Term project report

Shpir Mariia, Volodymyr Beimuk, Daniil Bilohrudov

December 2022

Data Prepocessing

- Removed variables: 'partlybad', 'date', 'id'.

We looked at the training dataset and found out that there is only one unique value ('False') for a variable 'partlybad'. Therefore it can be removed. In addition, when examining the test data, we found that the variable 'date' is hidden. Consequently, we cannot use it for fitting the model.

- Created variable 'class2'.

```
df['class2'] = np.where(df['class4'] == 'nonevent', 0, 1)
```

- Encoded categorical variable 'class4'.

```
from sklearn.preprocessing import OrdinalEncoder
ohe = OrdinalEncoder()
df['class4'] = ohe.fit_transform(df['class4'].to_numpy().reshape(-1, 1))
```

Data Analysis: Problems and Issues

The number of datapoints in total: 464
The number of variables: 100

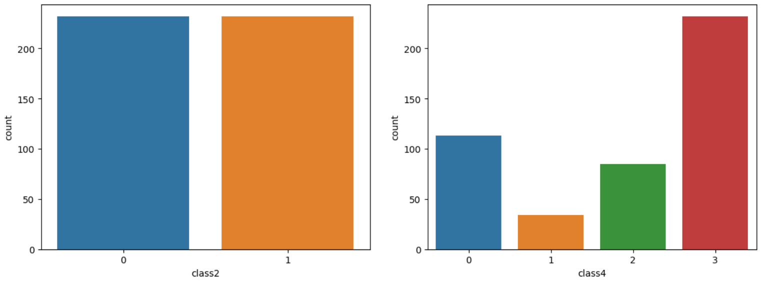


Figure 1. The number of datapoints for each class. (a) 0 - 'nonevent', 1 - 'event'. (b) 0 - 'II', 1 - 'Ia', 2 - 'Ib', 3 - 'nonevent'.

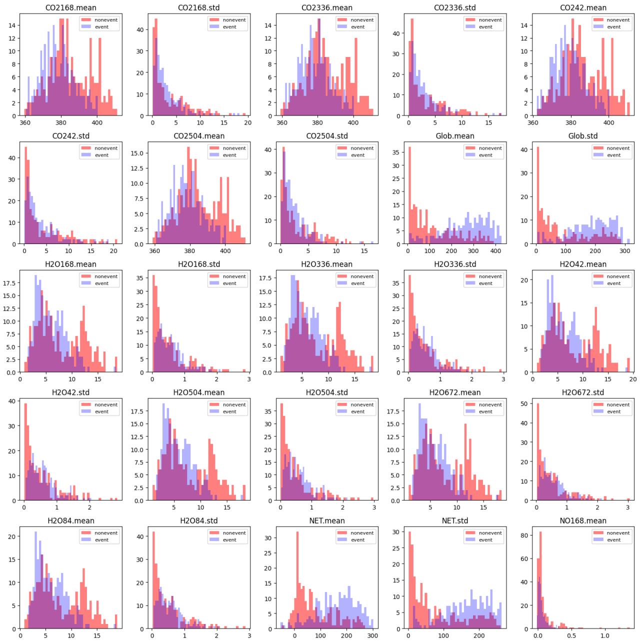


Figure 2. The distribution of the first 25 variables according to the binary class.

