Statistical Tasks

1. **Defaulting on Bills**: In this case we have a classifier for which all data can be grouped into 4 categories True Positives (TP) and True Negatives (TN) for labels classified correctly and False Positives (FP) and False Negatives (FN) for misclassified labels. The problem states that FPR is 4% this means by the FPR definition:

$$FPR = 0.04 = \frac{FP}{FP + TN}$$

By a similar definition we know the FNR to be:

$$FPR = 0 = \frac{FN}{FN + TP}$$

Therefore:

$$FN = 0$$

Also the we know that 1% of the bills have a default and we can express it as the true probability of defaulting or:

$$0.01 = \frac{\mathrm{TP + FN}}{\mathrm{TP + FN + FP + TN}}$$

Finally we have that the probability of default given that a shop tests positive is:

$$\mathrm{P(default \mid test +)} = \frac{TP}{\mathrm{TP{+}FP}}$$

We can get P(default I test +) by algebra manipulations and arrive at a figure of:

P(default I test +) = 0.2016129

To know if the insurance is cost effective the expected value of variable M representing the money, can be calculated as:

$$E(M) = 0.2016129 \times \$400 - (1 - 0.2016129) \times \$100$$
$$E(M) = 0.80645\$ \text{ per shop}$$

On average 80 cents are gained per shop, the deal should be taken.

2. **Normal distribution hypothesis testing.**

a. **Rejection Region:** For this case the type of test used should be a two tails z-test for the mean value. The problem complies with this test conditions (being a normal distribution, having more than 30 data points and knowing the standard deviation). The mean behaves as a normal distribution, where its equation is:

$$z = \frac{\overline{x} - \mu}{\sigma}\sqrt{n}$$

On this equation, z is given by the standard normal distribution, while n=100 (number of data points) and x hat is the sample mean. The rejection region with a 5% significance level, means that there are 2 tails of the z distribution each with 2.5% probability, according to the z table the z values are -1.96 and 1.96 for the left and right tail respectively. By solving the above equation for x hat the rejection region is:

$$\begin{gathered} \mu < 2.608 \\ or \\ \mu > 3.392 \end{gathered}$$

b. **The p-value** is the probability of having equal or more unlikely results after drawing a sample. For this case it is the cumulative probability of one of the tales depending on whether the sample mean is above or below 3. If the mean is 5 then then z is 10 according to the equation above. Therefore the p-value on the z-table for 10 is very close to 0. The hypothesis should be rejected because we have a p-value lower than 2.5%.

c. **The power** is the probability of rejecting the null hypothesis when it is false. It is the area under the curve for the normal distribution centered in 4, when it is outside of the rejection region from part a. Therefore it is:

$$p = P(\overline{x} \geq 3.392) + P(\overline{x} \leq 2.608)$$

$$p = P(z \geq \frac{3.392 - 4}{5}) + P(z \geq \frac{2.608 - 4}{5})$$

$$p = 0.9387$$

3. Bayes Table

a. There are 3 hypothesis (one for each die type). The six sided type has a prior probability of 0.5 while the others are 0.25 due to the fact that there are 2 six sided dies while only one eight and twelve sided dies. The likelihood of getting a 7 on a six sided die is 0 while the others is 1/8 and 1/12 for the eight sided and twelve sided respectively. By using Bayes Theorem one can conclude that if 7 appears there is a 0.6 probability that the eight sided die was selected while a 0.4 probability that the twelve sided die was selected. The Table is:

Bayes Table

hypothesis H	quantity	prior probability P(H)	likelihood P(D I H)	unnormalized posterior P(H) * P(D I H)	posterior P(H I D)
six sided die	2	0.5	0	0	0
eight sided die	1	0.25	0.125	0.03125	0.6
twelve sided die	1	0.25	0.0833333333333333	0.0208333333333333	0.4
			P(D) =	0.0520833333333333	1

b. The posterior probability that the die has 12 sides is 0.4.

c. If the exact same experiment is repeated then both experiments can be considered independent from one another. Therefore the probability that the next roll is a 7 would be P(D) or 0.052.

Machine Learning Task

Data Processing

The input fields are “name”(text), “description”(text), “shop”(categorical), “brand”, “price”(float) and indirectly the “label names” also contain some information.

Fields “name” and “description” were long strings with lots of words. This words were filtered with at least 4 characters long, counted and ordered by frequency. Then the top 1% were added back to their respective fields. After this the hash trick was applied to all fields except price. This is because hashing is convenient for managing categorical fields with lots of values and can output a tunable amount of dimensions. While hot encoding would have resulted in too many dimensions, difficult to deal later on the learning stage.

The price field was scaled to have a unit variance and zero mean, this makes it easier for the optimization algorithm to work, as widely uneven values lead to numerical errors.

A new field was created by searching “label name” on “name” and “description” fields, there was a small improvement. Also, another field was created by searching the brothers, those categories with common parent of “label name”, in “name” and “description”, but no improvement was found so this last change was discarded.

Learning

The problem was divided by levels.

Level 1

There are 3 categories on the tree top, and they were encoded on 3 arrays. A multi class neural network was designed with 4 relu neurons in the first layer and 3 output sigmoid neurons in the second layer. The system was trained with cross-validation first, to prevent any overfitting from tuning parameters.

Level 2, 3 and 4

For this levels there are too many categories for hot encoding and the problem didn't fit the multi class problem type because there could be 2 or more categories per row. Instead as the problem requires at least 1 true positive per row, the categories are order by frequency in the training set and the most frequent ones are trained as they are the ones with the greatest chance of scoring true positives. The models used here are binary classifiers learning to detect when their label is present, they use 4 relu neurons as input layer and a sigmoid neuron as output layer.

Tuning

Hashing dimensions

The number of dimensions resulting from the hashing of input fields was tuned by running the level 1 classifier on cross validation mode, in this process the "brand" field was removed completely as it didn't provide any improvement on the classification error, but added noise and dimensions.

Models

The loss parameter was tuned, for level 1 the "categorical_crossentropy" is used in conjunction with the "accuracy" metric. While on the other levels "mean_squared_error" is used. The size of the network was severely limited by computational resources, the size of the data set and available time.

Sensitivity and specificity

The output of the binary classifiers for levels 2, 3 and 4 are tuned such that their output has the expected value as the training set. Added to this formula there is a specificity multiplier that can alter this equation and shift the output towards sensitivity or specificity, it is set on specificity_multiplier = 0.9 (were 1 is neutral) slightly towards sensitivity.

Number of categories per level

Only the most frequent categories were selected to run models on them. Tuning of the top number of categories was done by looking at the improvement over time.

Results

This results use 7% of the total data or 14000 rows. They took 4765.77s to complete, computational resources and time were limited.

- Data percentage with at least a true positive for a level 2 category: 73.57%

under specificity_multiplier =1.0 and 10% of all data:

- Data percentage with at least a true positive for a level 4 category: 13.63%
- Data percentage of rows correctly unlabeled for all 4 level categories: 54.37%
- Bonus, percentage with multiple level 4 rows: 5.33%

Additional statistics

Accuracy for level 1: 71.32%
Data percentage with at least a true positive for a level 3 category: 41.18%

Other

The code runs by executing:

```
$ python machine_learning_task.py
```

This script will call helper scripts:

models.py
text_analisis.py
util.py
result.py

Code written on python 3.5.1 with libraries:

1. Keras using TensorFlow for the backend
2. sklearn
3. pandas
4. numpy