Lab 3. Logistic Regression with TensorFlow

In this lab, we are going to take use the ‘bank\_data.csv’ file, which contains marketing data from a Portugese bank to determine as to whether or not to classify a particular customer as being a proper target for a marketing campaign to sell a new product.

The marketing data has the following fields:

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| Age | Numeric |
| Job | Categorical (“admin”, “blue-collar”, “entrepreneur”, “housemaid”, “management”, “retired”, “self-employed”, “services”, “student”, “technician”, “unemployed”, “unknown” ) |
| Marital | marital status (categorical: “divorced”, “married”, “single”, “unknown”) |
| Education | Education (categorical: “basic.4y”, “basic.6y”, “basic.9y”, “high.school”, “illiterate”, “professional.course”, “university.degree”, “unknown”) |
| Default | Default: has credit in default? (categorical: “no”, “yes”, “unknown”) |
| Balance | Contact: contact communication type (categorical: “cellular”, “telephone”) |
| Housing | Housing: has housing loan? (categorical: “no”, “yes”, “unknown”) |
| Loan | Loan: has personal loan? (categorical: “no”, “yes”, “unknown”) |
| Contact | Contact: contact communication type (categorical: “cellular”, “telephone”) |
| Day | day\_of\_week: last contact day of the week (categorical: “mon”, “tue”, “wed”, “thu”, “fri”) |
| Month | Month: last contact month of year (categorical: “jan”, “feb”, “mar”, …, “nov”, “dec”) |
| Duration | Duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y=’no’). The duration is not known before a call is performed, also, after the end of the call, y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model |
| Campaign | Campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact) |
| Pdays | pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted) |
| Previous | previous: number of contacts performed before this campaign and for this client (numeric) |
| Poutcome | poutcome: outcome of the previous marketing campaign (categorical: “failure”, “nonexistent”, “success”) |
| y | The outcome (1 = “Yes”, 0 = “No” |

In the following assignment, you will write a logistic regression model to classify customers as being qualified for a successful marketing outcome.

Step 1. Load the dataset into your code. It is recommended that you use the pandas module to do this.

Step 2. which features you will train the model on.

Step 3. Drop all non-essential data columns

Step 4. Convert categorical data (if any) to one-hot vectors. Hint: Look, at the pd.get\_dummies() method from Pandas. Don’t forget to drop the ‘unknown’ columns from the data.

Step 5. Split your data between the testing and training sets. Try a 70/30 percentage split, where the first number is for the training set and the second is for the testing set.

Step 6. Define your logistic regression model.

Step 7. Define your cost function (Hint: look at the tensorflow function tf.nn.softmax\_cross\_entropy\_with\_logits())

Step 8. Define your gradient optimizer function. (There are many to choose from)

Step 9. Once you’ve built the graph run it on the training set. Print out the cost for this model every 50 iterations.

Step 10. Feed the testing data to the model.

Step 11. Print out a *confusion matrix* (Hint: Look at the tf.confusion\_matrix() method.). From this matrix, calculate and print the F score, the model’s precision and the model’s recall.