Data Mining Techniques 1 - Group 139

Tianhao Xu[1,2[2740825]], Ruijia Lei[1,2[2733510]], and Rui Pan[1,2[2717917]]

Vrije Universiteit[1] and Universiteit Van Amsterdam[2]

1 Task 1: Explore a Dataset

Exploiting pivotal information and extracting unknown patterns from a large dataset, also known as a low-level knowledge discovery(KD) method—data mining[7], is a complicated process. Analysing a dataset consists of multiple stages including acquisition, wrangling, modelling and reporting [12]. The goal of the first task is to clean a small dataset—ODI-2022, then experiment with classification of dataset to explore the performance difference of varies classifiers. We choose `sklearn` library as our data mining tool to derive the implicit correlations and use matplot to visualise data.

1.1 Task 1A: Exploration

Overview of Dataset The ODI dataset is collected from a questionnaire during lecture in which 17 attributes coincide with 17 questions whose values are the distribution of answers polled by 304 students. All attributes are nominal type except the Tijdstempel, which belongs to data-time type. To get an overview of the entire dataset, we follow the process EDA to check the data quality and observe the logical structure. We first note data quality is poor owing to some nominal attributes existing missing value, inconsistency and error values, like Study Program, What Makes a Good Day, Birthday, Stress Level and Random Number, etc. Therefore, cleaning and standardising dataset are inevitable.

Pre-processing We choose python libraries `pandas` and `numpy` as feature engineering tools. For gender column, we only kept male and female and transformed the others to 'unknown' for convenience. For study programme column, we first converted it to lower case for better pattern matching latter, then we use regex to match and replace programmes which have different values but belong to the same class such as 'AI' and artificial intelligence'. For taken course column, we transformed them to yes, no and unknown. For date of birthday, we used regex to match keep valid date that includes month and day, and saved in the format 'MM-DD'. For stress level and random number, we ignored instances that is out of range. For answer of "the time you want to be yesterday", we transformed the format of time to numerical type ranging from 1 to 12 and renamed to 'Bedtime'. For those values don't fulfill the above criterion, we replaced them with `NA` for removing. After the feature engineering, the dataset remains 238 rows and 17 columns.

Statistics of Dataset We first use pie graph to investigate statistics of categorical attributes. For the study programmes, students are mainly from artificial intelligence(35.3%), business analytics(29.4%), computational science(13.4%) and computer science(7.6%). The result is shown in Figure 2.

We note that over 60% of students have taken machine learning, information retrieval and statistics, while only half have taken databases. And 61.8% of students in this course are male, 33.6% are female. We also noticed that 78.6% of the students were sitting in the class. Students average stress level is 44.8, we discretized the attribute into 3 groups—[0, 33], [33, 66], [66, 100], then we notice that 40.9% of students have mild stress, 30% have moderate stress and 29% have high stress. Finally, 26.6% of students believe sunny weather can make for a good day, 11.4% voted for good foods and 6% voted for sleep. We tried to explore the correlation between gender, stress level and bedtime, the result can be found in Figure 1. We observe that no significant linear correlation exists, most people sleep very late. And We note that sleeping late could cause higher stress level.

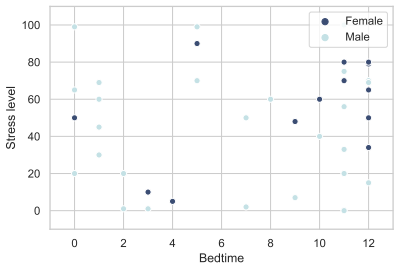

Fig. 1. Correlation of Gender, Stress Level and Bedtime

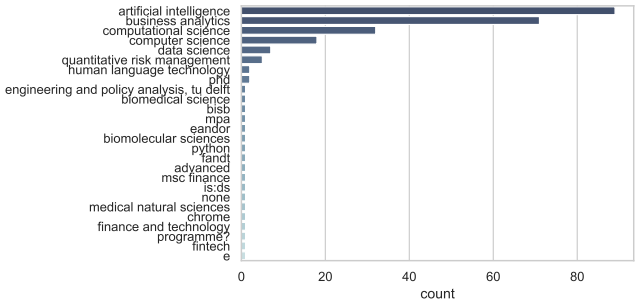

Fig. 2. Counts of Programme in Each Categorical Bin Using Bars

1.2 Task 1B: Basic Classification

We conducted several classification experiments to predict gender by stress level and bedtime to compare which of the three classifiers: *K nearest neighbour classification, Neural network, Random forest* yield the highest prediction accuracy.

