Nicholas Jurgens

TIM 158

6/13/17

TIM 158 Final

**Problem 1: Planning**

1. **Define the Problem:**
   1. Create a plan and time-schedule for working on the problems below. Also, using an appropriate table (see Problem 6 below), track how well you execute your schedule, and make notes on issues and problems that prevent efficient execution.
2. **Create a Plan:**
   1. **Step 1:** Clearly state the intent of the midterm.
   2. **Step 2:** Determine the design/development sub-tasks and activities.
   3. **Step 3:** Create a design/development activity matrix purpose to understand the dependencies between the sub-task.
   4. **Step 4:** Create a schedule of tasks using a GANTT chart.
   5. **Step 5:** Identify the “critical path” for the project using PERT chart.
   6. **Step 6:** Create a table and update it with obstacles and problems that occur throughout the project.
3. **Execute**
   1. **Step 1:** The intent of the final is to gain a better understanding of the material. In order to do this I need to be able to complete all nine problems in a timely manner.
   2. **Step 2:**
      1. **A:** Planning
      2. **B:** Target Advertising For Uber
      3. **C:** Uber Surge Pricing
      4. **D:** Driver Attrition At Uber
      5. **E:** Software Architecture For Machine Learning Applications
      6. **F:** Execution Of Your Plan
   3. **Step 3:** Activity Matrix

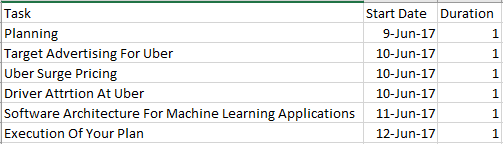
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** |
| **A** | **A** |  |  |  |  |  |
| **B** | **X** | **B** |  |  |  |  |
| **C** | **X** |  | **C** |  |  |  |
| **D** | **X** |  |  | **D** |  |  |
| **E** | **X** | **X** |  |  | **E** |  |
| **F** | **X** | **X** | **X** | **X** | **X** | **F** |

Notation:

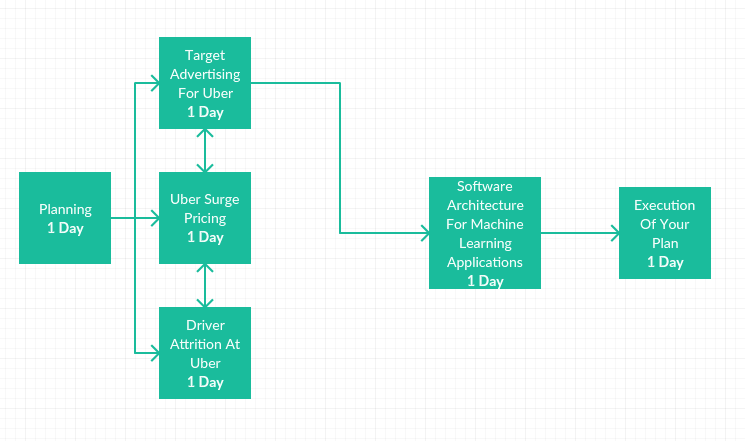
X = “depends on”

BxA = subtasks B depends on subtask A

* 1. **Step 4:** GANTT Schedule



* 1. **Step 5:** PERT Chart



* 1. **Step 6:**

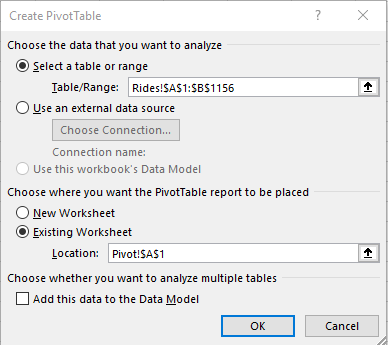
|  |  |
| --- | --- |
| **Problems that occurred** | **Solutions** |
| No major problems occurred. | N/A |

1. **Draw Conclusions**
   1. My plan was to have everything done by the 7th. However, I ran into problems and had to push some of the work back. This is due to underestimating how long each problem would take. I also didn’t take into account how long I would be working for my job. Next time I make a plan I will make sure to leave more room for error. This will help reduce any risk in surprises.

**Problem 2: Targeted Advertising for Uber**

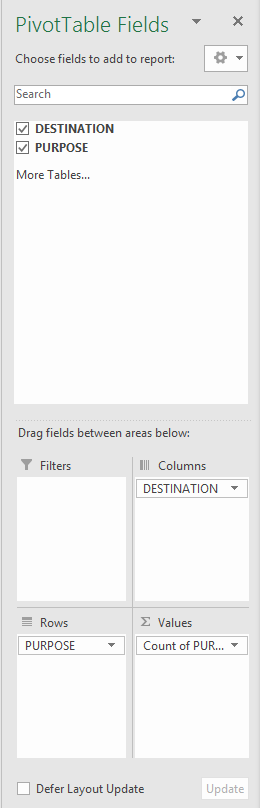
1. **Define the Problem:**
   1. In order to develop targeted advertising for its customer the Uber Marketing group has assembled the dataset shown in the workbook “*Uber\_Ads”*. This data set has two columns:
      1. The first column representing the destination of each trip
      2. The second column representing the purpose of each trip
   2. There are 10 well defined purposes such as “*Meals/Entertain”* and “*Meeting*”. Note that where no purpose is identified, the corresponding entry is a blank field. So, including the blank field, there are 11 purposes.
   3. The workbook “*Uber\_Ads*” contains the (destination, purpose” pair for 1155 customers. Each row of the data set represents a customer’s destination and her purpose for making the trip. For example, for customer 4 (row 5 of the worksheet) the destination is “*Fort Pierce*” and the purpose is “*Meeting*”.
   4. **Objective:** The marketing team would like you to use K-means clustering to segment the data into K=4 clusters in order to send targeted advertising to each rider based on knowledge of her destination. Therefore, each data point for clustering is one of the 189 destinations in the 11-dimensional feature space of purposes (including “*blank”*).
   5. **Clustering Process:** The data set provides information regarding 1155 riders going to 189 destinations: therefore, in general, there are multiple riders corresponding to each destination. Therefore, for each destination, you need to use a pivot table to find the total number of customers going to each destination. You should end up with a matrix (in Excel) with 189 columns (destinations) and 11 rows (the purposes corresponding to each destination).
      1. Segment the resulting 189 destinations (data points) into K=4 clusters in the 11-dimensional space of “purposes”.
      2. Determine the center of each cluster and identify the dominant (key) purposes for each center.
      3. Explain how the marketing group will use this information to target ads for each cluster.
      4. Determine the cluster and target ads for the following destinations: Berkeley, Cary, Midtown, Whitebridge, Santa Clara.
   6. **Deliverables:** Your solution must also include:
      1. Screenshots of the spreadsheet in which you calculate the location of each centroid.
      2. Screenshots of the spreadsheet in which you assign clusters based on the distance from each destination data point to the cluster centers.
      3. (Very specific) Conclusions and Recommendations based on your results.
2. **Create a Plan:**
   1. **Step 1:** Segment the 189 destinations (data points) in to K=4 clusters in the 11-dimensional space of “purposes”.
   2. **Step 2:** Determine the center of each cluster and identify the dominant (key) purposes for each center.
   3. **Step 3:** Explain how the marketing group will use this information to target ads for each cluster.
   4. **Step 4:** Determine the cluster and target ads for the following destinations: Berkeley, Cary, Midtown, Whitebridge, Santa Clara.
3. **Execute:**
   1. **Step 1:** Pivot Table
      1. We begin by creating a pivot table. To do this we create a new tab (Pivot) in the Uber\_Ads data set and insert a pivot table with the following data range. This is shown below.

*Figure 2.1: Pivot Table Data Range*



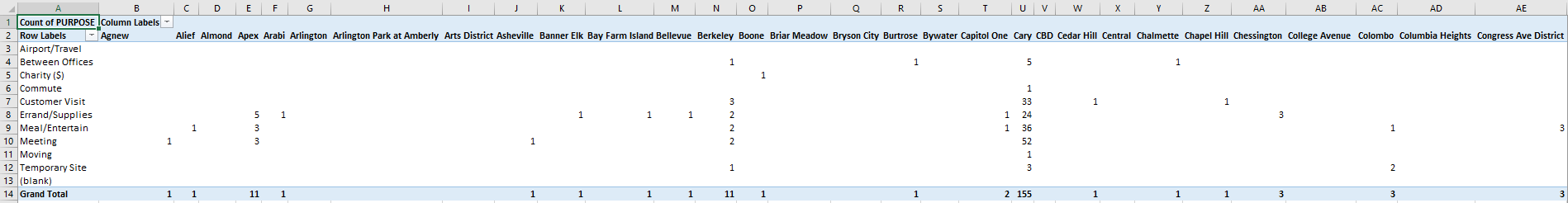
* + 1. This takes the data of the destination and purpose for the 1155 riders. Now we use the following fields to create a pivot table to find the total number of customers going to each destination.:
       1. Rows: Prupose
       2. Columns: Destination
       3. Values: Count of Purpose

*Figure 2.2: Pivot Table Fields*



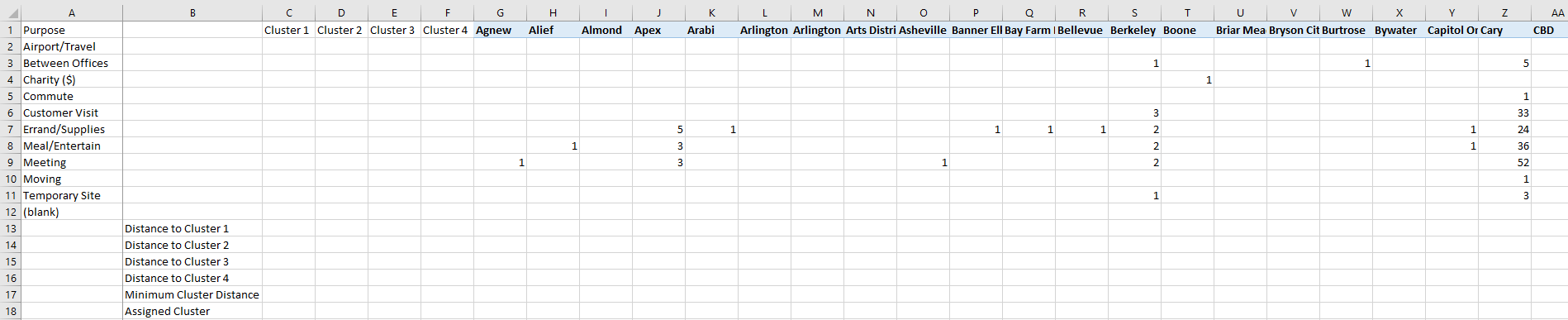
* + 1. The resulting pivot table is shown below.

*Figure 2.3: Pivot Table*



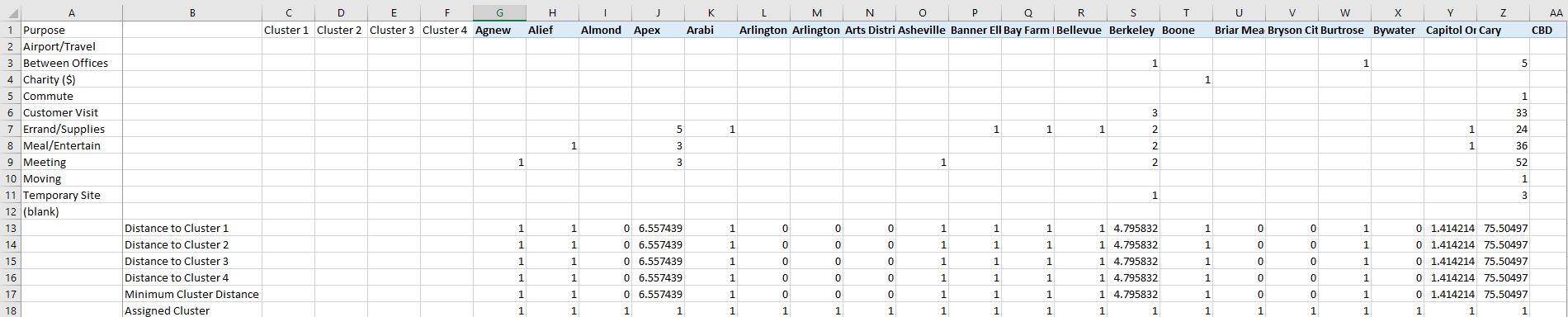
* + 1. Now that we have the final pivot table we can determine the center of each cluster and identify the dominant purposes for each center.
  1. **Step 2:** Cluster Centers and Dominant Purposes
     1. To do this we will create a new tab called 4MC. In column A we add the the 11 purposes for each destination trip. We then add the pivot table data to the columns C through GL. The blank column in B is so we can add the titles for Distance to Cluster, Minimum Cluster Distance, and Assigned Cluster. This gives us the table below.

*Figure 2.4: 4MC Table with Initial Values*



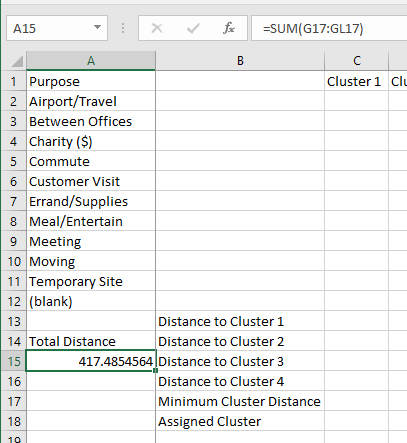
* + 1. Now we will add the following equations for each value we need to find( All equations are dragged down from column G to GL):
       1. Distance to Cluster 1: =SQRT(SUM((G$2:G$12-$C$2:$C$12)^2))
       2. Distance to Cluster 2: =SQRT(SUM((G$2:G$12-$D$2:$D$12)^2))
       3. Distance to Cluster 3: =SQRT(SUM((G$2:G$12-$E$2:$E$12)^2))
       4. Distance to Cluster 4: =SQRT(SUM((G$2:G$12-$F$2:$F$12)^2))
       5. Minimum Cluster Distance: =MIN(G13:G16)
       6. Assigned Cluster: =MATCH(G17,G13:G16,0)
    2. The Resulting table is shown below.

*Figure 2.5: 4MC Table with Equations*



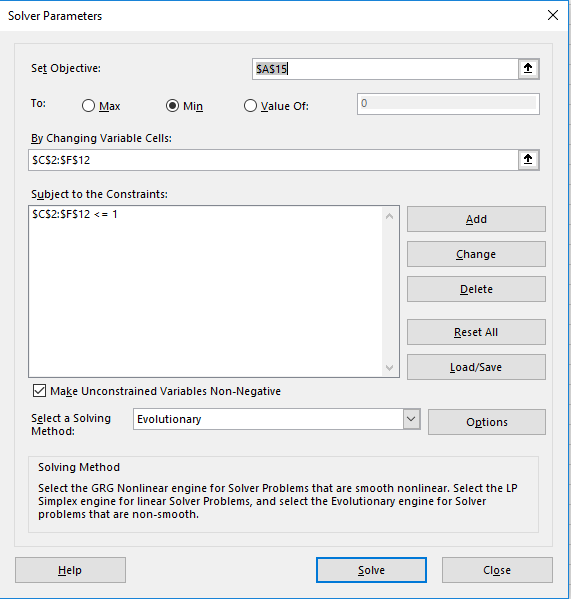
* + 1. Now we will calculate the total distance with the following equation: =SUM(G17:GL17). This results in the value shown below.

