**Homework 4**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Tyler DiNapoli-Chiappelli\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

There is 20 regular points in this assignment. The minimum increment is 0.5 point. Solve them and fill the answers in the blank space.

**1. Predicting Delayed Flights**

The file HW4\_FlightDelays.csv contains information on historical commercial flights departing the Washington, D.C., area and arriving at the New York City area. For each flight, there is information on the departure and arrival airports, the scheduled time, and so on. The goal is to predict whether a flight between the D.C. area and the New York City area is delayed ***before*** departure. The description of variables is listed below. Suppose there is no missing value.

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Binned\_CRS\_DEP\_TIME** | Binned scheduled departure time |
| **CARRIER** | The airline |
| **DEST** | Destination airport in NY: Kennedy (JFK), LaGuardia (LGA), Newark (EWR) |
| **ORIGIN** | Departure airport in Washington DC: National (DCA), Baltimore-Washington (BWI), Dulles (IAD) |
| **Weather** | Whether the weather on the destination airport was inclement (1) or not (0) when the flight ***arrived*** at its destination |
| **DAY\_WEEK** | Day of week. 1=Mon, 2=Tues, etc. |
| **Flight Status** | Whether the flight was delayed or on time (defined as arriving within 15 min of scheduled time). |

Binned\_CRS\_DEP\_TIME is the scheduled departure time binned into roughly 2hour bins according to the following table (next page).

a. Should we include all the variables in the table for our analysis? Explain your answers. Then, properly set up the variable types and make necessary transformations on the variables you choose to include. When you code a categorical predictor, drop the dummy that corresponds to the most frequent value (i.e., the mode value) of that categorical predictor in the data sample. You can either use Python or other software like MS Excel to retrieve that value for each categorical predictor. Allocate 20% of the data to the test partition. (2 points)

|  |  |  |  |
| --- | --- | --- | --- |
| Scheduled departure time | | | |
| Intervals value | | | Bin to |
| From | To |  | |
| 600 | 800 | 1 | |
| 800 | 1000 | 2 | |
| 1000 | 1200 | 3 | |
| 1200 | 1400 | 4 | |
| 1400 | 1600 | 5 | |
| 1600 | 1800 | 6 | |
| 1800 | 2000 | 7 | |
| 2000 | 2130 | 8 | |

**Please see the submitted Python notebook, section entitled “Part A (explanation)”, for the explanation of variables selection.**

**Please see the submitted Python notebook, section entitled “Part A (Python)”, for the code pre-processing the data.**

b. Use the remaining 80% of the data as the training partition to build logistic regression models with pre-specified penalty levels. Find out which predictor is the most important one. At what range of the penalty level (i.e., the alpha value range) will the most important predictor be rejected from the model? (2 point)

**Please see the submitted Python notebook, section entitled “Part B (explanation)”, for the explanation of the predictor. Please see Parts B1 and B2 for the code.**

c. Use cross-validation to select the model based on accuracy. Set the number of folds for cross-validation to 5. Organize the estimated coefficients and the intercept in the following table (add more rows to the table if necessary). List your predictors in alphabetical order and the intercept in the last row. (2 points)

|  |  |
| --- | --- |
| **Predictor** | **Estimated coefficient** |
| **Binned\_CRS\_DEP\_TIME\_1** | -0.450259 |
| **Binned\_CRS\_DEP\_TIME\_2** | -0.540088 |
| **Binned\_CRS\_DEP\_TIME\_3** | -0.660682 |
| **Binned\_CRS\_DEP\_TIME\_4** | -0.533879 |
| **Binned\_CRS\_DEP\_TIME\_5** | 0.322535 |
| **Binned\_CRS\_DEP\_TIME\_7** | 0.124208 |
| **Binned\_CRS\_DEP\_TIME\_8** | 0.288676 |
| **CARRIER\_CO** | 0.473356 |
| **CARRIER\_DL** | -0.481035 |
| **CARRIER\_MQ** | 0.602919 |
| **CARRIER\_OH** | -1.099688 |
| **CARRIER\_RU** | 0 |
| **CARRIER\_UA** | -0.054363 |
| **CARRIER\_US** | -1.086747 |
| **DAY\_WEEK\_1** | 0.852705 |
| **DAY\_WEEK\_2** | 0.531485 |
| **DAY\_WEEK\_3** | 0.177251 |
| **DAY\_WEEK\_4** | -0.160297 |
| **DAY\_WEEK\_6** | -0.800274 |
| **DAY\_WEEK\_7** | 0.662175 |
| **DEST\_EWR** | 0.025905 |
| **DEST\_JFK** | -0.1851 |
| **ORIGIN\_BWI** | 0.439147 |
| **ORIGIN\_IAD** | 0.293619 |
| **Intercept** | -0.834278 |

