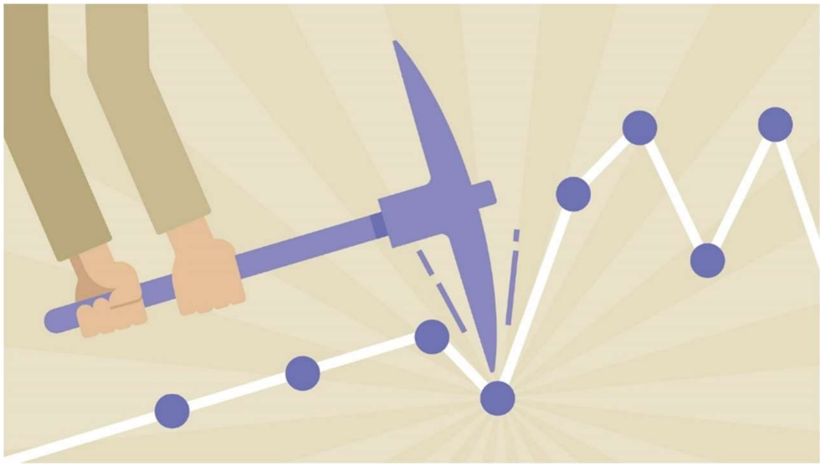

DATA MINING- FINAL REPORT
2024

Anh Lê

Simona Neaga

GitHub link: https://github.com/phanthuyanh/Final-Deliverables-Data-Mining

Table of Contents

Introduction and Execution Explained

The deliverable report aims to satisfy the stretch-level assessment. In this process, two datasets were chosen – first to mention is the Stress Detection based on sleep patterns, and second is the Weather Prediction based on meteoritical measurements. In each dataset, besides two mandatory modellings - Lazy Learning with k-Nearest Neighbors (KNN) and Naive Bayes (NB), one more optional modelling is also chosen and conducted, which is the Decision Tree Classifier. After the modelling to predict the dependent variables, the performance of each model is evaluated against each other in the same dataset, followed by relevant discussion and suggestions for the next step.

Hereby is the understanding summary of the three chosen models, first two of which are covered during the Data Modeling module and the remaining are explored.

- **Lazy Learning with k-Nearest Neighbors (KNN)**

KNN is an algorithm commonly used in data mining that predicts a new data point based on the majority class or average value of the closest neighbored data points. The number of the closest points for which the prediction of the new point depends on is referred to as k.

- **Naive Bayes (NB)**

NB classifier is a machine learning algorithm based on Bayes' Theorem in statistics about probability. To predict the outcome of one new data point, the algorithm calculates the probability of each class and picks the one with the highest probability, assuming all values are independent.

There are three types of NB classifiers: Multinomial NB, Gaussian NB, and Bernoulli NB. During class, the usage of Multinomial NB was demonstrated with the example of classifying spam from ham emails. After further research, we found out that Multinomial NB is specifically applicable for discrete data only, which makes it most optimal for natural language processing such as email classification. However, considering the nature of both of our datasets, where predictors are measurable and continuous, Gaussian NB would be the most suitable. The remaining NB option, Bernoulli NB, is used with Boolean variables, and thereby not matching for our dataset ("What are Naive Bayes classifiers?," n.d.)

- **Decision Tree**

A Decision Tree Classifier is a learning algorithm that has a hierarchical tree structure, consisting of a root node, branches, internal and leaf nodes. Starting with the root node, the outgoing branches are connected to the internal nodes. From these two nodes, evaluations are conducted to create groups with similar characteristics and represented by leaf nodes, or all the possible outcomes within the dataset ("What is a Decision Tree?," n.d.).

One highlight from our progress was that the firstly chosen dataset, Stress Detection through Sleep, is highly to be computer generated, which is supported by the precedented modelling on Kaggle platform, where the dataset was shared publicly. The results of the KNN and NB modelling on this dataset also yielded an accuracy score of 1 (https://www.kaggle.com/datasets/laavanya/human-stress-detection-in-and-through-sleep/discussion/308789), making the goal of evaluating the performance between different models as required by the assignment more challenging to reach. Therefore, we decided to choose one more dataset, which is the Weather Prediction, to better conduct the model-evaluating goal and detect the potential issue with the Stress Detection dataset from the beginning by comparison. Therefore, as to be observed in this report, we will go through each phase of the CRISP model, and in each part, explanations of the firstly chosen Stress Detection dataset will be written with more details, and then followed by the difference we noticed in the secondly chosen Weather Prediction dataset compared to the first one while conducting the same phase.

