Section 1.5: The Modeling Step (Model Selection Process)

Monday, September 5, 2022 17:15

[00:00] Introduction:

RECALL: We need to know which model to choose for our project:

- Logistic regression
- Decision tree
- Neural network
- Or many others

[01:02] Holdout + Train:

Selecting the best model:

Suppose that it is July and we train a model, X; we take y, and produce model g, then deploy the model in August. We can take a small part of the dataset, say 20% hide it away; then we use the 20% for training only. In essence, the 20% will act as the new data from August. The other 80% will be used for testing.

These percentage rates aren't fixed, but roughly an example. The percentage of data you would appoint to testing and training can vary.

In general, we want to see how the model will perform against the data it has not seen.

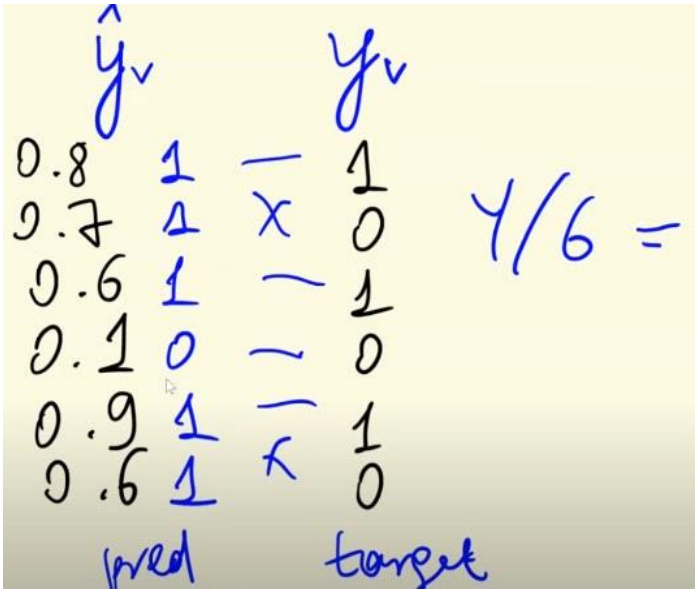
[03:28] Making Predictions:

So, now we have two groups of datasets: the training dataset and the validation dataset. We need to extract the Feature Matrix, **X** from the training data; we also have the **y**, also coming from the training data and we train our model **g**, using only **X** and **y**.

Then from the validation dataset, we extract a Feature Matrix unique to the validation dataset, **X-Validation** $\equiv \mathbf{X_v}$, and to get a **target variable** $\equiv \mathbf{y_v}$. We have the g from our training dataset, and we apply it to our validation dataset to get some predictions to be called, $\hat{\mathbf{y}}_\mathbf{v}$.

Next, we need to compare to compare $\mathbf{y_v}$ **to** $\hat{\mathbf{y}}_\mathbf{v}$**.**

So, in practice, we would get two target arrays that look something like this (prediction vs. target variables):



We can see, in how many cases this is correct. In four out of six case, the prediction is correct. This is to say, the model we have is 66% accurate.

[06:54] Scoring:

So, we can do this for logistic regression (LR), and we see that the model is 66% accurate while it operates under a specific g, say in this case, g_1.

Then we take a decision tree (DT), which is a different model family that operates under a different g, say g_2, and suppose that model is 60% accurate.