We first labelled `gender` as target attributes because it only has yes or no, which is easy to classify, and we split the 80% of dataset into training set and 20% into test set. We then use `StandardScaler` to scale stress level and bedtime for better performance and accuracy. Before training k-NN model, we adopted `GridSearchCV`, which tests all combination of hyperparameters and compute their scores, to tune the best hyperparameters. For k-NN classifier, which only has one hyperparameter—k-value, we tested k value from 1 to 40. The best score we got is 0.653 when k=23. To verify this output, we calculated the error rate of k-NN for each k value, the result is shown in Figure 3. We can observe that when k surrounds 23, the error rate is minimized. Hence we can safely adopt this value.

We then used 5-fold cross-validation to train k-NN model with k value 23. Each iteration of cross-validation involves predicting 244/5 = 48 response values, and 4/5 dataset will be used for training and 1/5 used for evaluation. And the mean of cross-validation scores is 0.636, which indicates the model could correctly predict 30 cases out of 48, which isn't a robust result and implies that a lot noise exist in dataset. We then analyse the performance of random forest classifier, which has more adjustable hyperparameters: n_estimators, max_features, max_depth and criterion. We got the best parameters: `'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'n_estimators': 200` through grid search. And the mean of cross-validation score is 0.540 and the accuracy is 0.674, which is worsen than k-NN classifier. Finally we analysed the performance of neural network. Due to the limited number of features, we created a 2 layer neural network with 2 units for each layer. The hyperparameter 'solver' is set to '*lbfgs*', which is an optimization algorithm and is great with small data sets[5]. And the cross-validation score is 0.645. In conclusion, we tested the dataset with three different classification algorithms to predict gender by stress and bedtime. By comparing the cross-validation score, we concluded that the best accuracy is Neural network, followed by k-NN, random forest is the worst. The result is shown in Table 1. We think the reason behind the differences in outcome of 3 classifiers is neural network training involves more hyperparameters affecting the

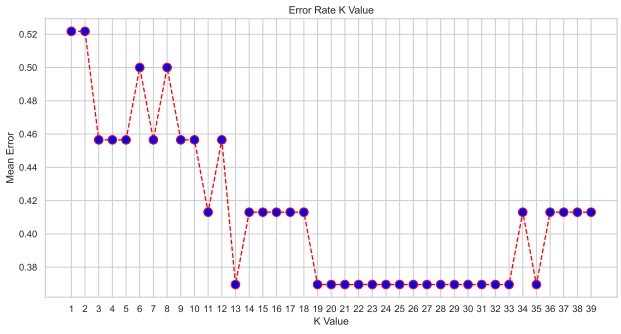

Fig. 3. Error Rate of k-NN Classifier at Different K-Value

size and structure of the network and the optimization procedure, which could produce better performance, while k-NN only has 1 hyperparameter. And once a neural network is trained, its paramaters are a good initializer for the similar task, which can hardly be achieved by k-NN.

Table 1. Accuracy and Cross-Validation Score of Classifiers

Classifier	Accuracy	Cross-Validation Score
k-NN	0.500	0.636
Random Forest	0.674	0.540
Neural Network	0.630	0.645

2 Task 2

2.1 Task 2A: Preparation

Overview of Dataset Importing the data from the Titanic training and test sets, we chose to work with both datasets for feature engineering. By looking at the composition of the data, it can be seen that the dataset has 12 features, PClass, Name, Ticket, Cabin and Embarked are nominal types, Age, SibSp, Parch, Fare and PassengerId are numerical type. The dataset has 891 and 428 rows for the training and testing test. The data type are int64, object and float64. From the correlation matrix as Figure 4, we can note that the 'Embarked', 'Sex', 'Parch', 'Fare', 'Cabin' are important features for 'Survived'. We also noted that feature Cabin has 687 missing values, "Age" and "Embarked" have 177 and 2 missing values separately. Age and Cabin have more missing values and require special handling.

