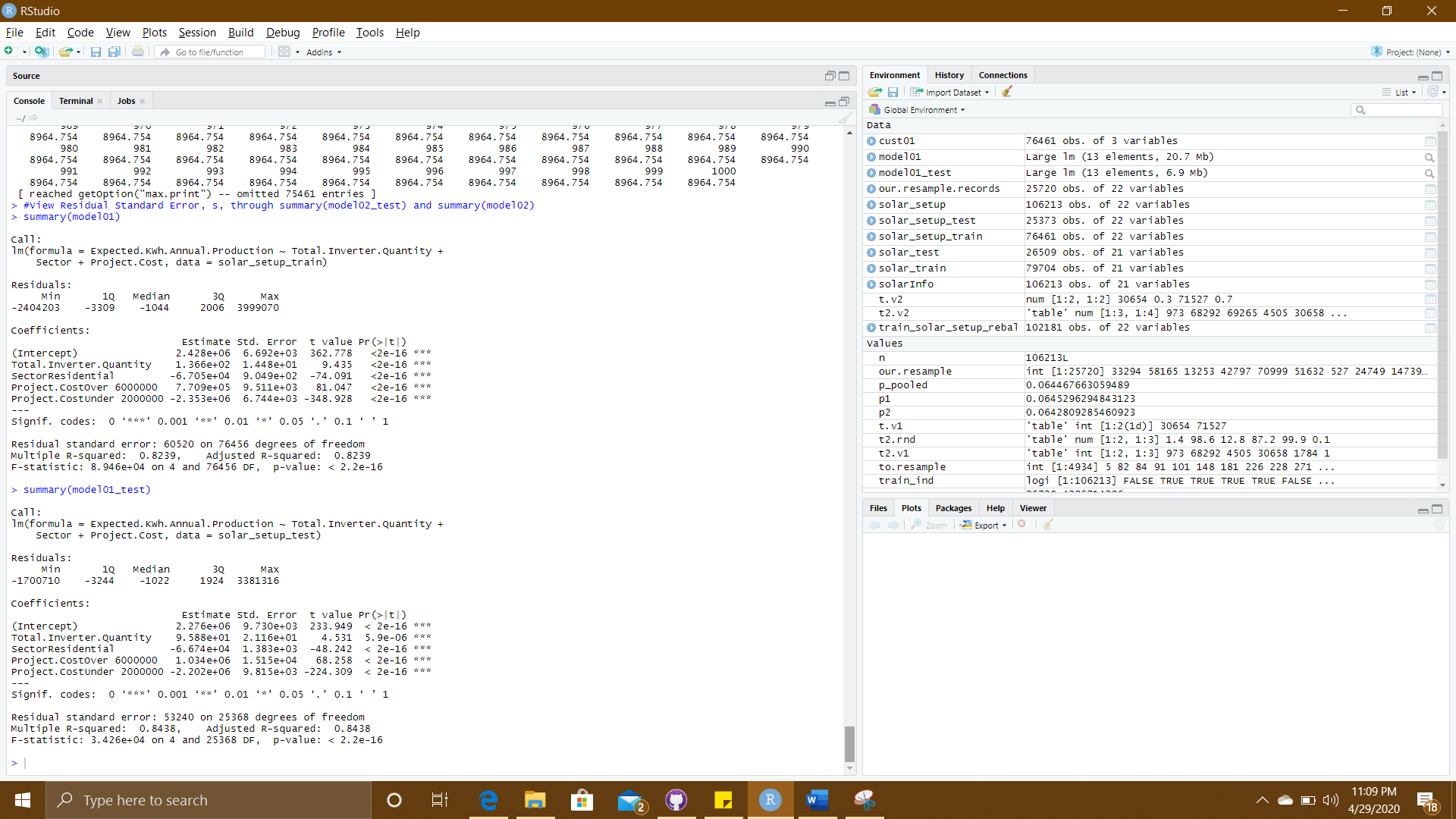


The model built on test set:



**Evaluation Phase:**

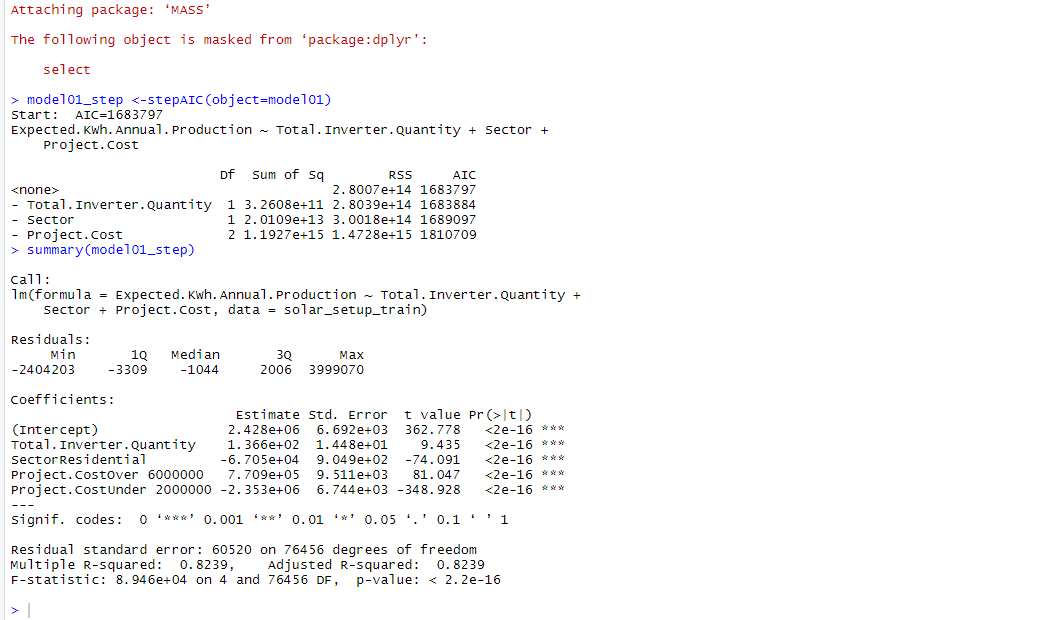
Firstly, we create a new data called cust\_01 to evaluate the model by keeping Inverter.Quantity as 10.



**Calculating Mean Absolute Error**



**Evaluating the Final Summary.**



**Model Phase: Neural Network**

Numeric Min-max Normalization for our expected annual production

>>solar\_setup\_train$Expected.KWh.Annual.Production.mm<-((solar\_setup\_train$Expected.KWh.Annual.Productionmin((solar\_setup\_train$Expected.KWh.Annual.Production))/(max((solar\_setup\_train$Expected.KWh.Annual.Production) -min((solar\_setup\_train$Expected.KWh.Annual.Production))

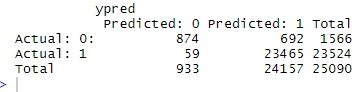
**Input Indicator Variables expected annual production and project cost to target variable Sector**

>>nnet01 <-nnet(Sector~Expected.KWh.Annual.Production+Project.Cost, data=solar\_setup\_train, size=1)

Then we can get the weights from nnet and analyze them.(Written in code file comments)

**Evaluation Phase: Neural Network**

We use test data from the set phase to do the evaluation for our model.



accuracy=0.9700678, precision= 0.9713541

As the results show, we have a decent model.

**Model Phase: Naive Bayes Classifier**

**Creating the Naive Bayes**

* Target Variable:
  + Sector
* Predictor Variables:
  + Expected.KWh.Annual.Production
  + Project.Cost

>nb01 <- naiveBayes(formula = Sector~ Expected.KWh.Annual.Production+Project.Cost, data = solar\_setup\_train)

>ypred <- predict(object = nb01, newdata = solar\_setup\_test)

**Creating table of actual vs. predicted**

>t.preds <- table(solar\_setup\_test$Sector, ypred)

>t.preds <- addmargins(A=t.preds,FUN=list(Total=sum),quiet=TRUE)

**A screenshot of a cell phone

Description automatically generatedEvaluation Phase:** **Naive Bayes Classifier**

t.preds

Accuracy = 0.935357

Precision = 0.935357

Although the accuracy and precision look good the table shows that Naïve Bayes is not a good model to be used on the solar\_Info data set.