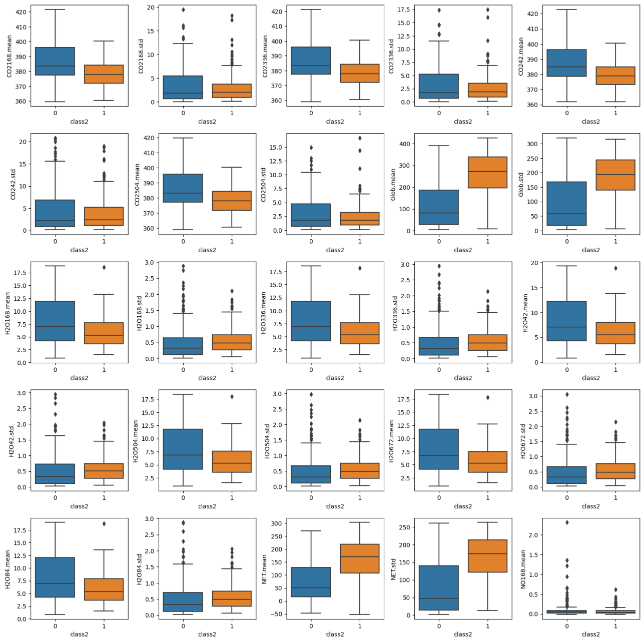
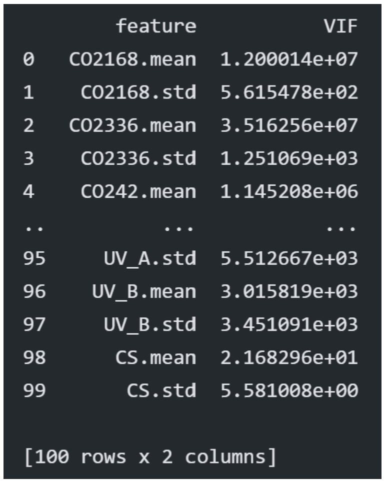


Figure 3. The boxplots of the first 25 variables according to the binary class.



```
         feature           VIF
0   CO2168.mean  1.200014e+07
1    CO2168.std  5.615478e+02
2   CO2336.mean  3.516256e+07
3    CO2336.std  1.251069e+03
4    CO242.mean  1.145208e+06
..          ...           ...
95     UV_A.std  5.512667e+03
96    UV_B.mean  3.015819e+03
97     UV_B.std  3.451091e+03
98      CS.mean  2.168296e+01
99       CS.std  5.581008e+00

[100 rows x 2 columns]
```

Figure 4. VIF for random variables.

Conclusion:

- **Quite small dataset:** Due to the small size of the dataset, we should be careful with model selection and not use complex architectures (such as neural networks) to avoid overfitting.

- **Outliers:** (Based on Figure 3) We should remove the outliers to improve the accuracy of the models.

- **Imbalanced data for multi-class prediction:** (Based on Figure 1) Ignored data imbalance can greatly affect the accuracy of the results. It is worth processing the data before training the model.

- **Large number of variables. Multicollinearity:** (Based on Figures 2, 4) Multicollinearity is noticeable when the VIF value is large, as well as when finding similar distribution plots. It is worth either reducing the number of variables or using various automatic algorithms (e.g. PCA, Regularization etc.)

Method: Choosing model

Dealing with the outliers: In our work we used Unsupervised Outlier Detection using the Local Outlier Factor (class LocalOutlierFactor from sklearn.neighbors). The analysis revealed 12 outliers. After the removal the training features' shape is (452, 100), the training labels' shape is (452,).

Next, we decided to deal with two different problems: Binary and Multi-class Classification. First we will define the binary class of the datapoint, and then (in case of detecting an event) the multi-class label. If we predict both binary and multi-class labels, the models may predict different classes (e.g. 'nonevent' and 'Ia'), which is completely contrary to our goals.

Binary classification.

For binary classification we decided to test the accuracy of such models: GaussianNB, SVC, RandomForestClassifier, DecisionTreeClassifier, QDA, KNeighborsClassifier, LogisticRegression.

We fitted the models and tuned hyperparameters with 20-fold Cross-Validation algorithm (using class GridSearchCV from sklearn.model_selection). The hyperparameters for each model will be presented below, as well as lists of parameter settings to try as values. Also the top 3 best models and their accuracy will be reported accordingly.