*Figure 2.6: 4MC Total Distance*

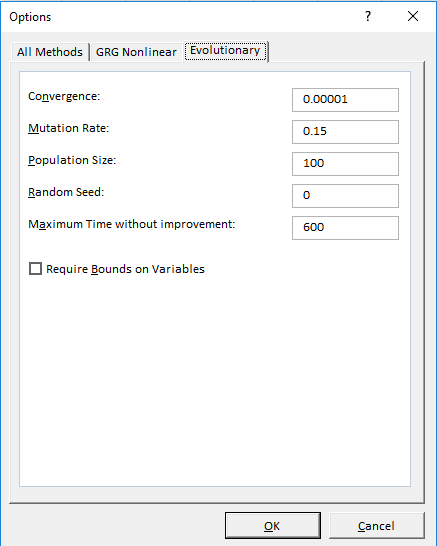


* + 1. Now we will use solver to minimize the distance. To do this we will have the following fields in the solver show below.

*Figure 2.7: Solver Parameters*

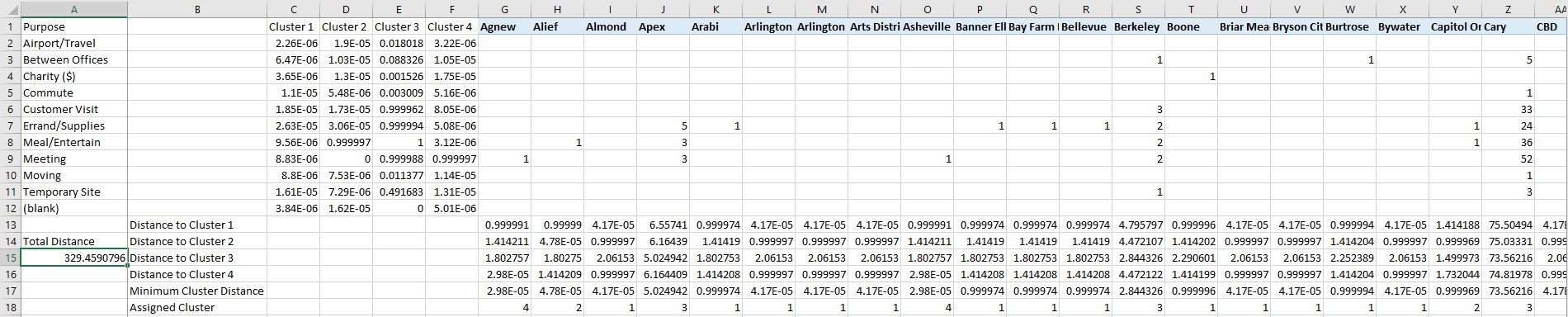


*Figure 2.8: Solver Evolutionary Options*

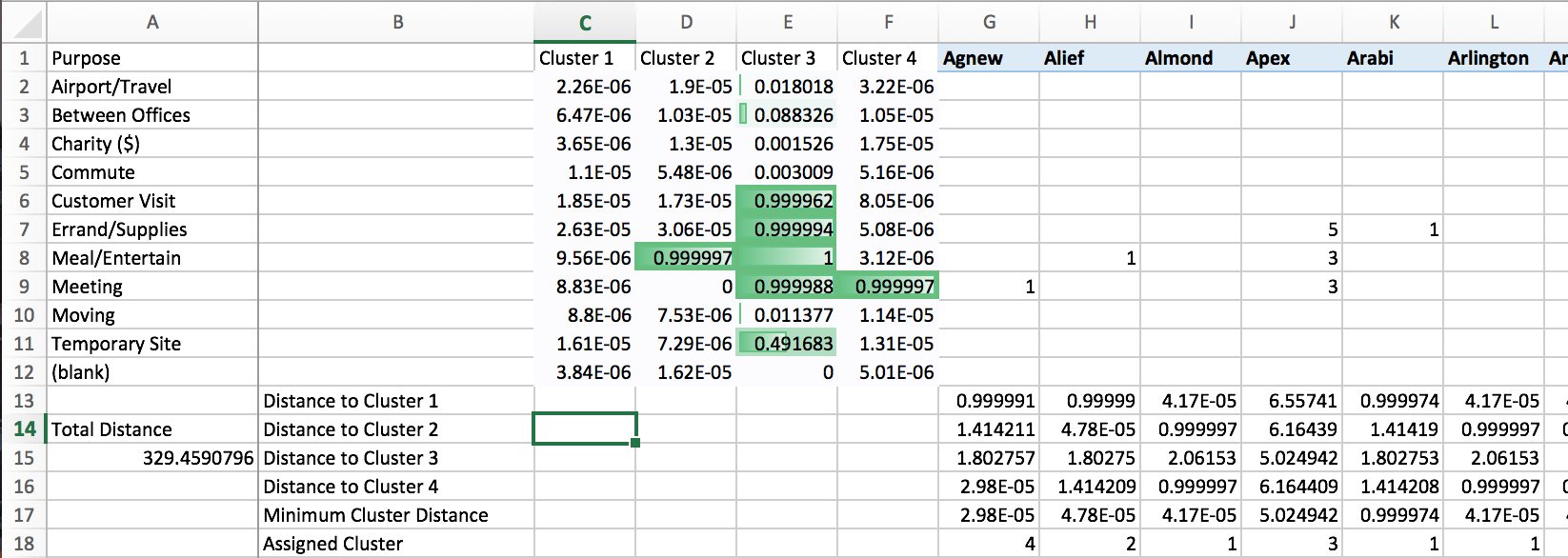


* + 1. After entering the above tables and clicking solve we end with the total distance of 329.46. This is shown below.

*Figure 2.9: 4MC Final Table*

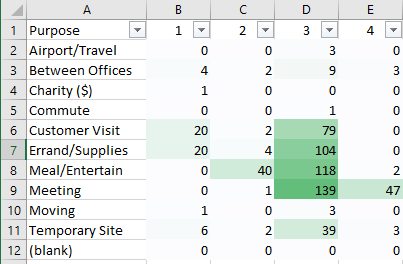


* + 1. We can apply conditional formatting to figure out what the centroids are for each cluster. The result can be seen below.



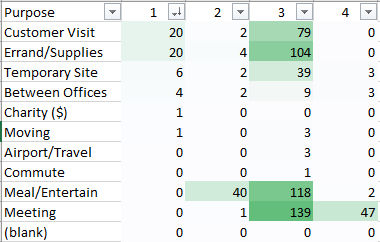
* + 1. We can see that for cluster 1 there is no obvious center. However, the greatest value is 2.63E-05 which pertains to Errand/Supplies. For cluster 2 we can see that the greatest value is 0.999997 which is Meal/Entertain. For cluster 3 the greatest value is 1 which is also Meal/Entertain. However, there are several other values that are practically valued as 1. This includes customer visits, errand/supplies, meeting and temporary site. Lastly, cluster 4’s greatest value is 0.999997 which is Meeting. (Greatest Value pertains to the center of each cluster).
    2. Now that we have the assigned clusters for each destination we can create the final table for determining who is assigned to each cluster and which purpose they’re going for. We can do this by using the equations for each cluster column.
       1. Cluster 1: =SUMIF('4MC'!$G$18:$GL$18,'4MC-Final'!B$1,'4MC'!$G2:$GL2)
       2. Cluster 2: =SUMIF('4MC'!$G$18:$GL$18,'4MC-Final'!C$1,'4MC'!$G2:$GL2)
       3. Cluster 3: =SUMIF('4MC'!$G$18:$GL$18,'4MC-Final'!D$1,'4MC'!$G2:$GL2)
       4. Cluster 4: =SUMIF('4MC'!$G$18:$GL$18,'4MC-Final'!E$1,'4MC'!$G2:$GL2)
    3. After adding these equations we get the results below.

*Figure 2.10: 4MC Cluster Assignments/Purposes*



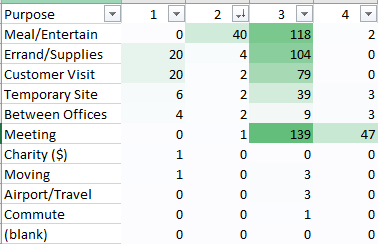
* + 1. Now we can sort each cluster in descending order to see the trends. This is shown below.

*Figure 2.11: Cluster 1*



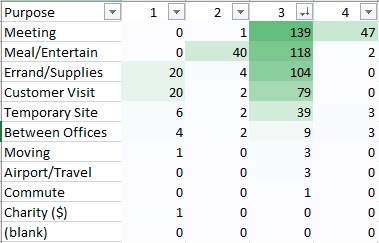
* + 1. From the figure above we can see that cluster 1’s main purposes are Customer Visits and Errand/Supplies. The marketing group can use this information to send ads relating to these 2 purposes.

*Figure 2.12: Cluster 2*



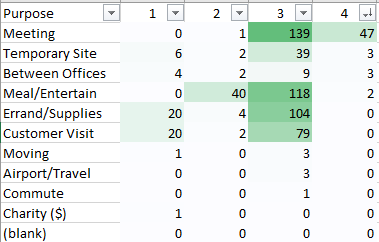
* + 1. From the figure above we can see that cluster 2’s main purpose is Meal/Entertain. The marketing group can use this information to send ads relating to this purpose.

*Figure 2.13: Cluster 3*



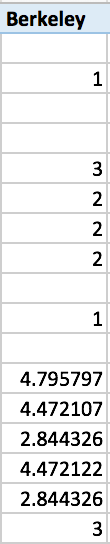
* + 1. From the figure above we can see that cluster 3’s main purposes are Meeting, Meal/Entertain, Errand/Supplies, Customer Visit, and Temporary Site. The marketing group can use this information to send ads relating to these purposes focusing mainly on the greater valued purposes.

*Figure 2.14: Cluster 4*



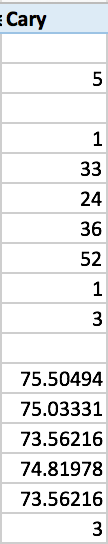
* + 1. From the figure above we can see that cluster 4’s main purpose is Meeting. The marketing group can use this information to send ads relating to this purpose.
    2. Lastly, we will look at the assigned clusters and target ads for the following destinations.
       1. Berkeley

*Figure 2.15: Berkeley*

****

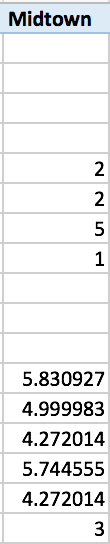
* + - * 1. We can see that Berkeley was placed into cluster 3 which targets Meeting, Meal/Entertain, Errand/Supplies, Customer Visit, and Temporary Site.
      1. Cary

*Figure 2.16: Cary*

****

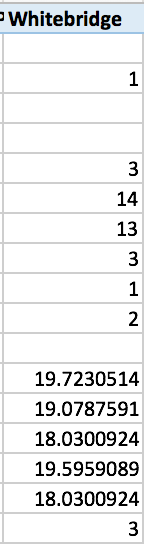
* + - * 1. We can see that Cary was placed into cluster 3 which targets Meeting, Meal/Entertain, Errand/Supplies, Customer Visit, and Temporary Site.
      1. Midtown

*Figure 2.17: Midtown*

****

* + - * 1. We can see that Midtown was placed into cluster 3 which targets Meeting, Meal/Entertain, Errand/Supplies, Customer Visit, and Temporary Site.
      1. WhiteBridge

*Figure 2.18: WhiteBridge*

****

* + - * 1. We can see that WhiteBridge was placed into cluster 3 which targets Meeting, Meal/Entertain, Errand/Supplies, Customer Visit, and Temporary Site.
      1. Santa Clara

*Figure 2.19: Santa Clara*

****

* + - * 1. We can see that Santa Clara was placed into cluster 1 which focuses on Errnad/Supplies and Customer Visit.

1. **Draw Conclusions:**
   1. Overall, I didn’t fine this problem to be too challenging. I found out the recommendations for each cluster which is shown earlier in the problem. Uber will be able to use these four clusters to determine what they best ads are for their customers. The only problem was that cluster 3 had several purposes which may make it difficult to narrow down the ones that correlate to each person.

**Problem 3: Uber Surge Pricing**

1. **Define the Problem:**
   1. Surge Pricing is a dynamic price scheme used by Uber to price rides based on time of the day. To facilitate surge pricing, the Uber data analytics team has been charged with creating a learning model, based on linear regression, which relates three key input parameters or features - total distance per trip(miles), total time per trip (minutes), and surge factor - to the output parameter. The total price per trip. The total price per trip is given by
   2. Where ,, are respectively, the distance/trip, time/trip, and surge factor. The coefficients of the linear regression model are:
      1. = Uber base charging fee ($)
      2. = Price per mile ($/mile)
      3. = Price per minute($/min)
      4. = Base price ($)
   3. **Training:** The given workbook “*Uber\_Rides*” is collection of Uber rides that contains 3 spreadsheets. The first sheet “Rides” contains the following information **for 1135 rides**:
      1. Start time of each ride
      2. Distance of each trip (miles)
      3. Time of each trip (minutes)
      4. Total price of each ride (dollars)
   4. The second sheet “surge” contains:
      1. Start time for each surge factor
      2. Surge factor
   5. While the “rides” spreadsheet contains the parameters ,, and the output y, it does not contain the surge factor . The surge factor for each trip needs to be computed from the “surge” spreadsheet based on the start time of the trip. For example, for the following start time of 3:08:00 PM, the table lookup from “Surge” spreadsheet gives a value of 1.25

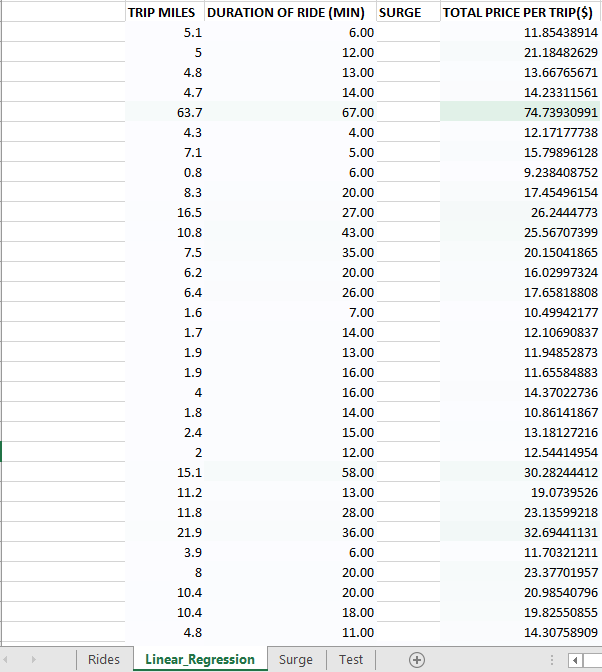
|  |  |
| --- | --- |
| Start time | Surge |
| 3:08:00PM | 1.25 |