Business Understanding

- **Dataset 1: Stress Detection**

Many studies have pointed out the deeply connected and two-way relationship between stress and sleep (Martire et. al, 2020). Based on the context provided for this dataset, being able to monitor and measure the physiological patterns and habits in sleep could assist with detecting stressed conditions. This could serve as the foundation for further diagnosis to detect stressors and provide timely treatment and adjustments to people's lifestyle.

The prediction of stress levels based on sleep patterns and habits also provides business opportunity to highly develop tech-driven self-monitoring tools, which could be more friendly and portable towards individuals with non-technical or non-medical background. One interesting example is SaYoPillow case, where the chosen dataset also came from (Rachakonda et. al, 2020).

- **Dataset 2: Weather Prediction**

Weather forecasts have been playing a vital role in people's lives. Being able to know the weather situation in advance could enhance planning quality and life experience, not to mention that necessary protective approaches could be executed timely in case of severe weather.

Data Understanding

General overview:

The size of the dataset has consistency, listing nine columns which represent the parameters, and 631 rows as answers/individuals' information generated through the Smart-Yoga Pillow experiment. The methods used for gathering information are mentioned in the article, which clearly states the following:

"An edge processor with a model analyzing the physiological changes that occur during sleep along with the sleeping habits is proposed. Based on these changes during sleep, stress prediction for the following day is proposed. The secure transfer of the analyzed stress data along with the average physiological changes to the IoT cloud for storage is implemented. A secure transfer of any data from the cloud to any third party applications is also proposed. A user interface is provided allowing the user to control data accessibility and visibility. SaYoPillow is novel, with security features as well as consideration of sleeping habits for stress reduction, with an accuracy of up to 96%." (Rachakonda et. al., 2020)

Every independent variable represents an important factor, which are named or abbreviated as the following:

- **Sr=** represents the snoring range of the user, (45 until 100)
- **Rr=** it shows the respiration rate.
- **t**= signifies body temperature**.**
- **lm=** denotes limb movement rate.
- **bo=** represents blood oxygen levels.
- **rem**= indicates eye movement.
- **no.1 sr=** number of hours of sleep.
- **sl=** reflects stress levels**,** 0 representing low/normal, 1- medium low, 2- medium, 3-medium high and 4 high).
- **hr=** denotes heart rate.

EDA:

Based on the exploratory data analysis, there are no missing values and no outliers. Moreover, with the help of pair plot, there is evidence of the changes in the high stress levels (3 and 4) based on the snoring rate, limb movement, eye movement, blood oxygen and temperature.

With the use of the heatmap, there are noticeable correlations between:

1) Respiration rate and heart rate, limb movement and heart rate, respiration and limb movement rate, body temperature and blood oxygen levels' (perfect positive correlation)
2) Numbers hours of sleep and stress levels, body temperature and stress levels. (almost perfect negative correlation)

Additionally, the dataset shows numerical/ordinal, with a meaningful order attributed to the stress levels.

Comparison with Seattle Weather Data

When it comes to Seattle weather, we notice a consistency of 1461 entries, indexed from 0 to 1460, with a total of six columns. There are non-null values, and primarily floats, stored numerical values with decimal points, excluding the date and weather which are type object (categorical), indicating string values.

With the use of the Correlation Matrix, we can notice a quite strong positive correlation between temp_min and temp_max, as one variable increases, the other tends to increase as well.

Each independent variable represents an important factor, which are named or abbreviated as the following:

- Temp_min: the minimum temperature on that specific date.
- Temp_max: the maximum temperature on that specific date.
- Weather: being a categorical variable, it lists five different types of weather: snow, sun, rain, drizzle, fog.
- Precipitation.
- Date.

Data Preparation

In the data preparation process, we will check thoroughly when it comes to missing values, duplications, outliers, and data splitting in order to ensure the quality of our datasets.