GaussianNB: 81.4%
KNeighborsClassifier: 80.9%
DecisionTreeClassifier: 82%
LDA: 88.6%
QDA: 82.2%

SVC: 87.8%
'degree': [1, 2, 3, 4],
'C': [10000, 1000, 100, 10, 5, 1, 0.5, 0.1]

C	degree	accuracy
1000	3	87.8%
1000	1	87.5%
100	3	87.3%

LogisticRegression: 87.5%
'C': [10000, 1000, 100, 10, 5, 1, 0.5, 0.1, 0.01, 0.001, 0.0001],
'fit_intercept': [True, False]

C	intercept	accuracy
0.001	True / False	87.5%
0.1	False	87%
1000	True	86.7%

RandomForestClassifier: 88.9%
'max_depth': [5, 6, 7, 8, 9],
'n_estimators': [2, 4, 8, 16, 32]

depth	estimators	accuracy
5	16	88.9%
8	16	88.5%
6	32	88.4%

We did not select hyperparameters for the all models, since the initial result of some models was significantly lower compared to others. It is obvious that because of multicollinearity naive Bayes assumption does not work, hence the accuracy is quite low and we could not use this model. KNN classifier is too 'simple' for this type of task. Decision tree showed lower results compared to the Random forest, so we will not use this algorithm too.

Also based on the results, we can see that polynomials not of the first degree show better results in the case of SVM. Therefore, we decided to additionally check the polynomials of the second and third degree for LogisticRegression.

LogisticRegression with 2 degree Polynomial: 88.7%

C	intercept	accuracy
10000	False	88.6%
0.1	True	88.5%
0.001	True / False	88.3%

LogisticRegression with 3 degree Polynomial: 85%

C	intercept	accuracy
0.0001	False	85%
0.1	True	84.7%
0.01	True	84.7%

Based on the results, we can conclude that it is not worth checking degrees further, since the best accuracy of the model at higher complexity is smaller. And also, we managed to achieve higher accuracy with a second degree polynomial.

Next we selected the best models and compute their accuracy using Leave-One-Out CV method.

model	accuracy	description
LDA	89.4%	
RF	88.6%	depth = 5, estimators = 16
LR	87.2%	degree = 2, C = 10000, intercept = False
SVC	86.7%	degree = 3, C = 1000

And then we finally checked the data on the remaining test dataset (20% of the full dataset, 'npf_train.csv' file).

model	accuracy	perplexity
LDA	87.9%	1.36
RF	85.7%	1.35
SVC	85.7%	1.34
LR	84.6%	1.32

Experiments: Additionally, we have experimented with different scalers (MinMaxScaler and StandardScaler from sklearn.preprocessing), also with Principal component analysis (PCA from sklearn.decomposition). The results are as follows:

- We trained our models with PCA decomposition using different values for the variance (from 50% to 95%). In addition, we previously scaled the data (with StandardScaler). As the result, every model presented smaller accuracy than it was expected.

- Additionally, we removed all mean values presented in the data and vice versa the standard deviation values. According to the results, again, there was no improvement.

- We tried scaling the data without PCA and, to our surprise, SVC model got 91.2% accuracy and 1.28 perplexity on the test dataset.

Summary for binary classification: At the end, we decided to use scaled SVC since both the accuracy and perplexity of this approach showed the best results. But in addition we experimented further with both RF and LDA.

Multi-class prediction.

Next, we selected all the events. To predict the 'class4' variable and to deal with imbalanced data we used oversampling algorithm as it performs better than undersampling.

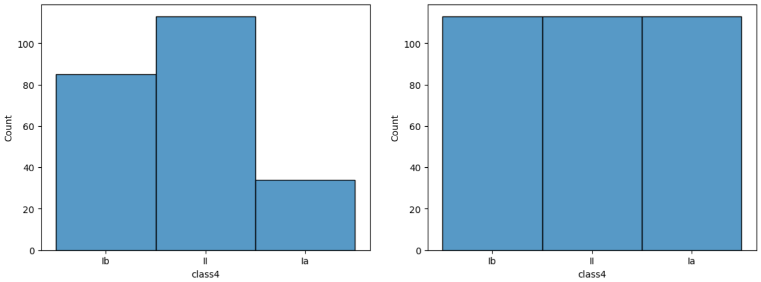


Figure 5. The number of datapoints for each class before and after oversampling algorithm.

