Introduction for ML: Assignment 1

Vsevolod Klyushev v.klyushev@innopolis.university

Report:

Preprocessing:

Encoding:

Because column 'var6' has only 2 values 'yes' and 'no', I decided to encode it using ordinal encoding. It is quite obvious that these values have relationships with each other.

There are quite a lot of values in column 'var3' (236), it can be encoded using ordinal encoding or one-hot encoding. In the case of using ordinal encoding technic, we will get an unnecessary relationships between different values, but this method does not increase the size of our dataset. One-hot encoding, in turn, allows you to encode categorical dates without creating unnecessary links in it, however, in this case the number of columns in the dataset will increase significantly, which will complicate the training of models in the future.

Column 'var7' with the date can be dropped or encoded using cyclic encoding. In order to encode column 'var7', you first need to parse the data in it into years, months, days, hours, minutes and seconds. This way you can get 6 new columns. However, I got rid of the columns over the years, because all the data in it were the same, which will not help the models in any way. The remaining columns can be encoded using cyclic encoding, using the $sin(\frac{2\pi \cdot val}{freq})$ and $cos(\frac{2\pi \cdot val}{freq}))$ functions, thereby we encode the information in each time column using two columns with sinusoidal and cosinusoidal functions, respectively.

Thus, the categorical data in the original dataset can be encoded in 4 ways. I created 4 datasets corresponding to each encoding option:

- `df_ord_country` - ‘var6’ and ‘var3’ are encoded by an ordinal encoder and ‘var7’ is dropped.

- `df_one_hot_country` - ‘var6’ is encoded by an ordinal encoder, ‘var3’ is encoded by the one-hot encoding method, and finally ‘var7’ is dropped.
- `df_ord_country_with_date` - ‘var6’ and ‘var3’ are encoded by ordinal encoder and ‘var7’ is encoded with help of cyclic encoding.
- `df_one_hot_country_with_date` - ‘var6’ is encoded by an ordinal encoder, ‘var3’ is encoded by the one-hot encoding method, and finally ‘var7’ is encoded with help of cyclic encoding.

Data scaling technique:

Since we want to ensure that the results of different models can be compared correctly and we do not want to face the fact that any feature affects the model more than others because of its value, we must use data scaling technique. I decided to use a standard scaler for columns with non-categorical data ('var1','var2','var4','var5') to bring the data to a normal distribution for each dataset. Moreover, since after applying ordinal encoding to the ‘var3’ column, the values in it become from 0 to 235, it makes sense to use a standard scaler for ‘var3’ column in `df_ord_country` and `df_ord_country_with_date` datasets.

Data imputation:

Before filling in the missing values in column ‘var4’, you need to decide which model is best to use for this. I decided to compare linear regression and polynomial regressions with degrees from 2 to 4. For datasets with one-hot encoding of column ‘var3’, I considered polynomial regression only of the 2nd degree, because in this case the calculation time becomes quite large.

In general, linear models are quite good, but they have less flexibility compared to polynomial models, although the last ones have the risk of overfitting. Linear models assume a linear relationship between features and a goal, while polynomial models consider a more complex relationship between features and a target. Both linear and polynomial models have a convex form of the loss function, which allows it to be minimized. I will compare the performance of the models using the MAE and MSE metrics.

It happens to be that polynomial regression with 2nd degree has the best performance for all datasets. Moreover ordinal encoding of countries without date has the best performance among encoding techniques. Therefore I'll use polynomial regression model with 2nd degree and ordinal encoding of var3 without date dataset to predict the missing values in column 'var4' for all datasets.

Training:

I tested 3 types of models (Logistic Regression, Naïve Bayes, KNN) on all 4 datasets. To select hyperparameters for models with KNN, I used Grid Search with a cross validation parameter cv=3 and an F1 score. I evaluate the performance of the other two models using cross validation with 3 folds and an average F1 score for them.

After comparing of all obtained F1 scores, I came to conclusion that almost all models coped well with the predictions, with the exception of Naïve Bayes for datasets with one-hot encoding for the 'var3' column. I suppose this is due to the fact that Naïve Bayes assumes the independence of features from each other, while all columns with cities are mutually exclusive of each other. This is what worsens the F1 metric. However, KNN model with cosine metric, distance weight and n_neighbours=11 on a dataset `df_ord_country` showed slightly better results than other models.

Feature analysis:

In order to find out which features most affected the target variable, we can consider the weights assigned by the logistic regression for each of them. According to my observations, the most significant features are: ‘var1’, ‘var2’, ‘var4’, ‘var5’, ‘var6’. When encoding ‘var3’ with one-hot, a number of features that characterize countries such as Hungary, Denmark, San Marino, etc. stand out well. When encoding ‘var7’, it can be highlighted that the day column has some influence. The rest of the features do not particularly affect the model.

In general, for datasets

- `df_ord_country` most critical features are ‘var1’, ‘var2’, ‘var4’, ‘var5’, ‘var6’, and redundant feature is ‘var3’
- `df_ord_country_with_date` most critical features are ‘var1’, ‘var2’, ‘var4’, ‘var5’, ‘var6’, ‘day_sin’, ‘day_cos’, and redundant feature is ‘var3’
- `df_one_hot_country` most critical features are ‘var1’, ‘var2’, ‘var4’, ‘var5’, ‘var6’, some of ‘var3_Country’ (Hungary, San Marino, etc.), and redundant features are almost all that we receive from ‘var3’ by using one-hot encoding
- `df_one_hot_country_with_date` most critical features are ‘var1’, ‘var2’, ‘var4’, ‘var5’, ‘var6’, ‘day_sin’, ‘day_cos’, some of ‘var3_Country’ (Hungary, San Marino, etc.), and redundant features are almost all that we receive from ‘var3’ by using one-hot encoding

PCA influence on data models:

The dimensionality reduction by the PCA significantly improve performance for some of the models according to F1 score, in particular, for Naïve Bayes for datasets with one-hot encoding for the 'var3' column. I suppose that the reason for it is that PCA reduces redundant columns and columns with strong relationship between each other. Performance of other models doesn’t changed a lot (within 1%). It might be because KNN and Logistic Regression models don’t overfit on redundant features.

Multilabel learning problem:

Multi-label learning problem is generalization of multiclass classification problem. In such problem there is no constraint on the maximum number of the classes the instance can be assigned to. However, there must be at least two unique classes to which an instance can be assigned.

We can transform our problem into a multi-label problem by simply making it so that all categorical variables in our dataset can have multiple class labels at the same time.

However, in this case, we will face the problem that our models cannot cope with the task. If we want to use the same models, we can Transform multilabel learning problem info multi-class classification problem by encoding each categorical feature as a powerset. Thus, each set of classes will have its own identifier, which we will predict.