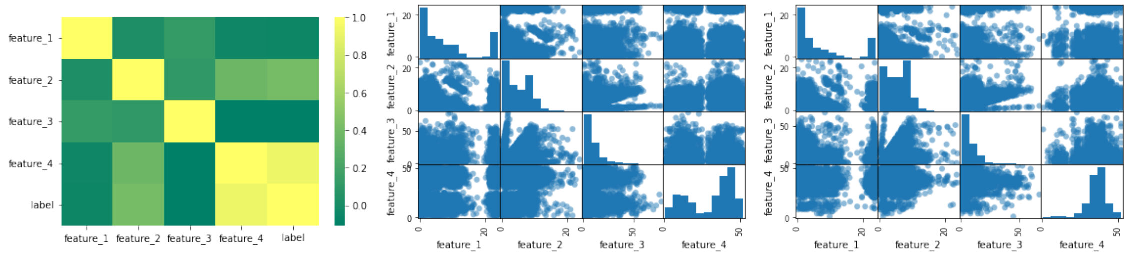
MINI-PROJECT REPORT

Author: Huihan YANG;   Student ID: 120090438

Github repo-link: https://github.com/foxintohumanbeing/cuhksz_dda4210_miniproject.git

- Report focuses on the data from training and augmented data. The discussion of tesing data will be in the last section.
- I submit two prediction result on bb. If only one is permitted, please use the "**vb_120090438.csv**" to test.

1. Data Analysis



Left:Correlation Heatmap of Training Data.Center:Feature Histogrm of Training Data.Right: Feature Histogrm of Augmented Data

- The heatmap of training data shows that feature 4 is the most correlated feature with classification results.
- The histograms of train and augmented data show that they have similar pattern (mainly distributed around some values), which makes it reasonable to use good model from train data on the augmented data .

2. Binary Classification Algorithm

2.1 AlgorithmA: Majority vote of six small classifiers

```
Input: One data
Output: Label of data
label$_1$, $label_2, label_3, label_4, label_5, label_6 \leftarrow six\ small\ classifiers$
label $\leftarrow majority\ vote(label_1 : label_6)$
return label
```

- Other trivial operations like train-valid split are omitted for simplicity.

2.1.1 Small Classifiers and Their Weights:

Classifier_name	Random Forest1	Random Forest2	SVC1	SVC2	Adaboosting	Naive Bayes
Weight	3	1	1	1	1	1

2.1.2 Majority Vote

Denote the output label given by classifier j for a sample i is l_{ij}, corresponding weight is w_{ij} and the final prediction is l_i.

$$l_i = l_{ia}, \qquad where\ a = \arg\max_{a\in\{1,2,3,4,5\}} \Sigma_{j=1}^{6} w_j \mathbf{1}_{l_{ij}=l_{ia}}, \qquad \mathbf{1}_{l_{ij}=l_{ia}} = \begin{cases} 1, l_{ij} = l_{ia} \\ 0, l_{ij} \neq l_{ia} \end{cases}, \qquad \Sigma_{j=1}^{6} w_j = 1 \tag{1}$$

2.1.3 Key Components and Rationale

- I choose Random forest as major classifier. I ensemble other kinds of classifiers to modify the difficiency and reduce variance of RF.

1. Random Forest
   - **Key Algorithm**: Build many trees with limited number of features. Take majority to reduce variance.
   - $num_{fea_{max}}$ refers to the maximum number of feature used in each tree.
   - Random Forest1 - Parameter: num_{tree}=100, $num_{fea_{max}}$ = 2
     - **Role and Rationale**: **Major Classifier**

        - Feature4 is the most correlated factor. Select important factors and amplify their influences.
        - One random forest could reach the accuracy around 99%. However, it fails to reduce tree structure's bias. To fix it, I use other classifier to modify.
    - Random Forest2: - Parameter: num_{tree}=100, $num_{fea_{max}}$ = 3
        - **Role and Rationale**: Increase number of maximum number of features. Find more info missed by RF1.
2. SVC
    - **Key Algorithm**: Project data on higher dimension to seperate with penalty.
    - SVC1-Parameter: Kernel = 'rbf', C = 8, SVC2-Parameter: Kernel = 'poly', C = 2
        - **Role and Rationale**: Find more information of feature's correlation in high dimension. Reduce the influences of outliers using penalty term.
3. Adaboosting - Parameter: Nan
    - **Key Algorithm**: Sequentially build one tree with slow learning rate using adaptive gradient
    - **Role and Rationale**: Reduces tree structure‘s bias.
4. Naive Bayes - Parameter: Prior = Gaussian
    - **Key Algorithm**: Maximize feature's likelihood with features' prior assumption as Gaussian.
    - **Role and Rationale**: Helps to deal with unbalanced and limited size data.

2.2 AlgorithmB: Boosting SVM

2.2.1 Algorithm : use SVM classifier as base classifier. Develop it using Adam Boosting strategy.

2.2.2 Key Components and Rationale: Same SVC as described in 2.1.2-SVC1. Boosting strategy to sequentially reduce the bias from SVM structure.

3. Results and Analysis

3.1 Results

- For each model, I run 30 times to see the average accuracy and variance with different split of data(with shuffle).

	RF	Boosting	SVM	Gaussian Bayes	Vote-boosting	Ensemble-SVM
Mean	0.9922	0.9913	0.9882	0.9877	0.9933	0.9926
Variance	0.0345	0.0687	0.0728	0.0870	0.0253	0.0342

- To avoid the problem of underflow, I multiply the accuracy with 100 when calculating variance.
- Vote-boosting result is stored in "**vb_120090438.csv**" and ensemble-SVM result is stored in "**ev_120090438.csv**".

3.2 Analysis

- It is worth to note that both two models has highest mean accuracy and lowest accuracy variance.
- This result highlights two model's stability and generalization, which also gives confidence to use this model.
- We tried other ensemble models like **SVM-GMM** and **ensemble boosting**.
    - All of the ensemble models has relatively similar high performance on validation dataset.
    - The different percent of their prediction is in approximately [0.01%,4.55%].
    - It is hard to choose model to submit because the overfitting issue is vague. The training data lacks the data whoes feature4's range is in [23,29], which may leads to wrong prediction on augmented dataset(it has a number of data whose feature4's range is in [23,29])

4. Mention: Algorithm used on Kaggle

Because of test data's special data structure, we use SVM-GMM with only feature4 to achieve the accuracy of 98.75%. We train both GMM and SVM on training dataset. The fitted result of GMM is used to cluster the test data. The prediction result given by SVM is used to be the guidance to label the cluster.