Observations:

When it comes to outliers or duplicates, there have been no signs of missing values or duplicates (by using the duplicated () code) during the examination. There are neither sign of missing values.

However, it contains only one categorical variable in both sets, which necessitates minimal cleaning. Additionally, there was a need to adjust the data type, by modifying 'weather' from object to category to mitigate potential modelling issues. Since weather categories are independent, the ordering was set to False to maintain that independence. This has also been applied to 'stress_level' from object to category to facilitate model compatibility.

Within the DT model, there will be several libraries needed, being accuracy_score, confusion_matrix and also using the train and test data (X and Y).

The train-test split is allocated in such a way that 70% is the training and 30% for testing. For this approach, we have excluded from the training data the categorical columns in order to prevent bias.

Modelling

Stress Detection:

Respectively, the KNN, NB, and Decision Tree are conducted and evaluated. For evaluation, a confusion matrix is displayed to compare the predicted values against the real value in the dataset and wrapped up using the accuracy score. The modeling on this Weather Detection dataset is conducted as a comparison to the experience on the Stress Detection dataset, so we keep the test and train set size the same between the two datasets.

KNN:

When it comes to using the needed libraries for KNN, in order to make classifications and predictions on the set, we applied the code KNeighborsClassifier() and knn.predict(), from sklearn package. .

For the training data, as it is mentioned in the Data Preparation, we will have x_train and y_train by using the train_test_split() function. Once it has been trained, the prediction function is applied on the testing dataset (x_test). This makes it easier to apply a class label to each data piece.

Once done, we used confusion_matrix to present an explanation of the model's projections and actual results. The matrix helps in showing how many instances were correctly classified or misclassified.

NB:

We imported the GaussianNB from a comparable package to build the technique. Again, the training and testing tests are separated so that we can analyze their performance. After the training phase, we use the prediction technique (nb.predict()). All of this has been accomplished effectively by first eliminating the category column.

DT:

The code for the Decision Tree algorithm was imported from scikit-learn, which was then separated into data and model training. The Decision Tree Model (dt) is initialized during the Model training process using the Decision TreeClassifier(). Later, we train the model with the fit () function, supplying the training features and related labels, x_train and y_train. Finally, the accuracy score determined the model's overall accuracy on the test set, reflecting the percentage of properly anticipated cases.

Evaluation

Evaluation of Model Performance: Stress Detection

There have been various similarities and differences identified using the three models, KNN, NB, and DT. When it comes to accuracy, the Decision Tree Model scored 77%, lower than KNN and NB, both of which scored 100%. This mismatch can be traced to dataset overfitting, which is caused mostly by the presence of generated information.

Comparison to Weather Prediction:

We took into consideration the stress detection and meteorological data, as the weather in Seattle was not generated. This DT resulted in an accuracy of 84%, but in comparison, KNN and NB had an accuracy range of 76% to 84%.

Comparison of data patterns between the datasets

- **Unexplainable effect of multicollinearity on the prediction performance**

It was discovered that a greater correlation between independent variables, or multicollinearity, is available in Stress Detection as contrasted to Weather Predictions. Many sources support the fact that multicollinearity may affect the performance of parametric algorithms such as NB, as NB assumes that variables are independent of each other, or regressions due to the coefficients, but do not on Decision Tree (Sighn, 2023) (Cheruku, 2019). However, this does not apply to our results as NB yielded good accuracy score on both models; Decision Tree, theoretically should not be affected, performed worse on the dataset that has high multicollinearity (which is Stress Detection) and better on that with little multicollinearity. This remains inconcludable as we could not find a legitimate academic study regarding the effects of multicollinearity on predictive models' performance.

- **Distribution among each class of the dependent variables**

There is a noticeable difference where the distribution amount between each class of the dependent variables varied greatly, between the two datasets. The Stress Detection showed an equal distribution throughout stress levels, each level accounts for 20%; however, in the weather dataset, the two classes “rain” and “sun” each consumes 44% of the total entries, and the remaining five classes only consist of less than 5%. Secondly, based on the pair plots, the distribution of the predictors in the stress detection dataset is multimodal, while the distribution of those of the weather prediction dataset is mostly normal - skewed distribution, which indicates that outlier's effect in the Weather Prediction model is more significant.