	PassengerId	Pclass	Name	Embarked	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Survived
PassengerId	1.000000	-0.038354	0.998910	0.041304	-0.013406	0.028814	-0.055224	0.008942	0.707211	0.031428	0.179186	-0.005007
Pclass	-0.038354	1.000000	-0.039726	0.042356	-0.124617	-0.408106	0.060832	0.018322	0.038907	-0.558629	-0.602418	-0.338481
Name	0.998910	-0.039726	1.000000	0.037230	-0.014818	0.028658	-0.054473	0.009663	0.705570	0.032210	0.179975	-0.005007
Embarked	0.041304	0.042356	0.037230	1.000000	0.116904	0.042363	-0.072110	-0.094181	0.053201	0.058826	0.051640	0.101849
Sex	-0.013406	-0.124617	-0.014818	0.116904	1.000000	-0.063645	0.109609	0.213125	-0.126595	0.185523	0.074633	0.543351
Age	0.028814	-0.408106	0.028658	0.042363	-0.063645	1.000000	-0.243699	-0.150917	0.081549	0.178740	0.272991	-0.077221
SibSp	-0.055224	0.060832	-0.054473	-0.072110	0.109609	-0.243699	1.000000	0.373587	-0.309268	0.160238	-0.034496	-0.035322
Parch	0.008942	0.018322	0.009663	-0.094181	0.213125	-0.150917	0.373587	1.000000	-0.294080	0.221539	-0.022196	0.081629
Ticket	0.707211	0.038907	0.705570	0.053201	-0.126595	0.081549	-0.309268	-0.294080	1.000000	-0.207180	0.130462	-0.047298
Fare	0.031428	-0.558629	0.032210	0.058826	0.185523	0.178740	0.160238	0.221539	-0.207180	1.000000	0.408581	0.257307
Cabin	0.179186	-0.602418	0.179975	0.051640	0.074633	0.272991	-0.034496	-0.022196	0.130462	0.408581	1.000000	0.270495
Survived	-0.005007	-0.338481	-0.005007	0.101849	0.543351	-0.077221	-0.035322	0.081629	-0.047298	0.257307	0.270495	1.000000

Fig. 4. The Correlation Matrix of Different Features

Data Analysis Statistics and plots were used to gain a preliminary understanding of the correlations between the data, in preparation for constructing feature engineering and model building. Ratio of deaths to survivals: The number of deadth(0) is 549. The number of survived(1) is 342. The relationships between Plcass, SibSp, Parch, Sex, Embarked, Age and Survived are shown in Figure 5. We observed that female survivors account for nearly 70% of total survivors. The age distribution is quite normal, but we can note that age 28 has the highest survival rate, thereby we can assume that the younger passengers are easier to survive. We also can notice that people traveling with less than 2 siblings have more than 40% chance of survive. Moreover, the higher ticket class, the higher the survival rate.

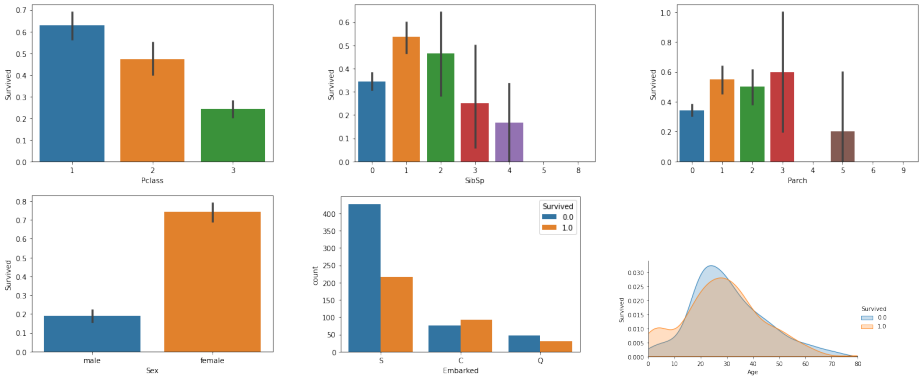

Fig. 5. The Relationship Between Different Features and 'Survived'

Pre-processing

Name Social rank has a very clear impact on our survival rate, so we can replace the name with the appellation prefix to generate a new feature 'Title', which is used to correspond to our social rank. In this way we have transformed a large list of useless name features into sub-typed features. We observed the survival rate of Mrs, Miss and Master, Royalty, representing the upper echelons of society, is much higher than that of Mr, representing men, and Officer, representing the middle and lower echelons of society.

Family The two features 'SibSp' and 'Parch' were combined into 'Family'. At the same time we grouped the parts with similar survival rates into one category and transformed the 'Family' feature into a subtype feature as figure Figure 6. This is because people with different numbers of family members do not have the same survival rate. This allows us to look more intuitively at the effect of characteristics on survival rates. We noticed the passengers with 2-4 family members has a higher survival rate.