The penalty level alpha of the final selected model is **\_0.3012973**

d. Write down the confusion matrix of the final selected model in (c) over the test partition. To earn the full credit, you need create a confusion matrix table like our class examples instead of simply copying and pasting the Python output. Derive the accuracy based on that confusion matrix. Show calculations. After this, include a screenshot of the accuracy reported by the Python method directly. Is it different from what you calculated? (3 points)

**Please see the submitted Python notebook, section entitled “Part D”, for the code.**

Confusion Matrix from manual Excel calculations based off of Python export (please see submitted Excel sheet for calculations):



Confusion Matrix from Python code:

-----Confusion Matrix-----

[[154 16]

[ 69 25]]

Accuracy Rating from Excel (please see submitted Excel sheet for calculations):



Accuracy Rating from Python:

The model's accuracy against the test partition is 0.678030303030303

As can be easily observed, these two calculated accuracy ratings are in fact the same.

e. If we will take a United (UA) flight from IAD to EWR on Monday at 11 am, can we use the final selected model in (c) to predict the probability of being delayed? If yes, please calculate the probability. If no, explain why. (2 points) (hint: you can make a CSV file for this new data observation.)

**Please see the submitted Python notebook, section entitled “Part E (explanation)”, for the explanation and “Part E (Python)” for the code and probability result.**

Submit your Python code with the filename [DM2020] HW4\_Q1\_YOURFULLNAME.ipynb

2. **The Universal Bank Personal Loan**. Recall our personal loan class example. Download the dataset PersonalLoan.csv from Content \ Python Materials \ Logistic Regression on Blackboard. Suppose now the cost of sending an offer is $2 and the net profit from one acceptance after factoring in the cost of sending the offer is $5. **(Need the class content on Oct. 26th)**

a. What is the decision cut-off value we should use for the decision rule here? Show calculation. Then write down the decision rule as well. (2 points)

x = the probability that a customer will accept the loan offer

Expected profit (send) = 5x - 2(1-x) = 7x - 2

Expected profit (don’t send) = 0

E(send) = E(don’t send) = 0 = 7x – 2

decision value = 2/7 = x

**Decision Rule:** If P(acceptance of loan) > 2/7, THEN send; ELSE do not send

b. Allocate 20% of the data to the test partition. Find the observations in the test partition. Calculate the average net profit if the Universal Bank follows the naïve strategy – sending the offers to everyone – in the test partition. Is the average net profit positive or negative, i.e., is the bank making money or losing money? (2 points)

**Please see the submitted Python notebook, section entitled “Question #2: Part B (explanation)”, for the explanation and “Question #2: Part B (Python)” for the calculation**

c. Use cross-validation to select the logistic regression model based on average net profit. Set the number of folds for cross-validation to 5. What is the average net profit over the test partition if the University Bank adopts the final selected model and the decision rule in (a)? (2 point)

**Please see the submitted Python notebook, section entitled “****Question #2: Part C” for the result**

d. How much is the difference between the average net profit achieved by the naïve strategy in (b) and the average net profit achieved by the final selected model in (c)? (1 point)

**Please see the submitted Python notebook, section entitled “Question #2: Part D” for the result**

e. Now, the cost of sending an offer is updated to $8. The net profit from one acceptance after factoring in the cost of sending the offer is updated to $20. Is the final selected model with the updated cost and benefits the same as, or different than, the final selected model in (c)? Justify your answers by attaching the related Python output. (2 point)

**Please see the submitted Python notebook, section entitled “Question #2: Part E (explanation)”, for the explanation and “Question #2: Part E (Python)” for the code.**

Submit your Python code with the filename [DM2020] HW4\_Q2\_YOURFULLNAME.ipynb