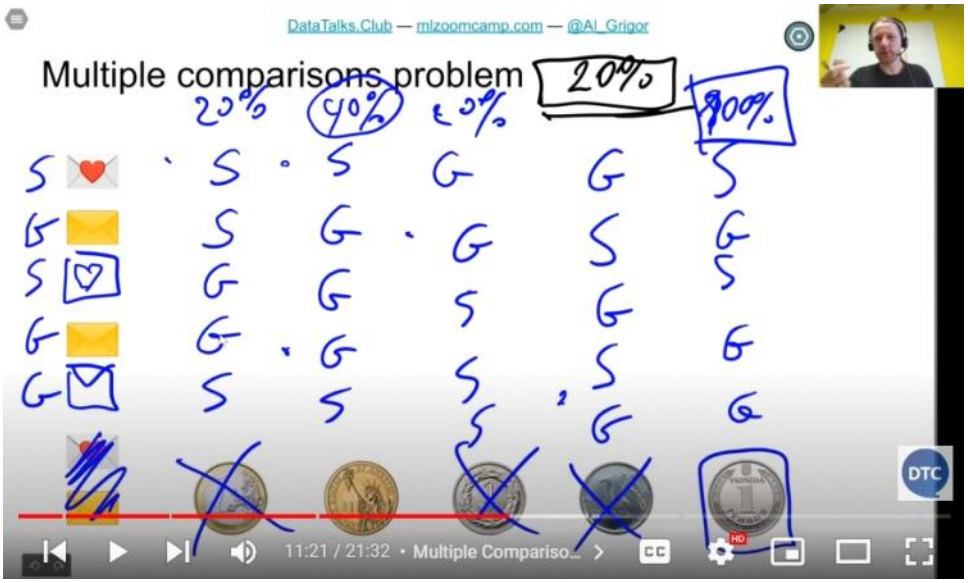
Another model we might execute another model RT, where it is 67% accurate.

Suppose we execute a neural network (NN) that is 80% accurate; we would select this model because it has the greatest amount of accuracy.

[07:58] Multiple Comparison Problems:

There could be a problem with this approach; assume our model is not a logistic regression model or a neural network model, it's a coin.

Take our 20% of data which happened to be, say 5 emails, we take our coin, and if it lands on heads, then it is spam, versus if it lands on tails, it is not spam.



It is possible for models to get lucky.

[13:12] Train + Validation + Test

To guard against cases where luck comes into the fold, what we do is instead of holding out just one dataset, we hold out two datasets.

So, we may take 20% for validation, 20% for testing, and 60% for training. What we have are three, non-overlapping subsets. We hide away the test dataset. Next, we do the modeling selection as previously described, and determine all of our variables.

For the training dataset, we take our Feature Matrix, X, and plug into our model g to get y; for the validation dataset, we take our Validation Feature Matrix, X_v to get y_v. Here, we select the best model.

To ensure the particular model did not get lucky with the validation dataset, we take it and apply it to the test dataset to extract our test feature matrix X_T, and our test target variable y_T and we conduct an extra round of validation with the test dataset.

[15:17] Further Scoring:

So, if we have logistic regression (LR) at 66%, decision tree (DT) at 60%, random force (RF) at 67%, and neural network (NN) at 80%; we know the NN is the best one.

Next, we take the NN and apply it to the test dataset. If the test dataset generates a 79% result, compare it to the 80% result generated by the NN, then we know that it is a good model to utilize.

[16:28] Model Selection (6 Steps):

1. Split the dataset.
2. Train a model.
3. Validate a model using a validation dataset.

Steps 1 through 3 are an iterative process; you repeat them as many times as necessary.

4. Select the best model.
5. Apply to the test dataset.
6. Check

Usually between steps 4 and 5 we can take the training dataset and the validation dataset, and train the model. Afterwards, we apply it to the test dataset. Then we check the test data set against the larger dataset.

[20:49] Summary:

We need to understand certain tools to utilize in the next sections; NumPy, Panda, etc.

Notes by Community:

The validation dataset is not used in training. There are feature matrices and y vectors for both training and validation datasets. The model is fitted with training data, and it is used to predict the y values of the validation feature matrix. Then, the predicted y values (probabilities) are compared with the actual y values.

Multiple comparisons problem (MCP): just by chance one model can be lucky and obtain good predictions because all of them are probabilistic.

The test set can help to avoid the MCP. Obtention of the best model is done with the training and validation datasets, while the test dataset is used for assuring that the proposed best model is the best.

1. Split datasets in training, validation, and test.
2. Train the models
3. Evaluate the models
4. Select the best model
5. Apply the best model to the test dataset
6. Compare the performance metrics of validation and test