Machine Learning Project

Methodology

Instructions

Please run the Project1.ipynb file from the same location as training.psv and eval.psv files.

Background

Given the task of predicting majors for students and datasets of students' grades in courses and levels those courses were taken, it is obvious that grades and levels would be input features while the output labels would be the majors.

Data Set Analysis

The data sets were given in psv format, it was necessary to pivot the data to get a dataset in the following format to provide input features:

Course Student_id	c1	c2	c3
st1	A+	NaN	B
st2	NaN	B+	NaN

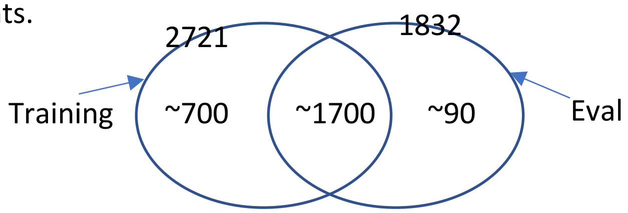
Course Student_id	c1	c2	c3
st1	Senior	NaN	Junior
st2	NaN	Sophomore	NaN

Additionally training set was pivoted to obtain labels as follows:

Student_id	Major
st1	major1
st2	major2
st3	major3

When both datasets were pivoted as above, following issues were noted:

1. There were 2721 courses in the training set while there were only 1832 courses in the eval set. These could be presented well in a Venn diagram as there were intersections and disjoints.



2. In order to have consistent sets of input features two approaches could be taken:
   I. Remove the courses data only in Training set and only in Eval set and consider only the intersecting courses data of ~1700 courses.
   II. Fill the missing courses data in Training and Eval with some assumptions and have all ~2800 courses data.

3. From the two approaches, II is preferred due to following reasons:
    I. Approach I will ignore valuable courses data of ~800 courses, which is a waste of very valuable resource.
    II. Approach I will only give ~1700 courses for the model to train, which could adversely impact the accuracy of predictions, because less data gives less accuracy.
    III. If we assume that the dataset is complete, then missing courses data in training set indicates that students in training set has not taken some courses in the eval set throughout their four years in college. So we can fill NaN values for those courses for the students in the training set. Similarly we can do for missing courses in eval set. This will give more data and greater accuracy.

Assumptions

- Dataset is complete and there are not any data missing from the dataset.
  For instance, if there are courses in training set that are not in eval set, that reflects no student in eval set has taken those courses. Further this does not reflect any courses data missing from the eval set.
- The grades ‘I’ and ‘WX’ are considered as separate grades even though they are not mentioned in the project assignment document.

Preprocessing

I. As suggested above, data sets are pivoted to get student_id as index, courses as columns and grades as values. In the similar fashion, another pivoted table is obtained student_id as index, courses as columns and levels as values.

II. NaN values were replaced with 0’s in both datasets.

III. Both **integer encoding** and **one hot encoding** has been used to convert categorical values of grades, levels and majors in the datasets. However majors could only be encoded in one hot as there is a requirement to obtain 3 top majors for a student. If integer encoding is used we will be able to provide only one major for each student.
Integer encoding of grades:
{'A+':21, 'A':20, 'A-':19, 'B+':18, 'B':17, 'B-':16,
'C+':15, 'C':14, 'C-':13, 'D+':12, 'D':11, 'D-':10,
'R':9, 'WX':8, 'P':7, 'AUS':6, 'S':5, 'AUU':4,
'N':3, 'U':2, 'I':1 }
Integer encoding of levels:
{'Senior':4, 'Junior':3, 'Sophomore':2, ‘Freshman':1}

IV. When integer encoding is used the features were **normalized** to improve training speed.

V. Several approaches have been configured with one hot encoding, with integer encoding, with both grades and levels as input features and with only grades as input features.

Libraries

Python **pandas** library

Training model

Several approaches were considered as highlighted in the following table:

Approach	Validation Accuracy	# of Input features	Batch size	Pros	Cons
One hot encoding grades and levels(1)	~90%	~18000	64		Slow training Overfitting the training set due to large number of festures
One hot encoding grades only(2)	~90%	~11000	64	Faster than above	Slow training Overfitting the training set due to large number of festures
Integer encoding grades and levels(4)	~95%	~5600	32	Faster	Fast training Less overfitting
Integer encoding grades only(3)	**~95%**	**~2800**	**16**	**Fastest**	**Fastest training** **Least overfitting**

Batch sizes were selected primarily to reduce training time. Since higher input features require longer training time, they were trained with large batch size.

Parameters for best neural network architecture

Model Type : Sequential

Input Layer

Layer type : Dropout layer with 0.5 dropout(reduces overfitting)
Input features : Grades for courses
No of Nodes in Input layer : 2721 nodes for courses

Hidden Layer

Layer type : Dense layer
No of Nodes in Hidden layer : 200
Activation function : Relu(could use Tanh too)

Output Layer

Layer type : Dense
No of output nodes : 81 nodes for 81 majors
Activation function : Softmax layer(since probabilities are required for each major)

Optimizer	:	Adam(could use rmsprop too)
Loss	:	Categorical Crossentropy(Since output is one hot encoded)
Metrics	:	Accuracy

Libraries

Python **keras** library

Outcome

Majors were predicted using the model trained with the approach 3 and output to pred.psv file. Major1, major2 and major3 can be interpreted as follows as I have extracted the majors for top 3 probabilities:

major1 = Most probable major
major2 = Second most probable major
major3 = Third most probable major