Cabin To deal with hatch data with a large number of missing values, we need to process the features into sub-typed features. The missing values are grouped into one category and the remaining identified values are then classified. The missing values are all filled with 'U' and the identified hatch number is extracted with its initials as its new feature deck number 'Board'.

Ticket We find that ticket numbers are not individually unique values and that multiple passengers can have the same ticket number at the same time. We suspect that this may be related to our 'Family' data, possibly because the same family shares a single ticket number. This would allow us to classify features based on the number of passengers with the same ticket number for a given ticket as figure Figure 6.

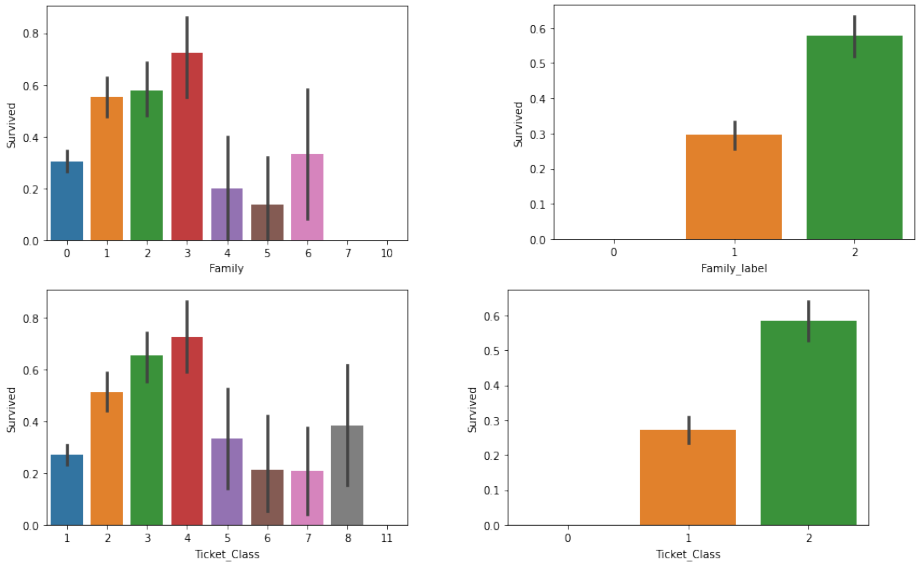

Fig. 6. Comparision of Attributes Before and After Discretize. Up Left Corner is Survival Rate at Different Family Size, Up Right Corner is Survival Rate at Different Family Size Group, Lower Left is Survival Rate at Different Ticket Class, Lower Right is Survival Rate at Different Ticket Class Groups

Embarked The Embarked missing amount is 2. Passengers with missing Embarked information all have the Pclass of 1 and the Fare of 80. Since the median Fare of passengers with Embarked of C and the Pclass of 1 is 80, so the missing value is filled with C.

Fare The amount of Fare missing is 1. The Embarked of passengers with missing Fare information is S and Pclass is 3, so the median Fare of passengers with Embarked S and Pclass is 3 is used to fill in.

Age Since there are so many missing values for 'Age', it is not reasonable to use the plural or mean alone to fill in the missing values. Moreover, there are many characteristics associated with age, such as social class, gender, name, etc., so we construct a random forest model with Sex, Title, Pclass and fill in the missing age values.

2.2 Task 2B: Classification and Evaluation

Construction of training and test sets In the process of feature engineering, a large number of features are generated, and there is a certain correlation between features and features. Too many features can, on the one hand, affect the speed of training and, on the other hand, may make the model over-fit. So we selected the features 'Survived', 'Pclass', 'Sex', 'Age', 'Fare', 'Embarked', 'Title', 'Family', 'Deck', 'TicketGroup' and converted them into numerical variables to reclassify the training and test sets. For the training set, we trained the `Random Forest` and `Decision tree` using the 5-fold stratified cross validation for our models. Then we took the average of the results as the final grade. The optimal parameters were selected automatically using a grid search. For Random Forest, the optimal parameters we obtained using the grid search were n_estimators = 25 and max_depth = 6. For Decision tree, the optimal parameters were n_estimators = 24 and max_depth = 5.