 [hint: Use excel feature ‘lookup’]

* 1. Create a new sheet called “*Linear\_Regression*”, which contains the three inputs (features) and the output y as the data set of training examples for creating your linear regression model. Use the training example to determine the model coefficients ,,, and to minimize the sum of the square errors (MSS).
  2. Write down the equation for linear regression with the numerical values of the four coefficients
  3. [caution: for the linear regression problem in the book “Data Smart” the author set constraints on the model coefficients as “-1 <= coefficient <= 1 “ in “Solver”. Note that, in this problem, where we are performing the standard version of linear regression, these constraints are not applicable.]
  4. **Testing:** The third sheet “test” contains the following information for 20 rides:
     1. Start time of each ride
     2. Total number of miles of each ride
     3. Total time of each ride
  5. Run your linear regression model, obtained from the training set, on the test data in the worksheet “test” to determine the total price for each trip in the test set. Make sure you check your results.
  6. **Deliverables:** Your solution must also include:
     1. Mathematical model of the linear regression technique used
     2. Screenshots of the spreadsheets used for calculation of the model coefficients
     3. Screenshot of the results of computing total cost for the test data
     4. Conclusions based on your results

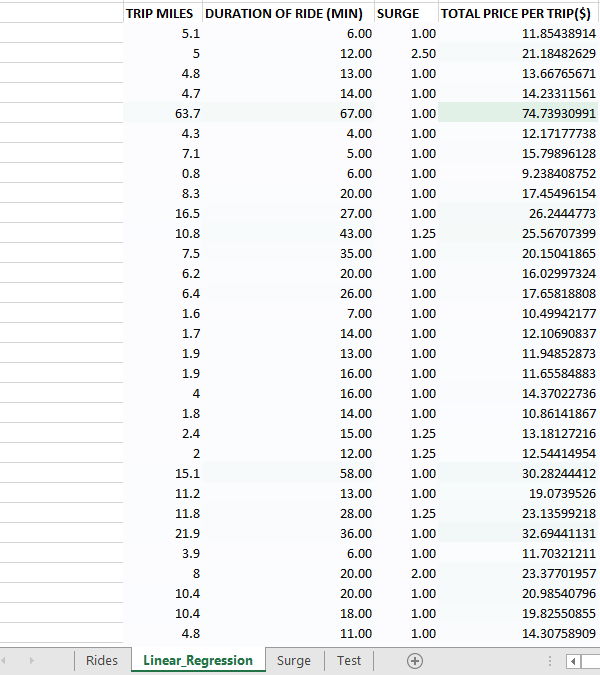
1. **Create a Plan:**
   1. **Step 1:** Create a new sheet called “*Linear\_Regression*”. Add the features and the output y as the data set of training examples for creating the linear regression model.
   2. **Step 2:** Determine the model coefficients ,,, and to minimize the sum of the square errors (MSS) using the training examples.
   3. **Step 3:** Write down the equation for linear regression with the numerical values of the four coefficients.
   4. **Step 4:** Run the linear regression model, obtained from the training set, on the test data in the worksheet “Test” to determine the total price for each trip in the test set.
   5. **Step 5:** Check the results
2. **Execute:**
   1. **Step 1:** Linear Regression Model
      1. To create our linear regression model we need to first create a new tab. We will call it Linear\_Regression. In this tab we will copy over the three features and the output y as the data set of training examples. The table is shown below.

*Figure 3.1: Linear\_Regession Feature Input*



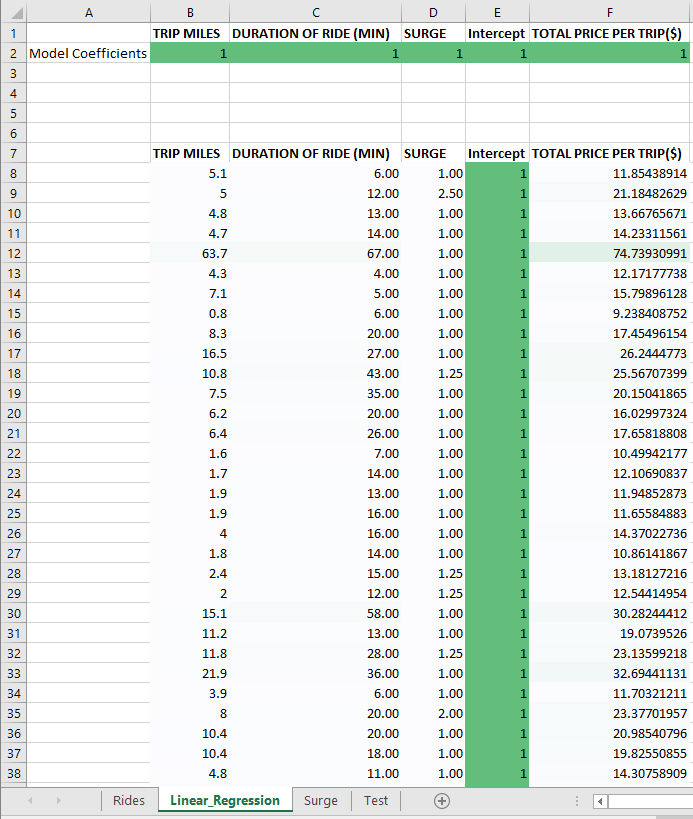
* + 1. Notice that the surge values do not appear. In order to find the surge values that correspond to each trip we need to use the VLOOKUP command. The equation goes as follows: =VLOOKUP(Rides!A2, Surge!$A$2:$B$11, 2, TRUE). After adding this equation we can now see the surge values as shown below.

*Figure 3.2: Linear\_Regression Surge Values*



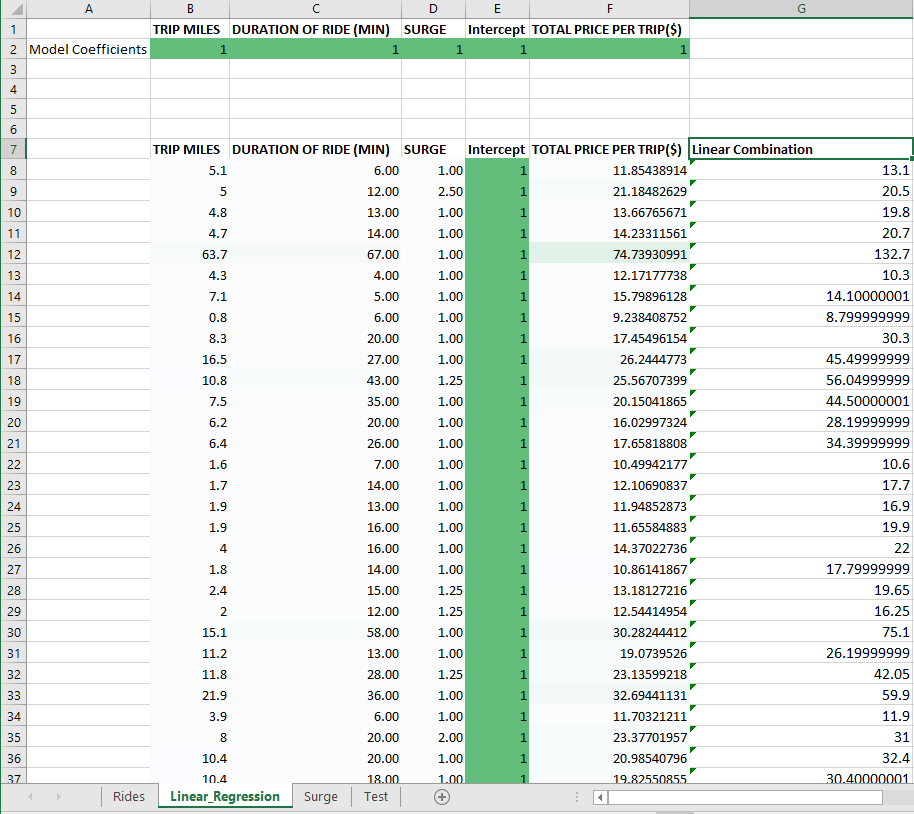
* 1. **Step 2:** Model Coefficients and Minimization of SSE
     1. Now we will add an intercept column between column D and E for a baseline. To incorporate the intercept into the model easier, we will fill the intercept column with 1s. This will allow us to evaluate the model by taking the SUMPRODUCT of the coefficient row with a data row that will incorporate the intercept value. We will also add the all of the coefficients for this model into row 2 and place a starting value of 1 in each cell. We will also apply conditional formatting. The result is shown below.

*Figure 3.3: Linear\_Regression Intercept/Model Coefficients*



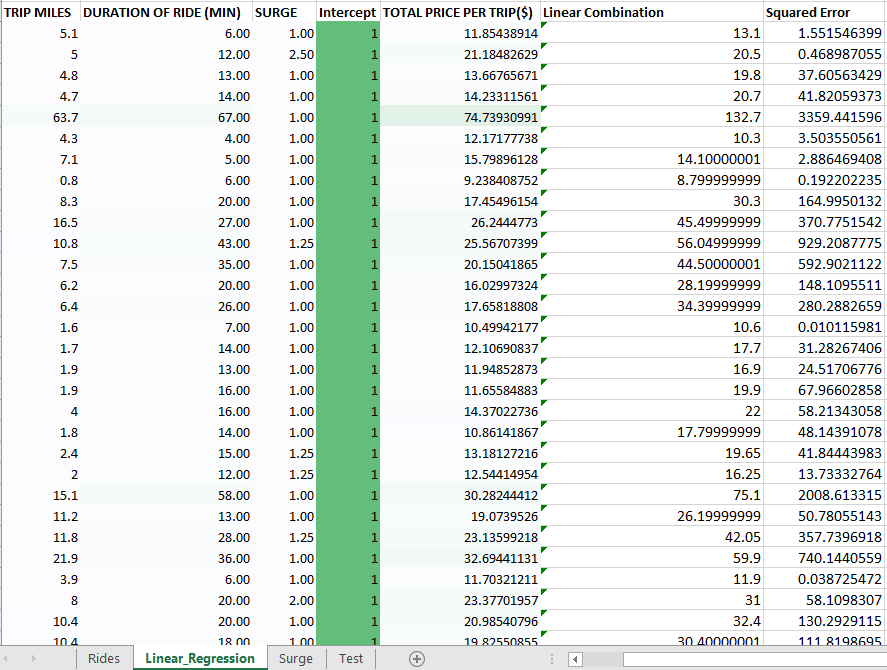
* + 1. Now we will take the linear combination of the coefficients with a row of ctrip data and get a total price prediction. The first step is to add a column to the spreadsheet with a prediction on one of the rows of data. So in column G we add the label Linear Combination to row 7 and the equation: =SUMPRODUCT(B$2:E$2,B8:E8). This gives us the table below.

*Figure 3.4: Linear\_Regression Linear Combination*



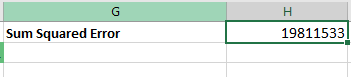
* + 1. As we inspect the values we can see the predictions are off. This is because we used coefficients of 1 for every variable. Now we need to get the computer to set the model coefficients for us. To do this we need to add an error calculation in column H. We will use the square error. So for the first trip we have the equation =(F8-G8)^2. Dragging this equation down gives us the rest of the values as shown below.

*Figure 3.5: Linear\_Regression Squared Error*



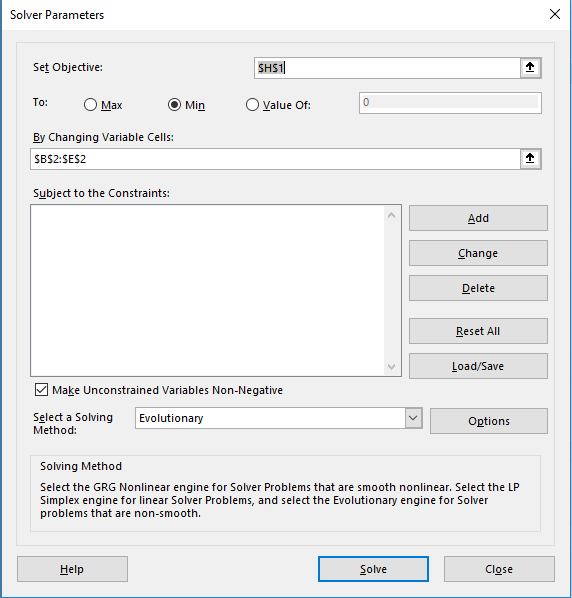
* + 1. We then add a cell above the predictions in cell H1 with the equation =SUM(H8:H1142). This gives us the SSE. The result is shown below.

*Figure 3.6: Linear\_Regression SSE*

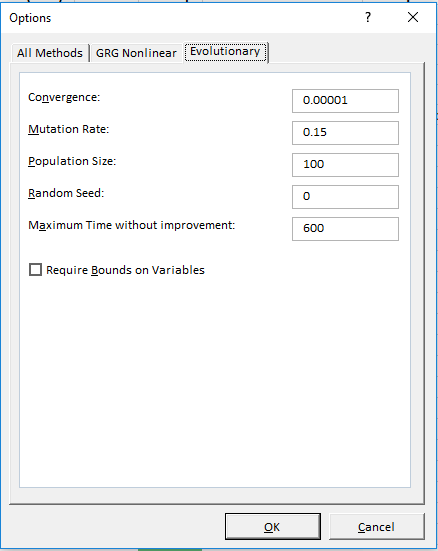


* + 1. Now we will train with the linear model. We will do this by setting the coefficients for each variable such that the sum of the squared error is as low as it can be. To do this we will use the solver and set it up as shown below.