Similar to the selection of a model for binary classification, we selected a model for multi-class classification. We used 20-Fold CV to find best hyperparameters for every model. A table of results will be shown below:

model	accuracy	description
RF	78.3%	depth = 8, estimators = 32
SVC	66.9%	degree = 4, C = 10000
LDA	63.5%	
LR	55.5%	degree = 2, C = 0.5, intercept = True

Here it is obvious that Random Forest Classifier has the highest accuracy.

Summary

In the end, we selected two models:
- SVC (degree = 3, C = 1000) with scaled data for binary classification.
- Random Forest Classifier (depth = 8, estimators = 32) for multi-class prediction.

Step-by-step:
- Removed the outliers with Unsupervised Outlier Detection.
- Standardized features by removing the mean and scaled to unit variance.
- Predicted the 'event', 'nonevent' days with SVC model.
- Used oversampling method for the 'event' days to balance the classes.
- Predicted the type of the 'event' days with Random Forest Classifier.
- 'Merged' the output from 2 models.

The results (using 'npf_test.csv'):

Performance measure	Competition	Now
Binary accuracy	85.6%	86.3%
Perplexity	1.44	1.36
Multi-class accuracy	65.1%	67.5%

We were able to slightly improve the results. Because of the extra time we used Leave-One-Out CV and found better models. Also we slightly changed our approach and now predicted 'event' day type based on the binary classification results, instead of the opposite, as it was done for the competition.

Confusion matrix for binary classification

_	nonevent	event
nonevent	460	66
event	66	373

Confusion matrix for multi-class prediction

_	II	Ia	Ib	nonevent
II	116	7	67	43
Ia	14	7	13	6
Ib	71	7	73	15
nonevent	39	13	18	456

According to the matrix, we can see that the low accuracy of multi-classification (65% on average) due to the fact that the model can not identify Ia 'event' day, which more often defines as 'nonevent'. In addition, 'Ib' often can be mistaken with 'II' and vice versa. On the other hand, the model defines 'nonevent' days quite accurately.

Grading

Grade for the deliverables: 5

The treatment of the topics shows in-depth understanding, the relevant source material is used and cited, and the discussions show maturity. Appropriate machine learning and other methods have been chosen and applied correctly.

It's pretty hard to evaluate on our own. We learned how to work with most of the machine learning methods presented in the course and used (or experimented with) them in our final work. For example, we studied and used PCA, and normalized and scaled the data (since it is presented in two different types (mean and std). Additionally, we used the regularization coefficient to avoid overfitting (which is possible, because we have a huge number of features on a small number of datapoints). We also used polynomials of varying degrees to work with linear models (Logistic regression), and thus improved the accuracy of this model. Finally, we used various cross validation methods (20-Fold and Leave-One-Out) both to select hyperparameters and to maximize 'accuracy of model accuracy'.

The methods used have been analysed sufficiently. The reporting is to the point and exact. The conclusions drawn are in-depth and to the end. The discussion of findings shows an aptitude for independent, critical, and innovative research and thinking.

We analyzed each method used and described all the results in the report. In addition, we used various figures and tables for a more accurate explanation.

The reports and presentations are polished and "camera-ready." The work has been creative and independent and progressed within the given schedule. The deliverables have been done by using the instructions provided.

The presentation, data for the competition and the preliminary report were sent on time and (certainly to our taste) not badly edited and prepared. We tried to meet all the requirements that were presented.

Grade for the group as a whole: 5
Each of us has done his part on this project. We all participated in the discussions and were responsible (most of the time) in completing our tasks.