Application and Evaluation of models We chose accuracy as the main metric to measure how well the Titanic survived. In addition to accuracy, we chose metrics such as accuracy score, recall score and F1 score, which are all good for doing classification problems. The results were saved as predict_test.csv and uploaded to Kaggle to score the model. For the random forest model, the corresponding Kaggle score is 0.78947 which is higher than that for the decision tree. For the decision tree model, the corresponding score is 0.76794.

Expectations The results did not meet our expectations. As we feature engineer, more and more features are generated, and training the model with a large number of features will make our training set fit better and better. But at the same time there may also be a gradual loss of generalisation and overfitting may occur. Of course the model we build may not only perform poorly on the prediction set but also on the training set, and be in an under-fitted state. Moreover, we can improve in the following aspects in the future: for feature engineering, we can find better features and deleting redundant features with high impact; for model hyperparameter debugging, we can improve the underfitting or overfitting state; for model framework, we can make better choices of models for each layer of the framework.

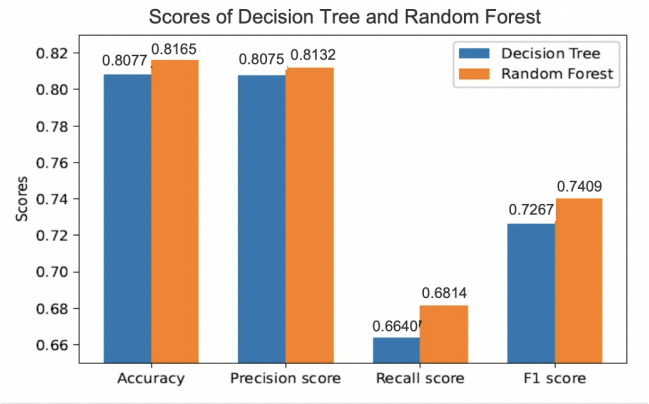

Fig. 7. Scores of Decision Tree and Random Forest

3 Task 3: Research and Theory

3.1 Task 3A: State of Art Solution

The competition title is Toxic Comment Classification Challenge and was held on March 13, 2018. The abuse and harassment online caused many people to stop expressing themselves and platforms were struggling even limit or shut down user comments. The Conversation AI team was researching models that can categorize toxic comments accurately, their models sometimes made mistakes at that time. The task is to build a better multi-headed model which is capable to detect the six possible types of comment toxicity - toxic, severetoxic, obscene, threat, insult, and identityhate. In addition, they were using mean column-wise ROC AUC(the score is the average of the individual AUCs of each predicted column) for evaluation.

The winner is Chun Ming Lee from Singapore working in H2O.ai. There are mainly used techniques, RNN model, pre-trained embeddings, test-time augmentation (TTA), rough-bore pseudo-labelling (PL), and applying cross-validation to add stacking framework.

Focusing on the more complex embedding layers, the techniques of FastText[2] and Glove[9] are used to obtain high-dimensional word vectors by relying on publicly available scientific datasets such as CommonCrawl, Wikipedia and Twitter for pre-training. For the post-embedding layers, the authors feed two BiGRUs[10] (bidirectional sequence processing model) into the dense layer. The authors also augmented the existing dataset and test set, using Test Time Augmentation[8] method which translates comments in languages such as French and Spanish into English. The authors also used a pseudo-labelling method to improve the performance of the model. Moreover, he considered taking a weighted mean of arithmetic averaging and stacking[4] which used primarily LightGBM(faster than XGBoost) and reached slightly better CV scores.

The points that make the approach the most competitive are:

1. Pseudo-labelling method obtains more accurate decision boundaries to improve the robustness of the model.
2. TTA translation method which leverages Pavel Ostyakov's idea of machine translations to increase the training data to make the model more powerful.
3. The author tried some PL variants like canonical per-batch updates and altering the loss functions etc. The best performing samples are then added to the training set for training to converge.
4. Ensembling Learning by stacking method, it also prevents overfitting by conducting parameter adjustment which uses small trees with low depth and strong l1 regularization. It bags 6 runs of DART and GBDT adopting different seeds to consider potential variance during stacking.

3.2 Task 3B: MSE Versus MAE

The formulae of MSE (mean squared error) and MAE (mean absolute error) are as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \qquad MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \tag{1}$$

The $\hat{y}$ is the predicted value of y

Regression models rely on distance metrics to determine the convergence performance. MAE is probably better than MSE in most cases when those two are only considered because MAE is less affected by outliers, MSE is equivalent to the expected value of the squared error loss or quadratic loss. The difference occurs because of the randomness. Moreover, according to the data source, MAE will help ignore the outlier.