*Figure 3.7: Linear\_Regression Solver Parameters*

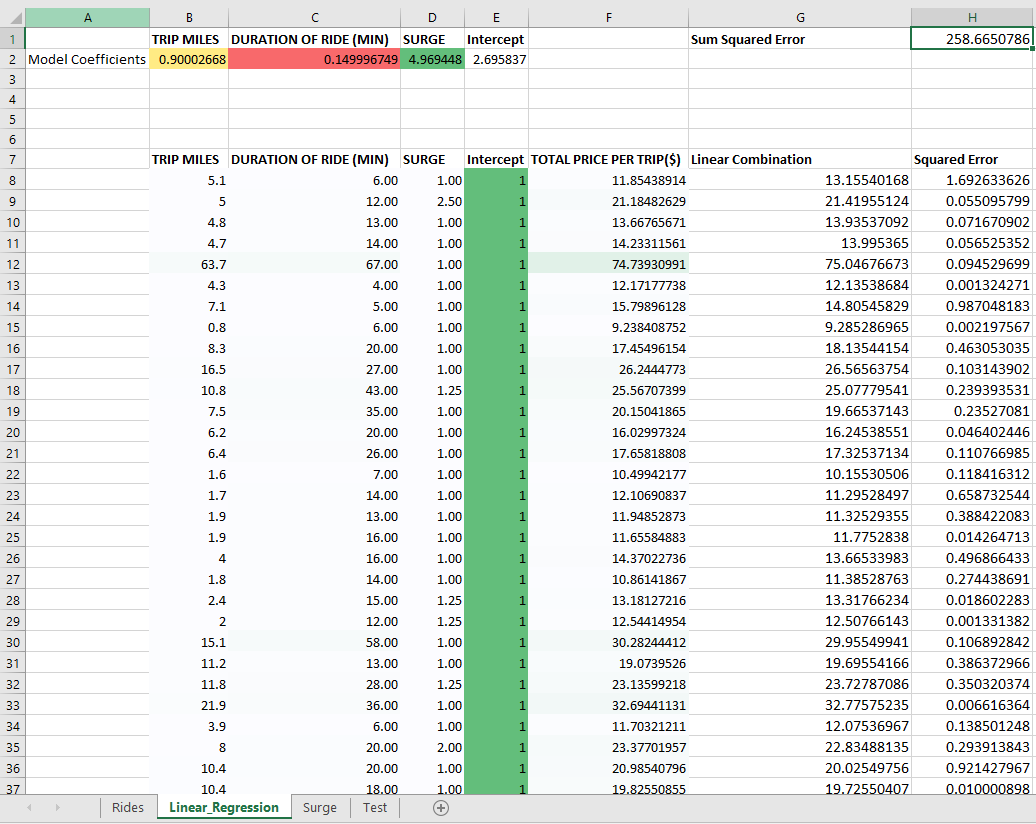


*Figure 3.8: Linear\_Regression Solver Options*



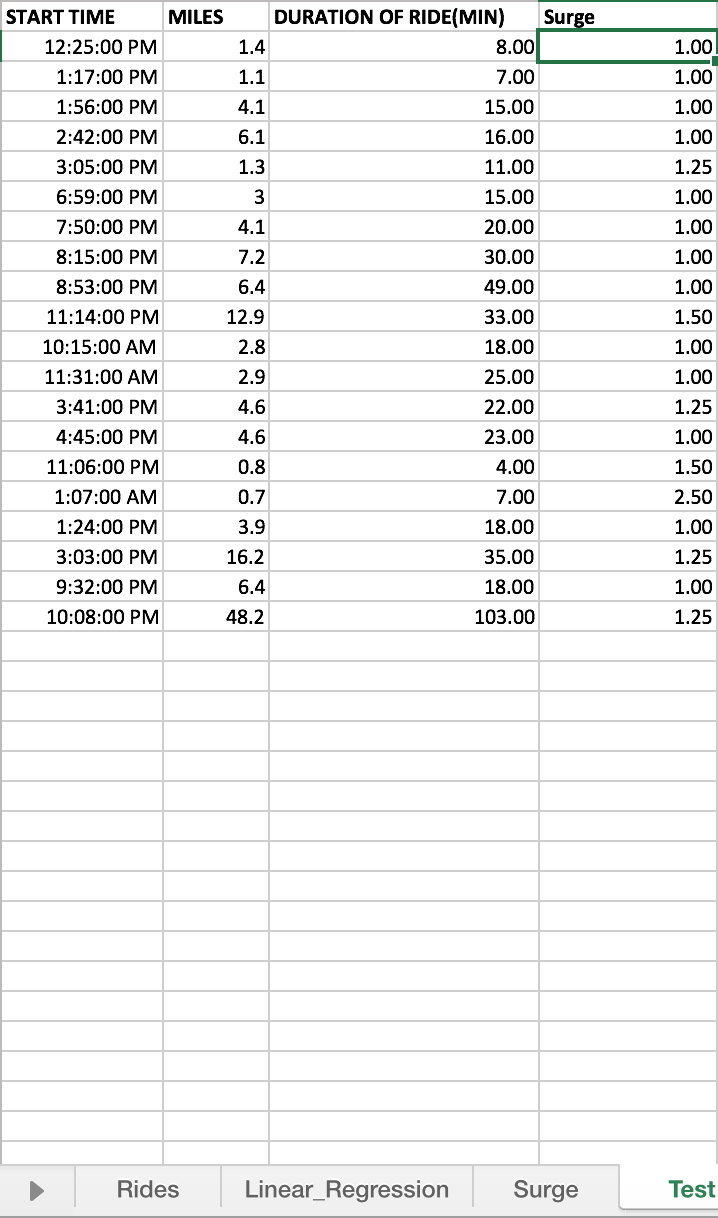
* + 1. After running the solver we get the results below.

*Figure 3.9: Linear\_Regression Solver Results*



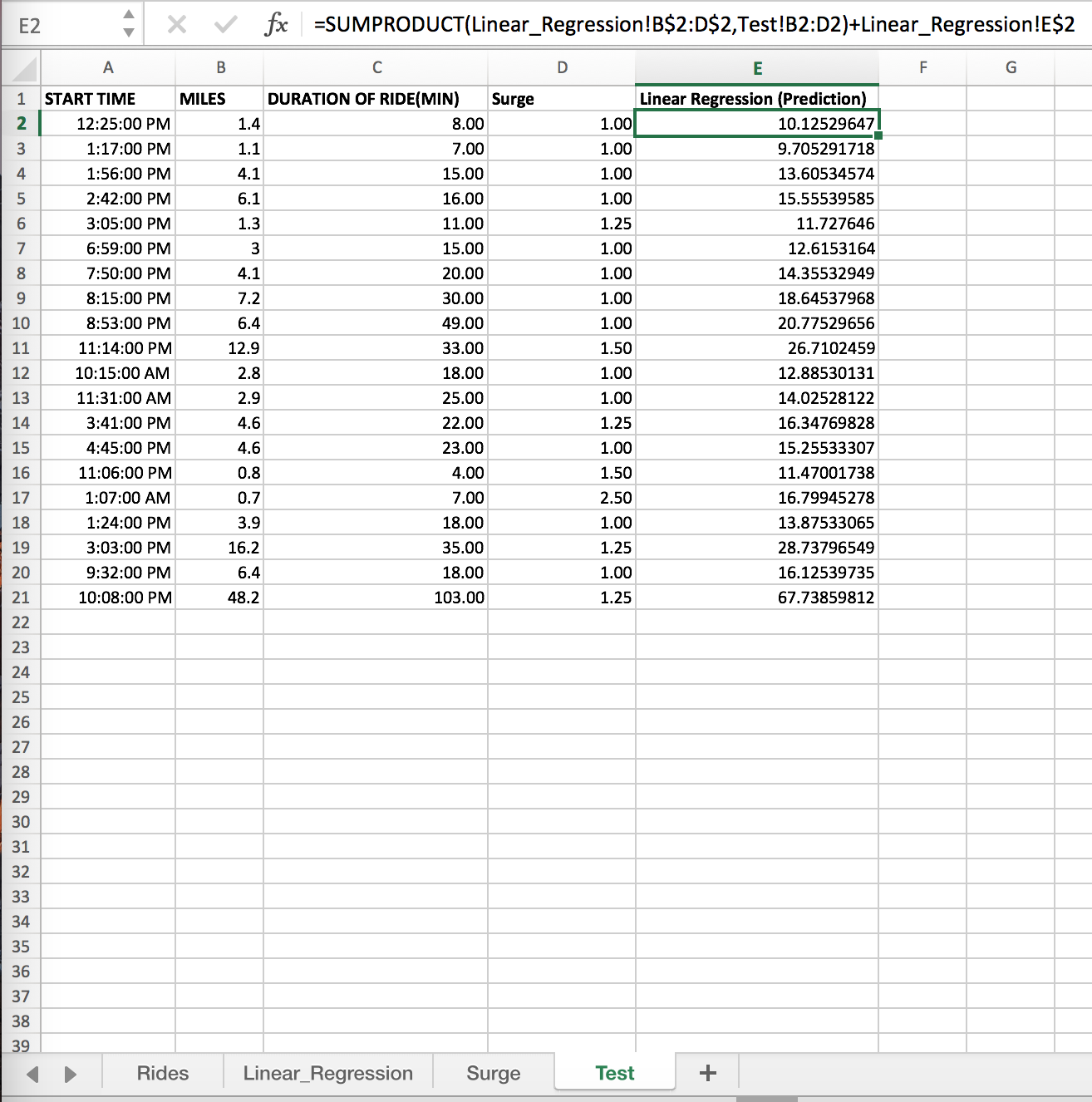
* + 1. As we can see the solver finished with a 258.665 sum of squared error. The model coefficients are also shown in row 2 columns B-D.
  1. **Step 3:** Linear Regression Equation
     1. The linear equation with the values we solved for in step 2 is as follows:
  2. **Step 4:** Linear Regression on Test
     1. The first step is to create a Surge column on the Test tab next to Duration of Ride. We use the VLOOKUP command again to solve for the surge values of each ride. The equation is as follows: =VLOOKUP(Test!A2, Surge!$A$2:$B$11, 2, TRUE). The result is shown below.

*Figure 3.10: Test Surge Values*

****

* + 1. Now that we have all of the surge values we can solve for the linear prediction of each. The equation we use for this is as follows: =SUMPRODUCT(Linear\_Regression!B$2:D$2,Test!B2:D2)+Linear\_Regression!E$2. This gives us the figure below.

*Figure 3.11: Test Linear Regression*

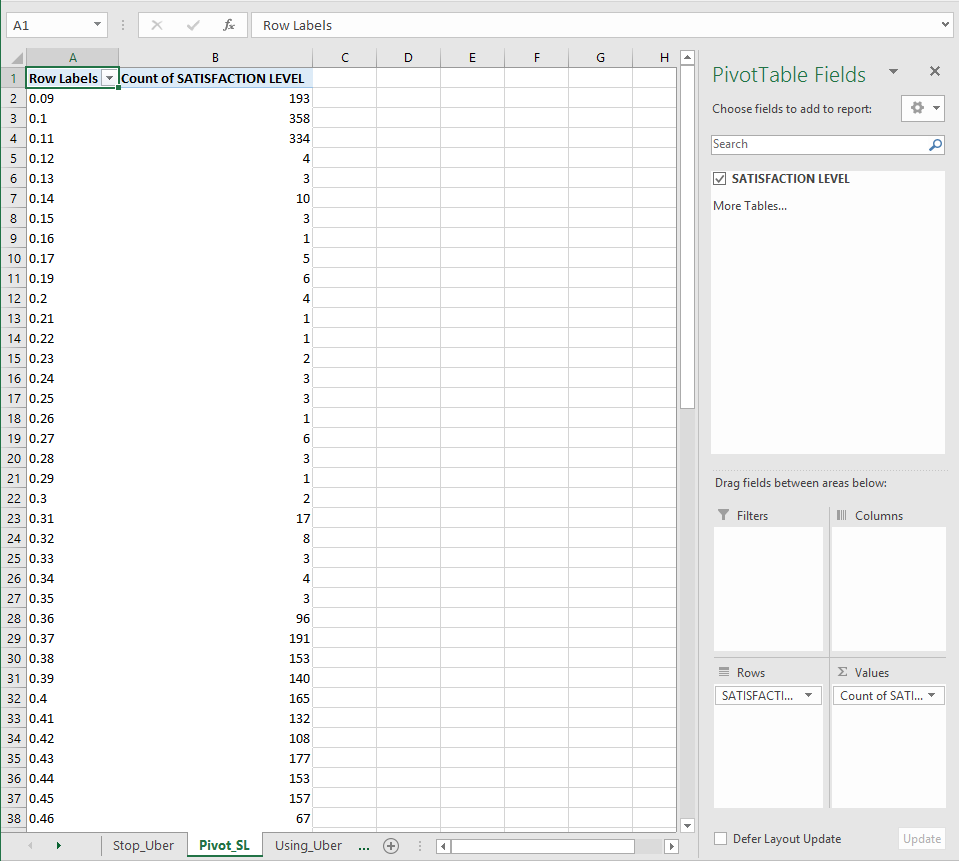
****

1. **Draw Conclusions:**
   1. Overall I found that this problem wasn’t too difficult. My final linear regression equation was . I was able to use this equation to solve for the total trip time for each ride.

**Problem 4: Driver Attention at Uber**

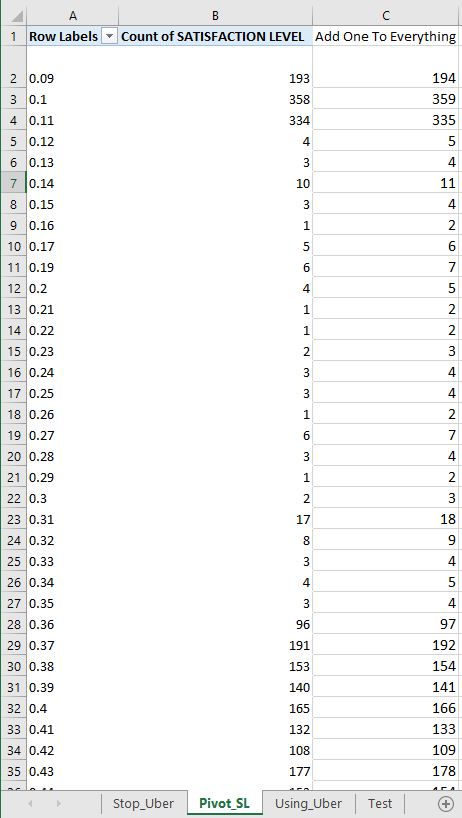
1. **Define the Problem:**
   1. The Uber management team has tasked the data science (DS) group with the problem of predicting whether a driver falls into one of two classes:
      1. Drivers who stopped using Uber, represented by 1.
      2. Drivers who are still using Uber, represented by 0.
   2. The DS group has assigned a training data set of 7100 drivers. Each training example (or row of the spreadsheet) has 3 input features:
      1. Satisfaction level of driver represented as a decimal number between (0,1)
      2. Driver rating represented as a decimal number between (0,1)
      3. Number of incidents represented as an integer between (1,9)
   3. and one output:
      1. Driver class = 1 i.e. stopped using Uber
      2. Driver class = 0 i.e. still using Uber
   4. This training data is available in the workbook “*Uber\_Drivers*”. There are 3 spreadsheets “*Stop\_Uber*”, “*Using­\_Uber*”, and “*Test*”. Each spreadsheet has 4 columns:
      1. Satisfaction Level
      2. Driver Rating
      3. Number of incidents
      4. Driver class
   5. For the spreadsheet “*Stop\_Uber*”, the Driver class (column 4) = 1 and for “*Using\_Uber*” the Driver class (column 4) = 0.
   6. The team decides to use a Naïve Bayes classification model to predict the driver class.
   7. **Training:** To use a Naïve Bayes classifier, we first need to determine the probabilities of each feature , conditional on belonging to class 1 or class 0, i.e. determine , and . Note that each has multiple values.
   8. For example, the feature (satisfaction level) for class 1 has 80 unique values ranging from 0.09 to 0.92. For each value of ,you need to compute the corresponding probability, e.g. for , you need to compute , and similarly for all other 79 values of . For Class 0, has 89 unique values ranging from 0.12 to 1. For each value of belonging to this class, you need to compute the corresponding probability, e.g., for , you need to compute , and similarly for all other 88 values of .
   9. **Testing:** Once you have calculated the probabilities for each unique value of , use this probability table to check how well your model works on classifying the drivers in the “*Test*” spreadsheet as either 1 (stopped using Uber) or 0 (still using Uber).
   10. **Deliverables:** 
       1. Mathematical model of Naïve Bayes model used.
       2. Screenshots of the spreadsheets used for computing the probabilities for each feature.
       3. Screenshot of the classifications of the test data set.
       4. Classifier Accuracy (%) = (Number of Correct Predictions/Total Number of Test Data Points)100
       5. Conclusions and Recommendations based on your results.
2. **Create a Plan:**
   1. **Step 1:** Determine the probabilities of each feature , conditional on belonging to class 1 or class 0, i.e. determine , and .
   2. **Step 2:** Use this probability table to check how well your model works on classifying the drivers in the “*Test*” spreadsheet as either 1 (stopped using Uber) or 0 (still using Uber).
3. **Execute:**
   1. **Step 1:** Feature Probabilities
      1. Feature 1 Class 1
         1. To find the probability we need to first determine how many times the same Satisfaction Level value appears. We can do this by creating a pivot table for Satisfaction Level Feature. In the pivot table fields we will have the following
            1. Rows: Satisfaction Level
            2. Values: Count of Satisfaction
         2. This gives us the table below.