We believe these differences, in class distributions and outlier’s prevalence, between the 2 datasets, explain why the KNN models performed worst in the Weather Prediction compared to the Stress Detection. This is understandable as these two factors affect the way KNN makes its predictions based on the nearest in distance variables (Awan, 2023).

Deployment

Based on the evaluation, deploying the model in the Stress Detection requires careful consideration of its performance, characteristics and potential challenges or obstacles.

Model Selection

Based on the high performance of the mentioned models, these are primes candidates for deployment.

Furthermore, understanding the data patterns differences between the two sets is crucial, as the presence of multicollinearity, highly varied distribution between each class, and data distributions, outliers require more careful addressing, as this may affect the prediction accuracy.

Conclusion and Reflection

Understanding the whole process in Python coding was initially challenging. However, the initial time and effort invested to understand the process based on the example given by the lecturer helps the process.

- It was hard to determine whether a dataset is high-quality, and if the quality of the models' prediction is due to the quality of the dataset itself or our data preparation and modeling.
- Knowledge in other courses of the minor such as Data Modelling is also helpful.

Future experience suggestion

- For future data mining experience, during the preparation steps, it might help improve the data quality for prediction by processing outliers – perhaps using Cook's D value to detect and then use judgment to consider keeping or dropping outliers.
- While doing research to determine the reason behind each model's accuracy performance, we found out that the differences in distribution between the test and train set may play a role. Therefore, spending time to examine the data patterns between the test and train set may add value to the learning experience.
- At the moment, the resources that address the effect of data patterns on the model's prediction performance are not easily accessible or available. Therefore, this might be an interesting topic to cover during this Data Mining course or other academic study.

References

Awan, A. A. (2023, June 14). K-Nearest Neighbors (KNN) Classification with R Tutorial. https://www.datacamp.com/tutorial/k-nearest-neighbors-knn-classification-with-r-tutorial

Bhandari, A. (2024, February 15). Multicollinearity | Causes, effects and detection using VIF (Updated 2024). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/03/what-ismulticollinearity/

Cheruku, S. K. (2019, December 8). What is Multicollinearity and how it affects model performance in Machine Learning? https://www.linkedin.com/pulse/what-multicollinearity-how-affects-model-performance-machine-cheruku/

Cheruku, S. K. (2019, December 8). What is Multicollinearity and how it affects model performance in Machine Learning? https://www.linkedin.com/pulse/what-multicollinearity-how-affects-model-performance-machine-cheruku/

L. Rachakonda, A. K. Bapatla, S. P. Mohanty, and E. Kougianos, "SaYoPillow: Blockchain-Integrated Privacy-Assured IoMT Framework for Stress Management Considering Sleeping Habits," IEEE Transactions on Consumer Electronics (TCE), Vol. 67, No. 1, Feb 2021, pp. 20-29.

L. Rachakonda, S. P. Mohanty, E. Kougianos, K. Karunakaran, and M. Ganapathiraju, "Smart-Pillow: An IoT based Device for Stress Detection Considering Sleeping Habits," in Proceedings of the 4th IEEE International Symposium on Smart Electronic Systems (iSES), 2018, pp. 161--166.

Martire, V. L., Caruso, D., Palagini, L., Zoccoli, G., & Bastianini S. (2020). Stress & sleep: A relationship lasting a lifetime. *Neuroscience & Biobehavioral Reviews*, 117, 65-77. https://doi.org/10.1016/j.neubiorev.2019.08.024.

Rachakonda, L., Bapatla, A. K., Mohanty, S. P., & Kougianos, E. (2020). Sayopillow: a blockchain-enabled, privacy-assured framework for stress detection, prediction and control considering sleeping habits in the IomT. National Science Foundation. https://par.nsf.gov/servlets/purl/10247733

Singh, R. (2023, March 5). Multicollinearity in machine learning. https://www.linkedin.com/pulse/multicollinearity-machine-learning-rachit-singh/

What are naïve Bayes classifiers? | IBM. (n.d.). https://www.ibm.com/topics/naive-bayes

What is a Decision Tree? | IBM. (n.d.). https://www.ibm.com/topics/de#:~:text=Multicollinearity%20may%20not%20affect%20the,when%20it%20comes%20to%20interpretability.