Considering that the result of MSE comes from the power of two, the result of MAE is purely the absolute value of the difference. We describe a dataset contains only values of 0,1, such as coin toss which are independent probability events of Bernoulli distribution(binomial distribution with n=1). All the prediction errors (i.e. residuals) are exactly ± 1 which will lead the identical results of MSE and MAE.

Diamond Dataset(A dataset from R containing the prices and other attributes of almost 54,000 diamonds.)[1] The reason of choosing this dataset is because it is an officially published set of scientific data which contains more than 54000 records. With a large amount of data as the basis, the MAE and MSE calculated based on regression models are more convincing.

We conduct experiment with regression models to predict the price of a diamond based on carat(weight) and generate the result by calculating MSE and MAE. For example, the results of using linear regression model is 2397955 and 1007.463. Regarding the results, if we square MAE, it is smaller than MSE. In another word, the RMSE(square root of MSE) is bigger than MAE which means MSE may be less robust than MAE, since the squaring of the errors will enforce a higher importance on outliers. It can also be concluded that the reason for their difference is that the volume of data itself is large and the presence of outliers has a sizable impact on the MSE which is more sensetive.

3.3 Task 3C: Analyze a Less Obvious Dataset

Data are characterized by labels and informational content only. After researching, since the data are labeled, we think there are some supervised and semi-supervised models can be applied. Support Vector Machine(SVM) is one of the suitable models. One very good fit is that the data itself is classified into two types which matches the SVM requirements quite well. The SVM is also appropriate for being a regression or classification model for text where the number of features is larger than the number of samples. SVM maps training data to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that identical space and predicted to pertain to a category based on which side they fall.

We adopt data smoothing to eliminate or reduce any discrepancies or any other noise to highlight the important features of the data. For example, eliminating some stop words and various meaningless characters and symbols. Moreover, on the basis of the hypothetical SVM as a model, data normalization is also a possible operation because it uses features for similarity or distance calculation and normalization to a certain range allows better "clustering" of outliers in a high-dimensional space.

Applying the SVM model and using accuracy and precision score to check the performance. After pre-processing data which needs to eliminate misspelled words, induced words and stop wordsand encoding them with a label encoder, we take TF-IDF which reflects the importance of word to do vectorization, and finally utilize the SVM model to train and test. After that, we find the accuracy is about 84.3% which is receptible and still has room for improvement. Then we conduct a contrast experiment that uses Naive Bayes for classification and the result is 83.1% which is less effective than SVM. To make betterment, we can replace old-fashioned feature engineering methods(TF-IDF) with word embedding techniques like Word2Vec[6] which has more dimensions, and let it be used at the embedding layer by a more powerful deep learning model LSTM[11], or utilize BERT[3] model for prediction, it can be improved via the combination of state-of-the-art techniques during feature extraction and model training, etc. Moreover, converting classification task to regression task(mapping classes into different numerical value) and utilizing regression models like Logistic Regression is also worth a try. More extensively, we consider taking different methods to change the distribution of the original training samples to build several different classifiers and linearly combine these classifiers to obtain a more powerful classifier and the approach is called Ensemble Learning to improve the performance. On the other hand, we think introducing the aid of emotion analysis to categorize might have made the results better.

References

1. Prices of over 50,000 round cut diamonds. https://ggplot2.tidyverse.org/reference/diamonds.html. Accessed: 2022-04-22.
2. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.
3. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
4. Jie Dou, Ali P Yunus, Dieu Tien Bui, Abdelaziz Merghadi, Mehebub Sahana, Zhongfan Zhu, Chi-Wen Chen, Zheng Han, and Binh Thai Pham. Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, japan. *Landslides*, 17(3):641–658, 2020.
5. ZH Fu. Comparison of gradient descent stochastic gradient descent and l-bfgs, 2016.
6. Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.
7. Lukasz A Kurgan and Petr Musilek. A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*, 21(1):1–24, 2006.
8. Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation. In *Conference on Uncertainty in Artificial Intelligence*, pages 1308–1317. PMLR, 2020.
9. Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
10. Rajib Rana. Gated recurrent unit (gru) for emotion classification from noisy speech. *arXiv preprint arXiv:1612.07778*, 2016.
11. Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.
12. Kanit Wongsuphasawat, Yang Liu, and Jeffrey Heer. Goals, process, and challenges of exploratory data analysis: an interview study. *arXiv preprint arXiv:1911.00568*, 2019.