*Figure 4.1: Pivot Table Satisfaction Level*



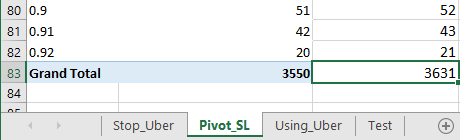
* + - 1. Now that we have the count of each value we can tack on the probabilities to each count value. Before we do this we will apply the additive smoothing concept by adding one to each count value. We will add a separate column called using the equation C2=B2+1. This gives us the table below.

*Figure 4.2: Pivot Table Add One To Everything*



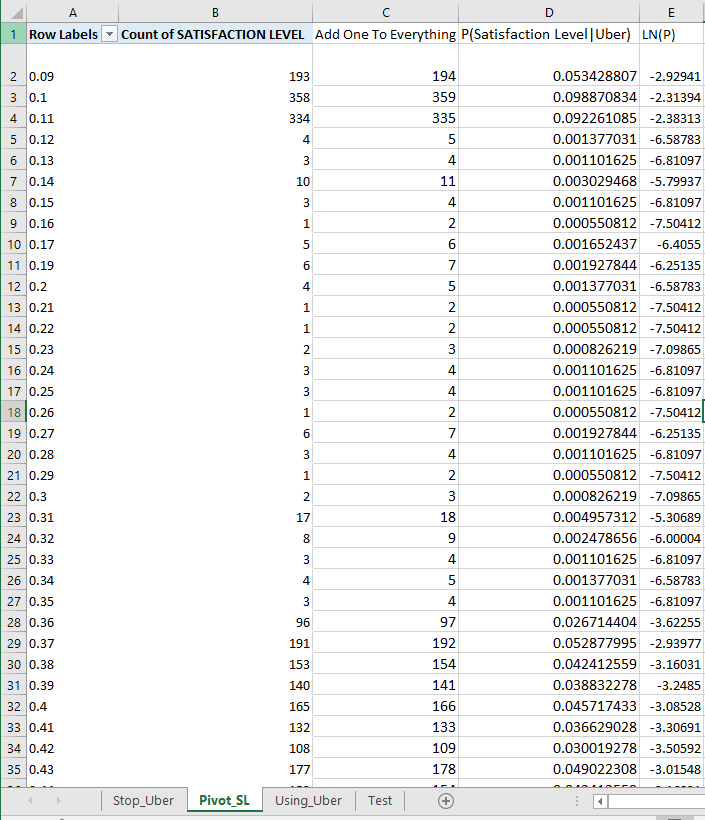
* + - 1. Since 1 has been added to everything we need a new grand total count. We can do this by simply setting the bottom cell sum to the cells above it using the equation =SUM(C2:C82). The result is shown below.

*Figure 4.3: Pivot Table Sum*



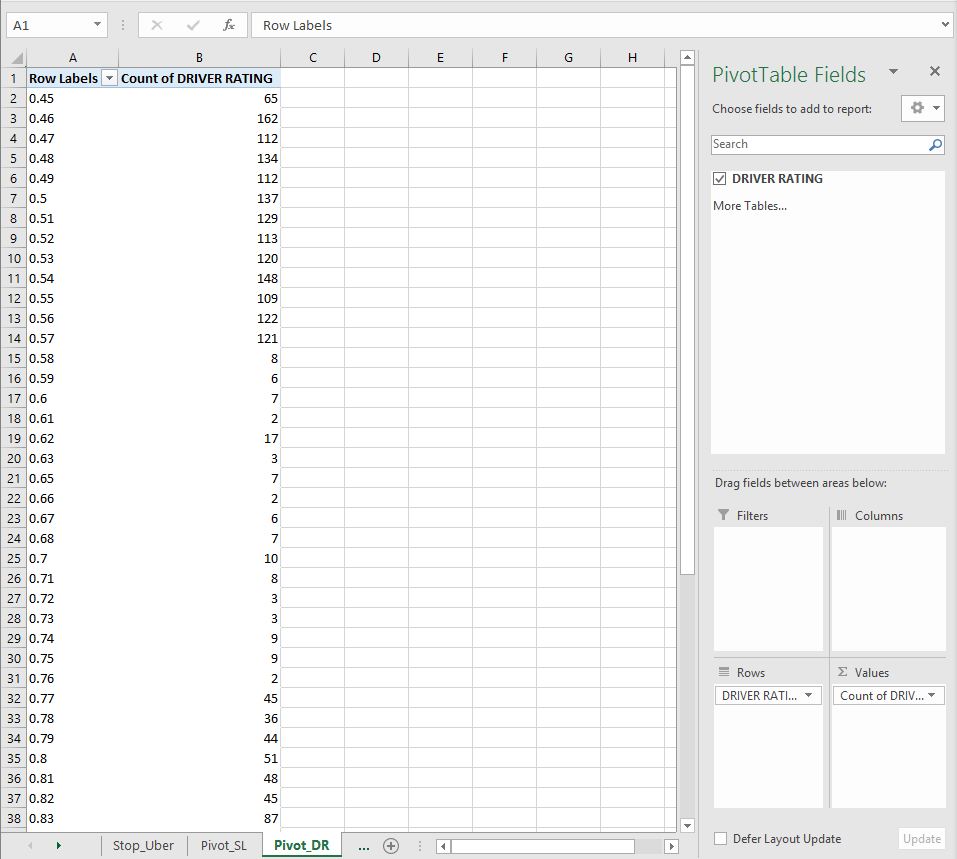
* + - 1. Now in column D we can calculate the probability of each Satisfaction Level as its count in column C divided by the total Satisfaction Level count. We will label D as P(Satisfaction Level|Uber) and insert the equation =C2/$C$83 in D2. We will also add the natural log of the probability in column E using the equation =LN(D2). This gives us the final table below with the end probabilities.

*Figure 4.4: Pivot Table Probabilities*



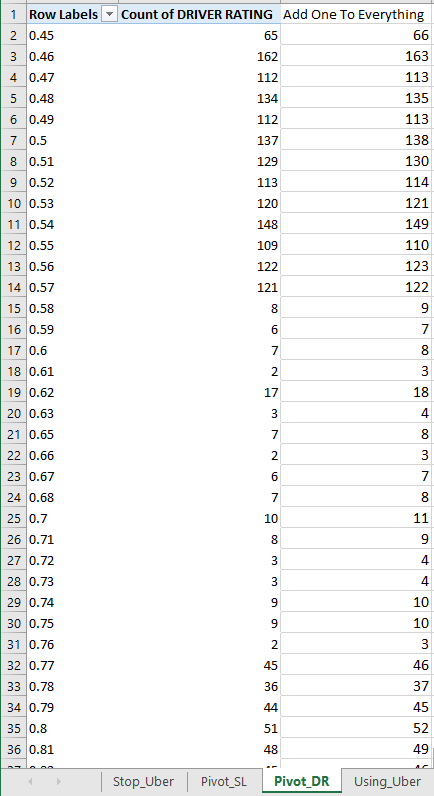
* + 1. Feature 2 Class 1
       1. To find the probability we need to first determine how many times the same Driver Rating value appears. We can do this by creating a pivot table for Driver Rating Feature. In the pivot table fields we will have the following
          1. Rows: Driver Rating
          2. Values: Count of Driver Rating
       2. This gives us the table below.

*Figure 4.5: Pivot Table Driver Rating*



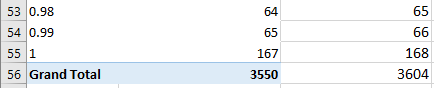
* + - 1. Now that we have the count of each value we can tack on the probabilities to each count value. Before we do this we will apply the additive smoothing concept by adding one to each count value. We will add a separate column called using the equation C2=B2+1. This gives us the table below.

*Figure 4.6: Pivot Table Add One To Everything*



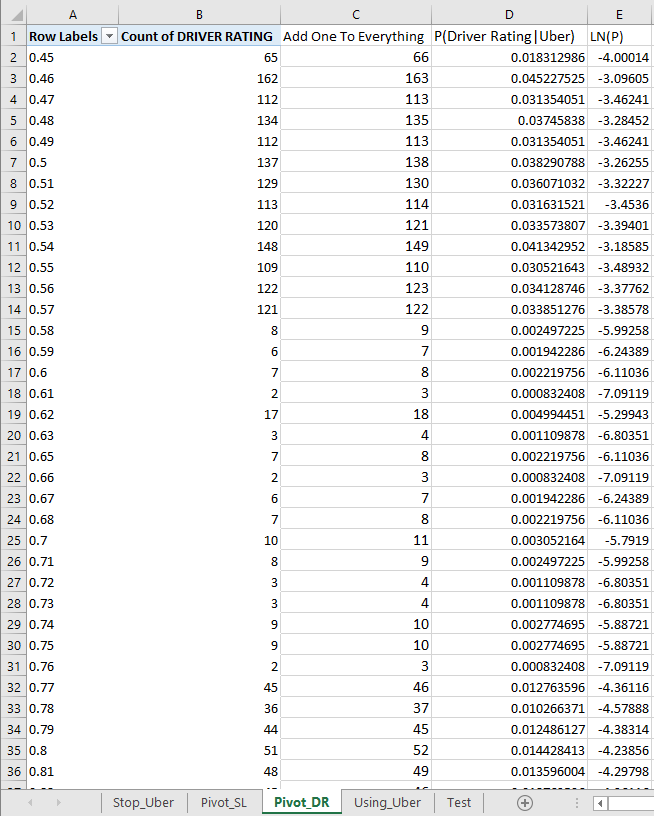
* + - 1. Since 1 has been added to everything we need a new grand total count. We can do this by simply setting the bottom cell sum to the cells above it using the equation =SUM(C2:C55). The result is shown below.

*Figure 4.7: Pivot Table Sum*



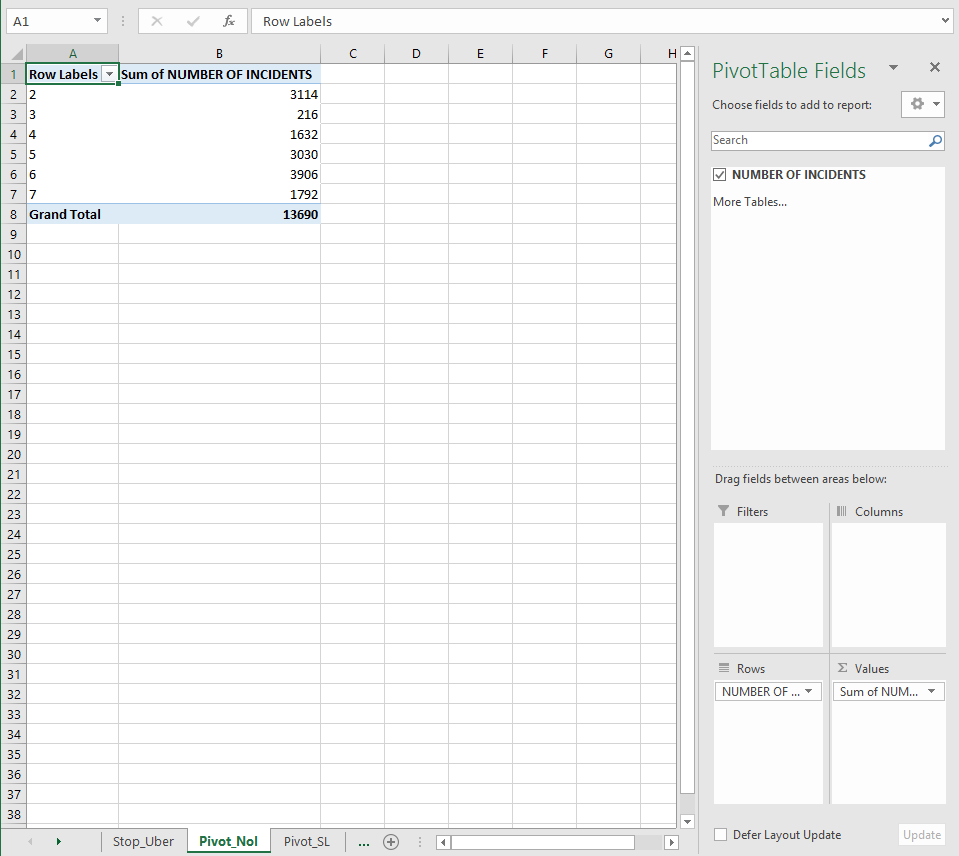
* + - 1. Now in column D we can calculate the probability of each Driver Rating as its count in column C divided by the total Driver Rating count. We will label D as P(Driver Rating|Uber) and insert the equation =C2/$C$56 in D2. We will also add the natural log of the probability in column E using the equation =LN(D2). This gives us the final table below with the end probabilities.

*Figure 4.8: Pivot Table Probabilities*



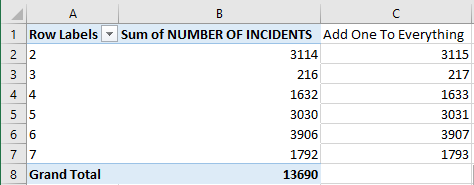
* + 1. Feature 3 Class 1
       1. To find the probability we need to first determine how many times the same Number of Incidents value appears. We can do this by creating a pivot table for Number of Incidents Feature. In the pivot table fields we will have the following
          1. Rows: Number of Incidents
          2. Values: Count of Number of Incidents
       2. This gives us the table below.

*Figure 4.9: Pivot Table Number of Incidents*



* + - 1. Now that we have the count of each value we can tack on the probabilities to each count value. Before we do this we will apply the additive smoothing concept by adding one to each count value. We will add a separate column called using the equation C2=B2+1. This gives us the table below.

*Figure 4.10: Pivot Table Add One To Everything*



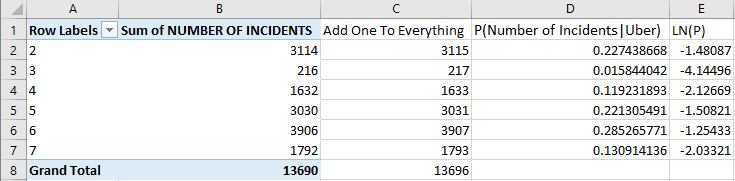
* + - 1. Since 1 has been added to everything we need a new grand total count. We can do this by simply setting the bottom cell sum to the cells above it using the equation =SUM(C2:C7). The result is shown below.

*Figure 4.11: Pivot Table Sum*



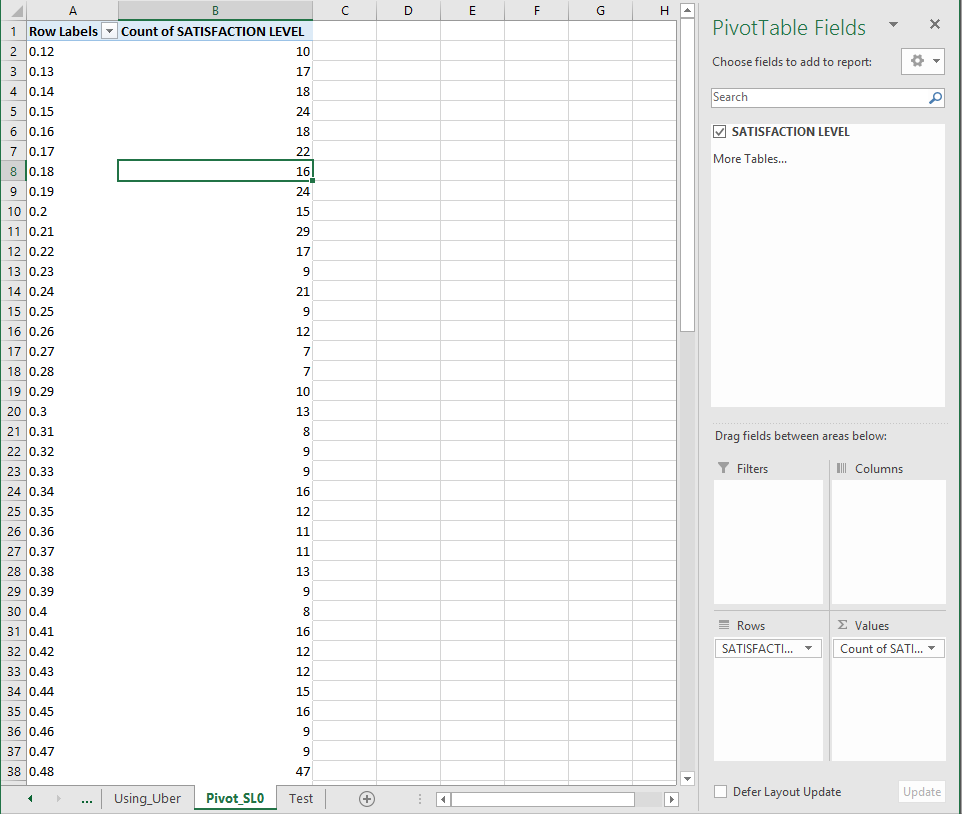
* + - 1. Now in column D we can calculate the probability of each Number of Incidents as its count in column C divided by the total Number of Incidents count. We will label D as P(Number of Incidents|Uber) and insert the equation =C2/$C$8 in D2. We will also add the natural log of the probability in column E using the equation =LN(D2). This gives us the final table below with the end probabilities.

*Figure 4.12: Pivot Table Probabilities*



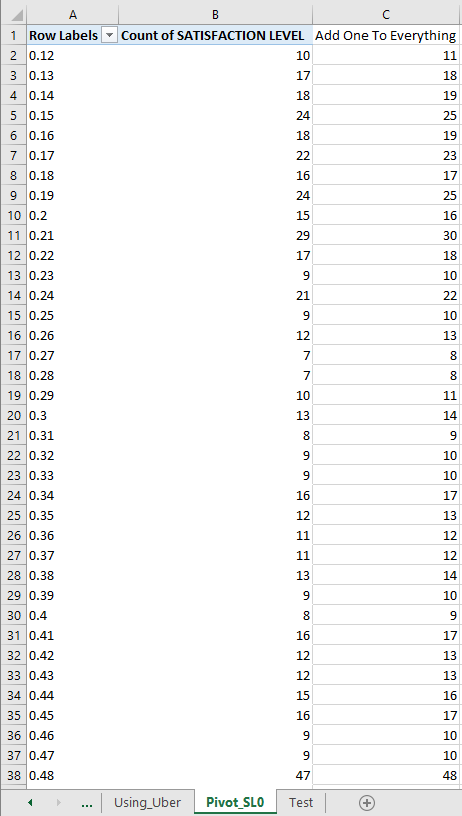
* + 1. Feature 1 Class 0
       1. To find the probability we need to first determine how many times the same Satisfaction Level value appears. We can do this by creating a pivot table for Satisfaction Level Feature. In the pivot table fields we will have the following
          1. Rows: Satisfaction Level
          2. Values: Count of Satisfaction
       2. This gives us the table below.

*Figure 4.13: Pivot Table Satisfaction Level*



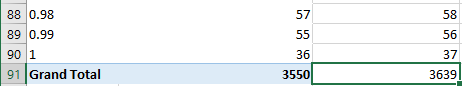
* + - 1. Now that we have the count of each value we can tack on the probabilities to each count value. Before we do this we will apply the additive smoothing concept by adding one to each count value. We will add a separate column called using the equation C2=B2+1. This gives us the table below.

*Figure 4.14: Pivot Table Add One To Everything*



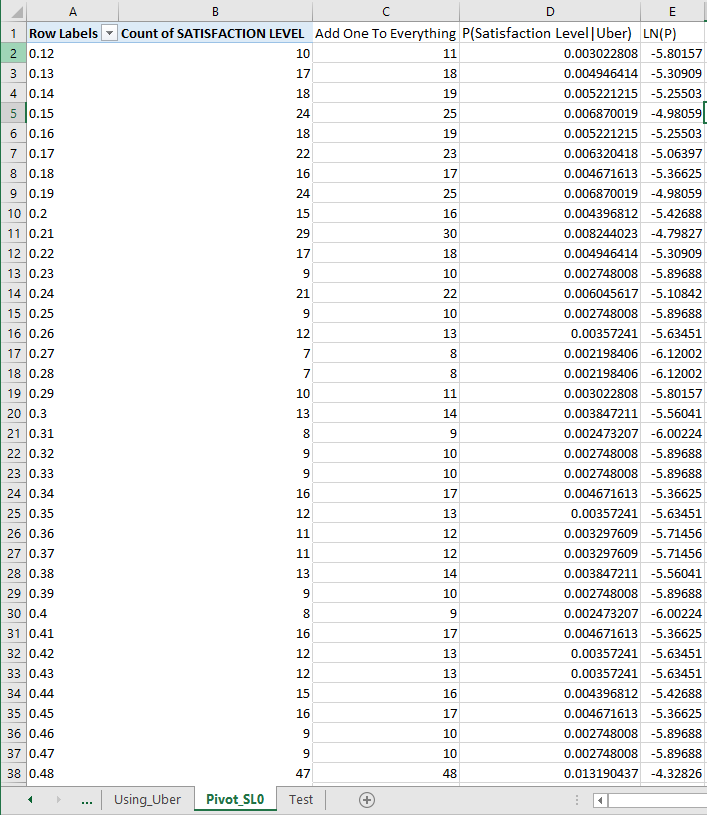
* + - 1. Since 1 has been added to everything we need a new grand total count. We can do this by simply setting the bottom cell sum to the cells above it using the equation =SUM(C2:C90). The result is shown below.

*Figure 4.15: Pivot Table Sum*



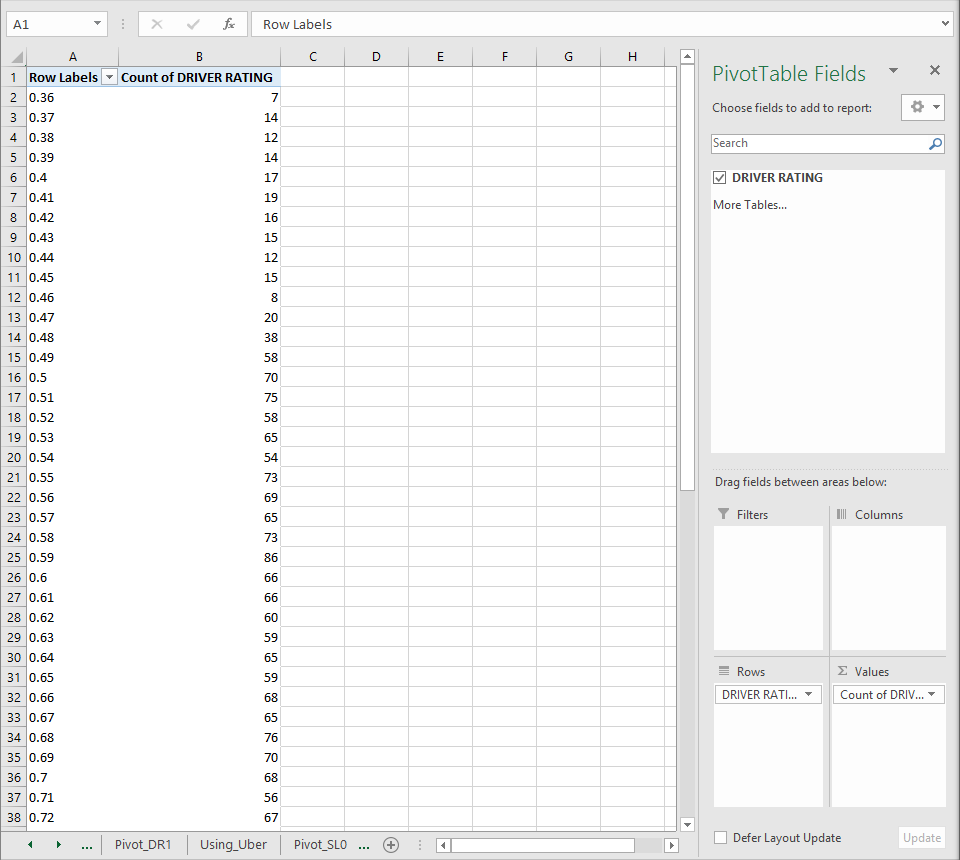
* + - 1. Now in column D we can calculate the probability of each Satisfaction Level as its count in column C divided by the total Satisfaction Level count. We will label D as P(Satisfaction Level|Uber) and insert the equation =C2/$C$91 in D2. We will also add the natural log of the probability in column E using the equation =LN(D2). This gives us the final table below with the end probabilities.

*Figure 4.16: Pivot Table Probabilities*



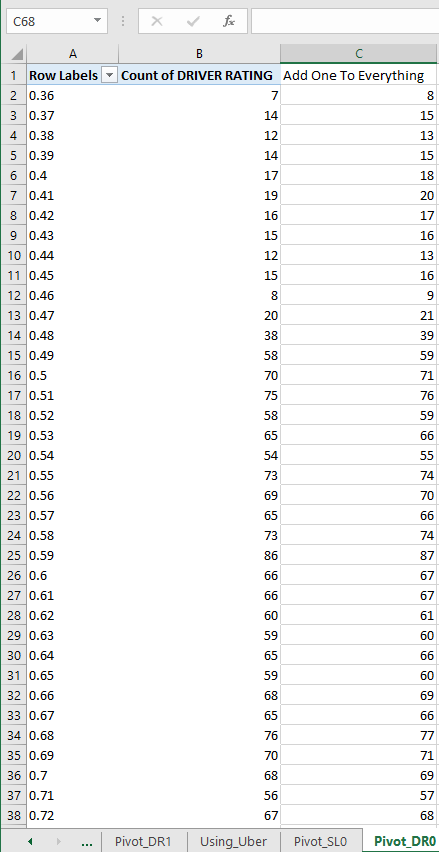
* + 1. Feature 2 Class 0
       1. To find the probability we need to first determine how many times the same Driver Rating value appears. We can do this by creating a pivot table for Driver Rating Feature. In the pivot table fields we will have the following
          1. Rows: Driver Rating
          2. Values: Count of Driver Rating
       2. This gives us the table below.

*Figure 4.17: Pivot Table Driver Rating*



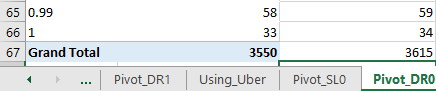
* + - 1. Now that we have the count of each value we can tack on the probabilities to each count value. Before we do this we will apply the additive smoothing concept by adding one to each count value. We will add a separate column called using the equation C2=B2+1. This gives us the table below.

*Figure 4.18: Pivot Table Add One To Everything*



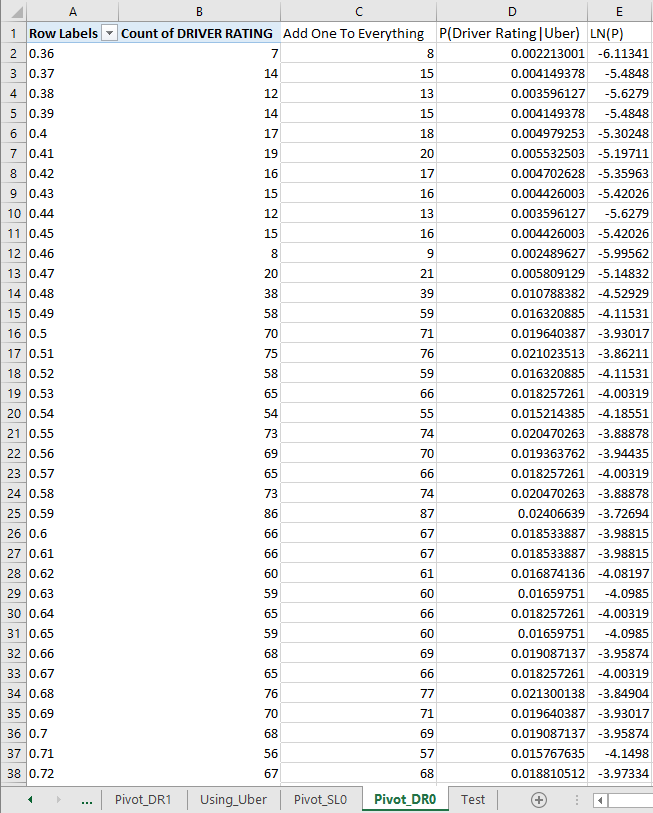
* + - 1. Since 1 has been added to everything we need a new grand total count. We can do this by simply setting the bottom cell sum to the cells above it using the equation =SUM(C2:C66). The result is shown below.

*Figure 4.19: Pivot Table Sum*



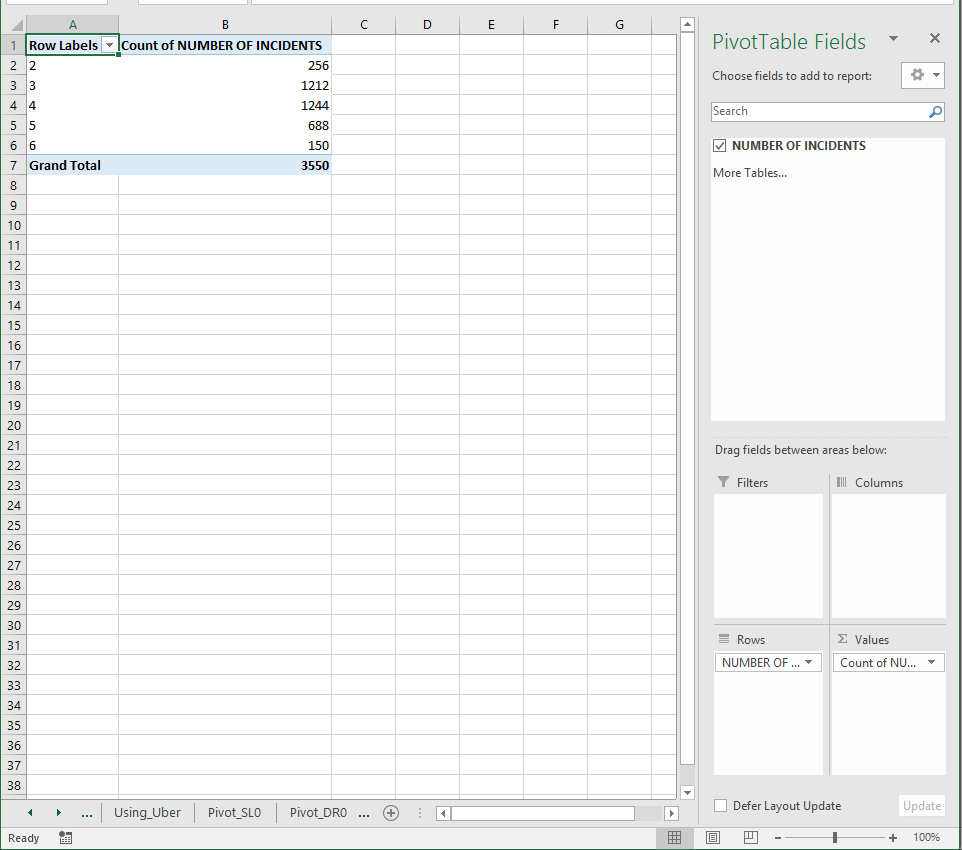
* + - 1. Now in column D we can calculate the probability of each token as its count in column C divided by the total token count. We will label D as P(Driver Rating|Uber) and insert the equation =C2/$C$67 in D2. We will also add the natural log of the probability in column E using the equation =LN(D2). This gives us the final table below with the end probabilities.

*Figure 4.20: Pivot Table Probabilities*



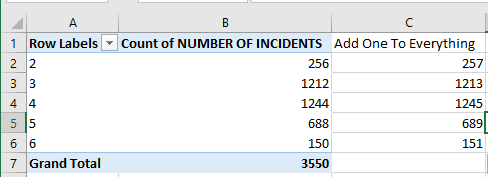
* + 1. Feature 3 Class 0
       1. To find the probability we need to first determine how many times the same Number of Incidents value appears. We can do this by creating a pivot table for Number of Incidents Feature. In the pivot table fields we will have the following
          1. Rows: Number of Incidents
          2. Values: Count of Number of Incidents
       2. This gives us the table below.

*Figure 4.21: Pivot Table Number of Incidents*



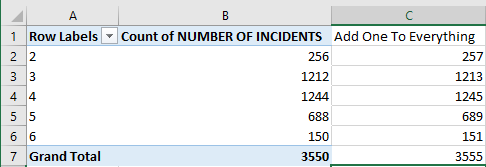
* + - 1. Now that we have the count of each value we can tack on the probabilities to each count value. Before we do this we will apply the additive smoothing concept by adding one to each count value. We will add a separate column called using the equation C2=B2+1. This gives us the table below.

*Figure 4.22: Pivot Table Add One To Everything*



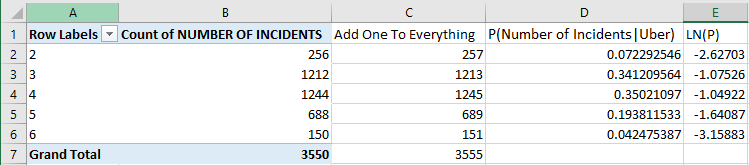
* + - 1. Since 1 has been added to everything we need a new grand total count. We can do this by simply setting the bottom cell sum to the cells above it using the equation =SUM(C2:C6). The result is shown below.

*Figure 4.23: Pivot Table Sum*



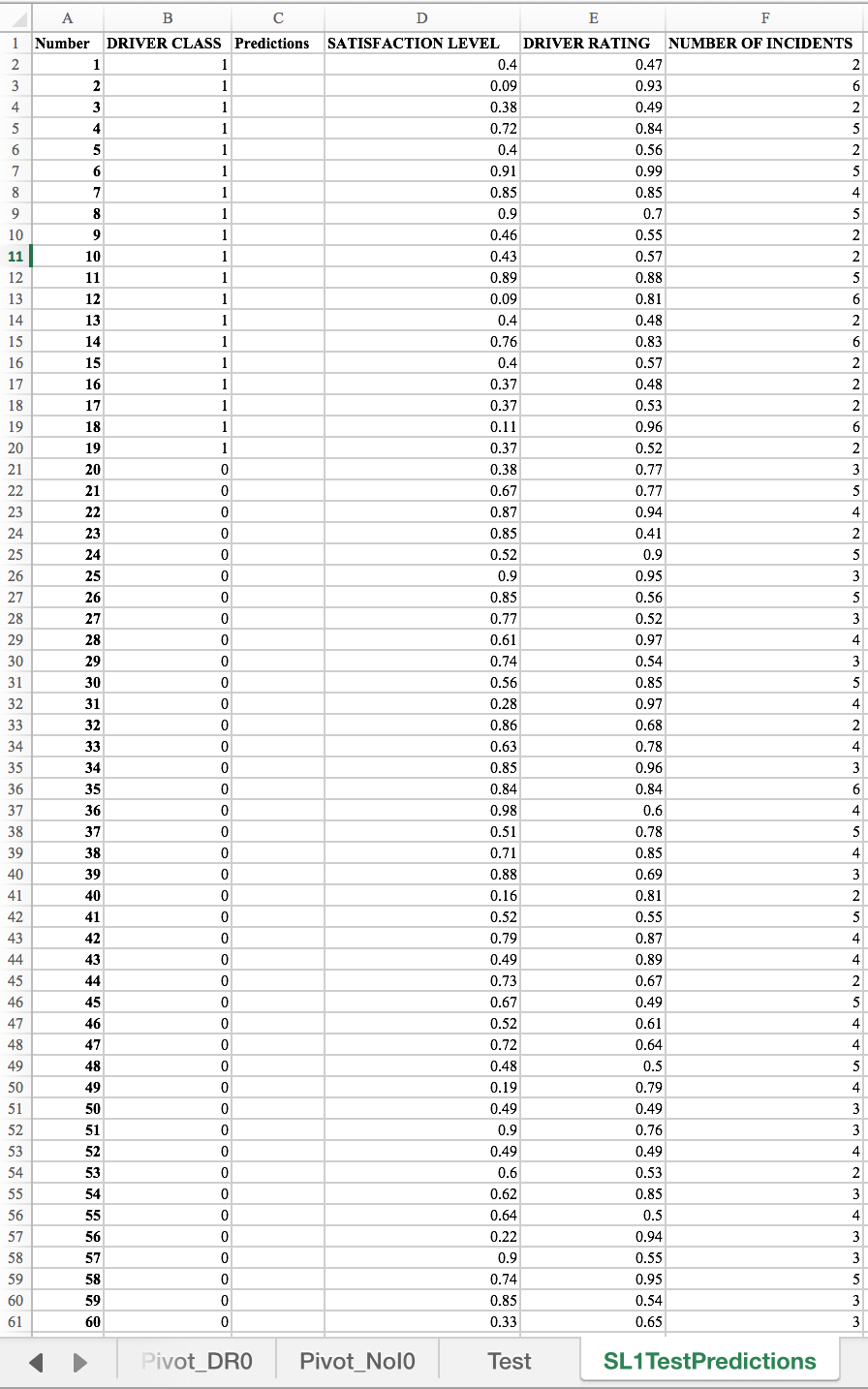
* + - 1. Now in column D we can calculate the probability of each Number of Incidents as its count in column C divided by the total Number of Incidents count. We will label D as P(Number of Incidents|Uber) and insert the equation =C2/$C$7 in D2. We will also add the natural log of the probability in column E using the equation =LN(D2). This gives us the final table below with the end probabilities.

*Figure 4.24: Pivot Table Probabilities*



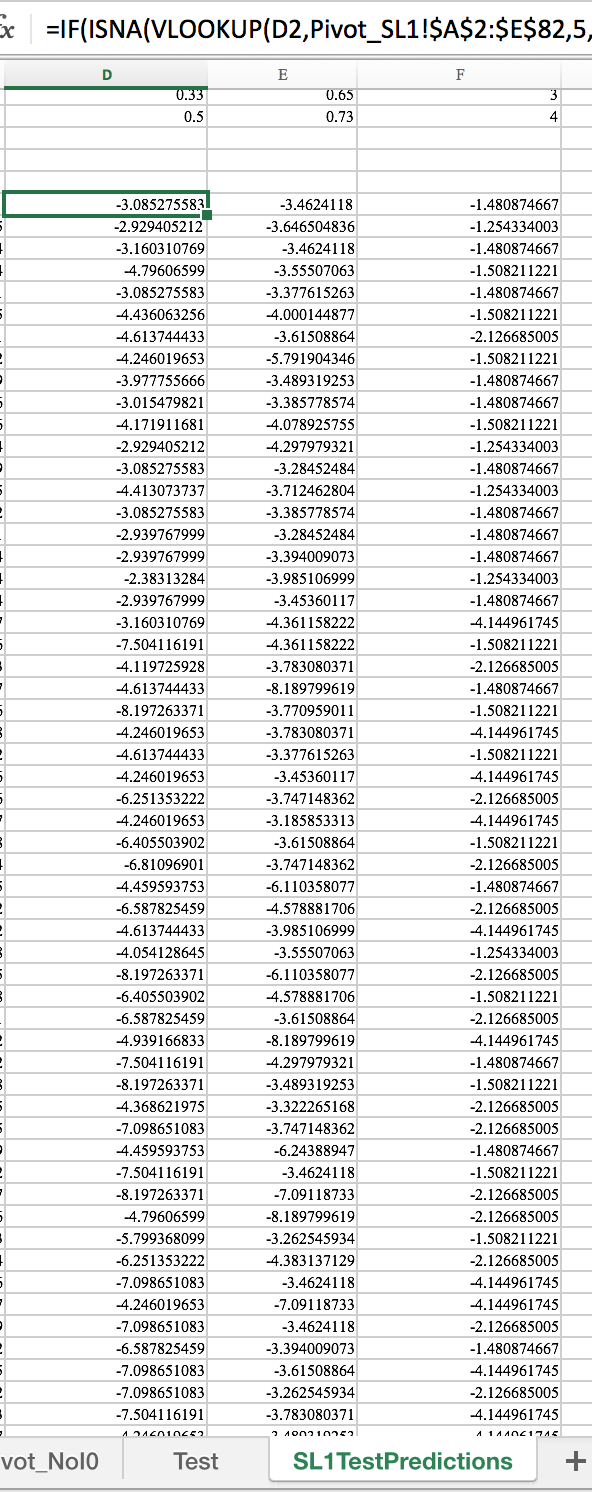
* 1. **Step 2:** Probability Check
     1. The next step is to use the Bayes models on the Test. To do this we will create a new tab called TestPredictions. Column A will filled with numbers listing each driver row. We will then copy Driver Class from the Test tab into column B. In column C we will have the predictions for each class. Lastly, we will copy columns A through C in the Test tab into columns D through F. This gives us the table below.

*Figure 4.25: TestPredictions Setup*

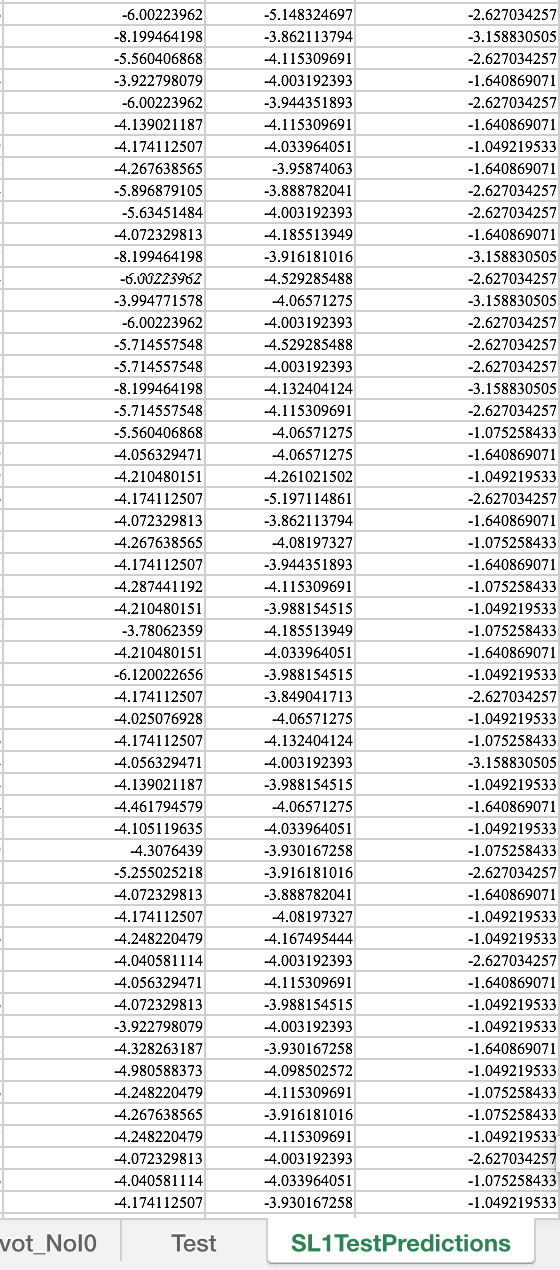
**

* + 1. Now we need to look up the probability for each feature of both classes. To do this we will use the VLOOKUP command. The equations for each class/feature are as follows.
       1. Class: 1
          1. Satisfaction Level: =IF(ISNA(VLOOKUP(D2,Pivot\_SL1!$A$2:$E$82,5,FALSE)),LN(1/Pivot\_SL1!$C$83),VLOOKUP(D2,Pivot\_SL1!$A$2:$E$82,5,FALSE))
          2. Driver Rating: =IF(ISNA(VLOOKUP(E2,Pivot\_DR1!$A$2:$E$55, 5, FALSE)),LN(1/Pivot\_DR1!$C$56),VLOOKUP(E2,Pivot\_DR1!$A$2:$E$55, 5, FALSE))
          3. Number of Incidents: =IF(ISNA(VLOOKUP(F2,Pivot\_NoI1!$A$2:$E$7, 5, FALSE)),LN(1/Pivot\_NoI1!$C$8),VLOOKUP(F2,Pivot\_NoI1!$A$2:$E$7, 5, FALSE))
       2. Class 2
          1. Satisfaction Level: =IF(ISNA(VLOOKUP(D2,Pivot\_SL0!$A$2:$E$90,5,FALSE)),LN(1/Pivot\_SL0!$C$91),VLOOKUP(D2,Pivot\_SL0!$A$2:$E$90,5,FALSE))
          2. Driver Rating: =IF(ISNA(VLOOKUP(E2,Pivot\_DR0!$A$2:$E$66,5,FALSE)),LN(1/Pivot\_DR0!$C$67),VLOOKUP(E2,Pivot\_DR0!$A$2:$E$66,5,FALSE))
          3. Number of Incidents: =IF(ISNA(VLOOKUP(F2,Pivot\_NoI0!$A$2:$E$6,5,FALSE)),LN(1/Pivot\_NoI0!$C$7),VLOOKUP(F2,Pivot\_NoI0!$A$2:$E$6,5,FALSE))
    2. Now that we have the equations we can drag them down and obtain all of the needed probabilities. The result is shown below.

*Figure 4.25: TestPredictions Class 1*

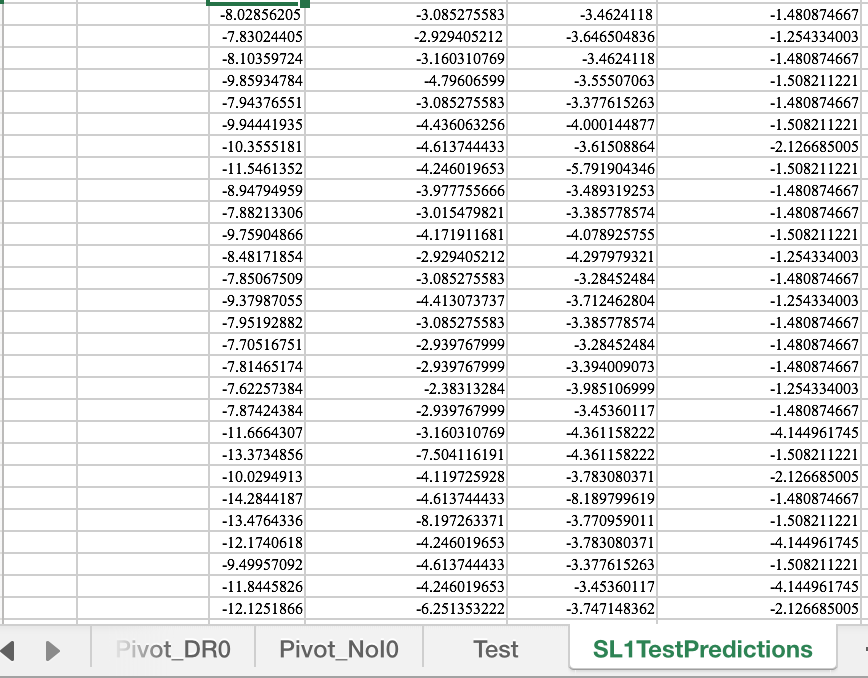
**

*Figure 4.25: TestPredictions Class 2*

**

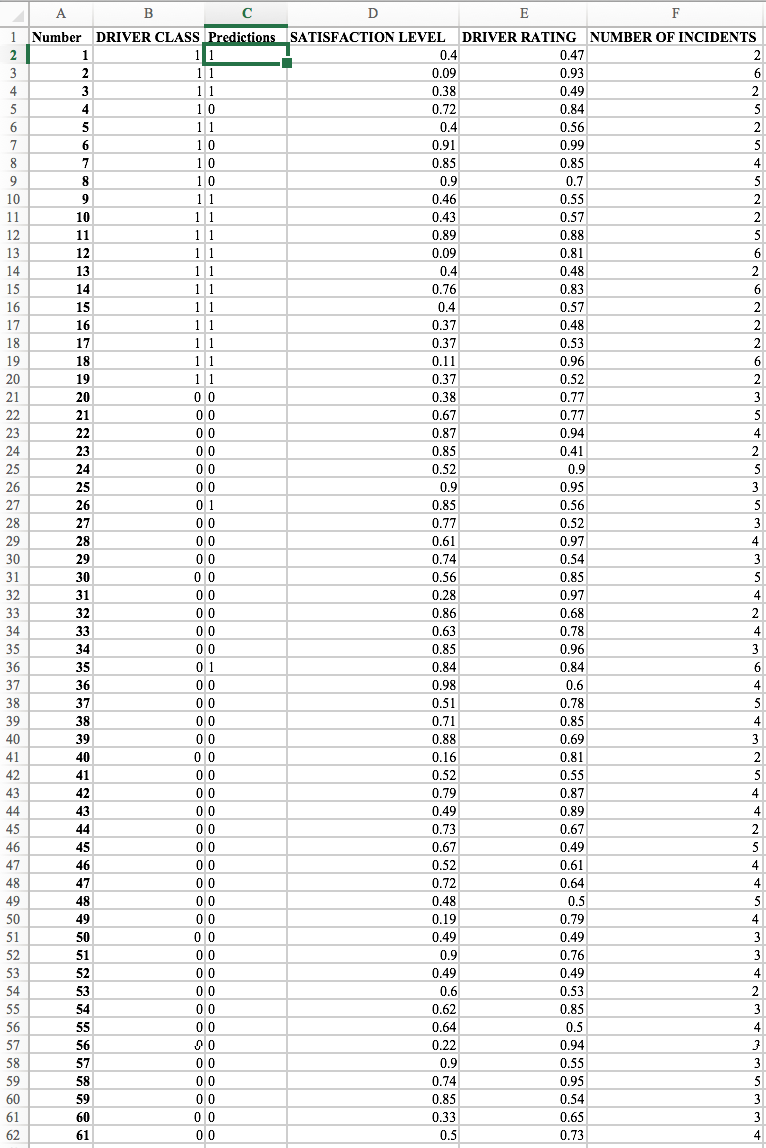
* + 1. Now that we have all of the probabilities we need to sum each row of probabilities with the equation =SUM(D66:F66). We can drag this down the rows for each of the probabilities. This gives use the figure below.

*Figure 4.26: TestPredictions Probability Sums*

****

* + 1. The next step is to classify the drivers. We do this by comparing the scores below in cells C66 and C128 using the following equation: =IF(C66>C128,"1","0"). This gives us the final result below.

*Figure 4.27: TestPredictions Final*

**

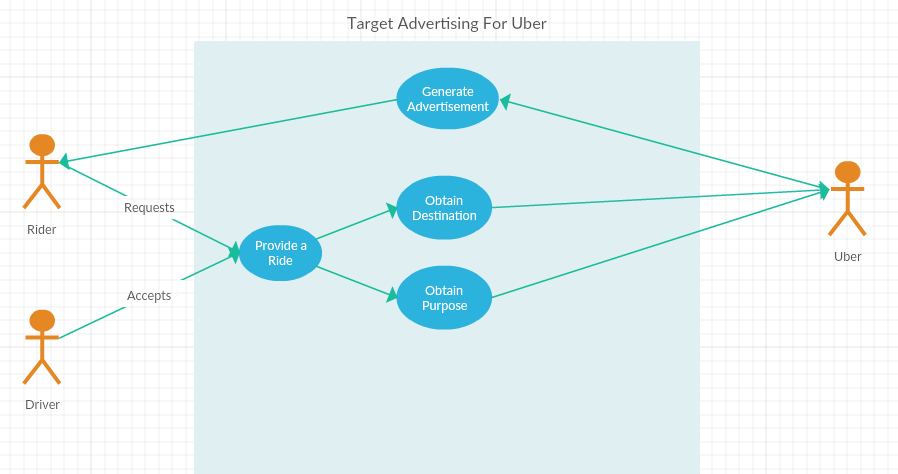
* + 1. As shown above we can see that we got a majority of the predictions correct. We can now calculate the Classifier Accuracy by dividing the number of correct predictions by the total number of Test Data Points. Out of 61 predictions we got 55 correct. This gives us a classifier Accuracy of 90.16%.

1. **Draw Conclusions:**
   1. Overall, I found this problem to be pretty simple. I ended up with a classifier accuracy of 90.16% which is pretty high. This means that 90.16% of the time the classifier will correctly determine whether a driver has stopped using Uber or is still using Uber.

**Problem 5: Software Architecture for Machine Learning Applications**

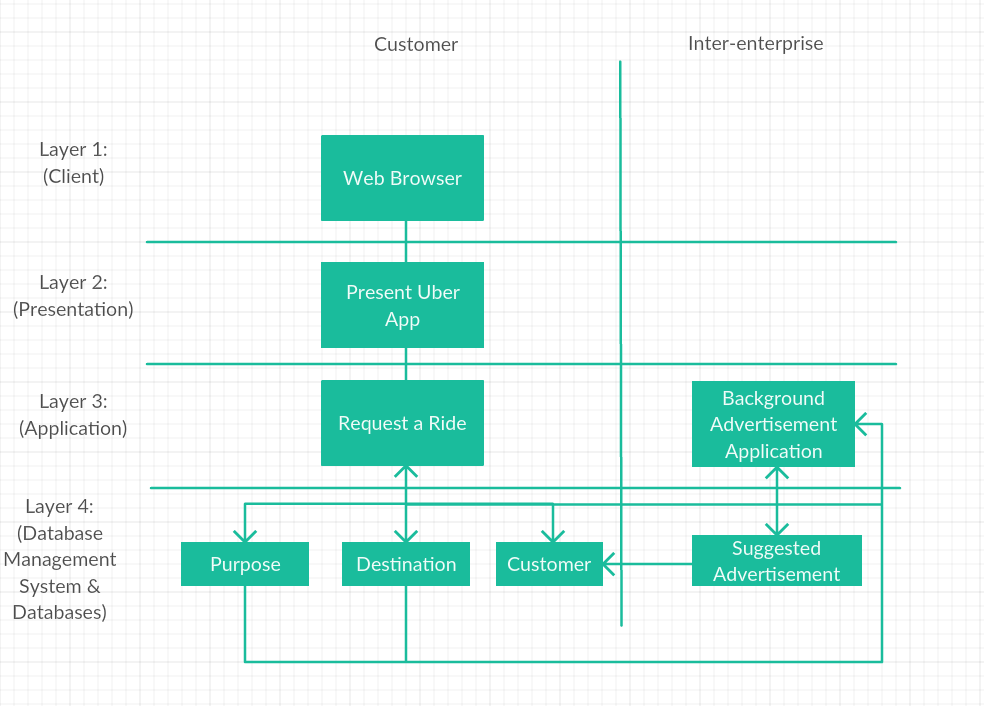
1. **Define the Problem:**
   1. For Uber's clustering machine learning application (Problem 2 above) define the necessary software architecture using the process you created in Problem 4 of the midterm. (Suggestion: First create a good use-case for this targeted advertising application, and then use it to define the layers of the software architecture)
   2. How would you deploy this specific application in the Uber data center that you designed in Problem 4 of the midterm?
   3. How would you deploy this application using Cloud Computing (Midterm, problem 7)?
2. **Create a Plan:**
   1. **Step 1:** Create a use-case for the targeted advertising application.
   2. **Step 2:** Determine the number of server tiers to host the software architecture in step 1.
   3. **Step 3:** Explain how you would deploy the application in the Uber data center created in Problem 4 of the midterm.
   4. **Step 4:** Explain how you would deploy this application using Cloud Computing.
3. **Execute:**
   1. **Step 1:** Use-Case

*Figure 5.1: Targeted Advertisements Use-Case*

****

* + 1. The above use-case shows us the process for Uber to use the targeted advertisements application. They received the destination and purpose information from each trip that the rider receives. This gives the sufficient information to determine their most relevant advertisement.
  1. **Step 2:** Software Architecture
     1. Layer 1: Enable the GUI(Uber application) for the client on the client’s mobile device.
     2. Layer 2: Present the necessary information: Advertisements, surge rates, available rides, driver information, etc..)
        1. Create the GUI for the client
        2. Creates advertisements for the customer.
        3. Enables visualization, data entry, communication with the application & output of results.
     3. Layer 3: Application Logic
        1. Customer Logic
           1. Customer Information
           2. Ride Destination
           3. Ride Purpose
        2. Advertisement Logic
           1. Advertisement Generation for Customer
     4. Layer 4: Manipulate and Store Data
        1. Customer
        2. Advertisement
        3. Purpose
        4. Destination

*Figure 5.2: Software Architecture Diagram for Advertisement Application*

****

* 1. **Step 3:** Application Deployment in the Uber Data Center
     1. To deploy this application in the Uber data center that I designed in Problem 4 of the midterm I would integrate it with the Customer Info and Ride Fulfillment Databases. This is because the ride fulfillment database will give me the necessary information to suggest an advertisement to the customer based off of their destination. The customer info will give me the purposes of the ride and the needed information to send the advertisement to them.
  2. **Step 4:** Application Deployment using Cloud Computing
     1. When we use cloud computing we are configuring multiple application programs to share data in the cloud. This allows the applications to communicate directly with each other. By integrating the advertisement application in the Uber cloud it would make it easier to access the needed information to suggest the advertisement. This would allow the application to access the information real time and update their suggestions more efficiently.

1. **Draw Conclusions:**
   1. Overall, I found this problem to be fairly easy. I looked back at my old midterm and was able to create a software architecture and use-case diagram by combining what I had with the advertisement application that Uber is developing. This allowed me to create a 4 tier software layer focusing on the new application.

**Problem 6: Execution of Your Plan**

1. **Define the Problem:**
   1. Using a table compare your plan from Problem 1 with its execution. Indicate the reasons for the difference between the plan and its execution. Add at least one more column. What should column 4 contain?
2. **Create a Plan:**
   1. **Step 1:** Create a table to compare the plan from problem 1 with its execution.
3. **Execute:**
   1. **Step 1:** Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Problem** | **Plan** | **Execution** | **Reasons for Difference** | **What could you have done better?** |
| 1 | I planned to have this part done by 6/10/17 | I was able to complete it in the planned time. | No difference. | Nothing |
| 2 | I planned to have this part done by 6/11/17 | I wasn’t able to complete it in the planned time. | I did everything except for the conclusion. | Written the conclusion the same day as completing the problem. |
| 3 | I planned to have this part done by 6/11/17 | I wasn’t able to complete it in the planned time. | I did everything except for the conclusion. | Written the conclusion the same day as completing the problem. |
| 4 | I planned to have this part done by 6/11/17 | I wasn’t able to complete it in the planned time. | I did everything except for the conclusion. | Written the conclusion the same day as completing the problem. |
| 5 | I planned to have this part done by 6/12/17 | I was able to complete it in the planned time. | No difference. | Nothing |
| 6 | I planned to have this part done by 6/13/17 | I was able to complete it in the planned time. | No difference. | Nothing |

1. **Draw Conclusions:**
   1. After doing this chart I realized that I have advanced considerably in my project planning skills. I was able to plan accordingly without any delays or major setbacks. I’m glad I was able to improve my project planning capabilities because this is what determines whether the end product will be good.