Using machine learning methods to identify legendary Pokémon

Candidate Number: 046781

1 Abstract

This project is aiming to use a random forest classifier to identify legendary Pokémon. This dataset we used detailed the different features of 800 Pokémon, which came from The Official Pokémon Website[1], also use a single decision tree as a control experiment, and both methods perform very well in this classification problem.

2 Introduction

2.1 Contextualization: Pokémon

Pokémon is a series of cross-media productions, including games, animations, comics, card games, and related products. This series originally proposed by GAME FREAK and promoted and released by Nintendo in 1996. In the Pokémon world, the word "Pokémon" represents a group of special creatures like animals in the real world, and players are trainers of Pokémon, who capture and training Pokémon, also use their own Pokémon to participate in competitions. Different Pokémon has different characteristics which relate to whether they can be a winner in Pokémon competition. Until the last version, there are a total of 898 different Pokémon in the Pokémon series, 18 different types, and some Pokémon have two different types, and there are 9 numerical data in the official information to describe each Pokémon's different characteristics [2].

2.2 Motivation

In the Pokémon series, the main purpose of the player is to capture the powerful Pokémon, and then train them in many battles, defeat

the boss in each generation of the story, and eventually become the "Pokémon Master". Some special Pokémon are called "Legendary", they are usually the protagonists of a certain series and stronger than other "Normal" Pokémon. Consequently, trainers who identify and capture those "Legendary" Pokémon will have stronger combat effectiveness and be more chances of winning in the game(competitions). However, there are no explicit criteria to identify those special Pokémon, the only way is through statements from official information.

2.3 Project Aims

This project aims to base on Pokémon's 10 characteristics use a random forest algorithm to identify a Pokémon is legendary or not and analyze the importance of different characteristics to legendary or not. On the other hand, as a comparison, this project also uses the single decision tree algorithm to perform the same experiment and evaluate the performance of these two algorithms on this classification problem.

3 Method

Identify the legendary Pokémon is a typical binary classification problem, there are many machine learning algorithms that can be applied to this kind of problem. In the aspect of data, each Pokémon's characteristics are detailed in official documents and easily found on the website.

3.1 Data

The dataset we used comes from Kaggle which is a platform mainly for developers and data scientists to hold machine learning competitions, host databases, write and share code. [3] This dataset collected 800 Pokémon's information, in addition to name and ID, it contains 11 variables per Pokémon [4]:

- **Type_1:** The main type of Pokémon, this category value can take 18 different values: *Bug, Dark, Dragon, Electric, Fairy, Fighting, Fire, Flying,*

Ghost, Grass, Ground, Ice, Normal, Poison, Psychic, Rock, Steel, and Water.

- **Type_2:** Part of Pokémon have the second type and the possible values is same as Type_1.
- **Total:** A good indicator of the overall strength of a Pokémon, and it is the sum of the next six numerical variables.
- **HP:** Base health points of the Pokémon. The bigger it is, the longer the Pokémon will be able to stay in a fight before they faint and leave the combat.
- **Attack:** Base attack of the Pokémon. The bigger it is, the more damage its physical attacks will deal to the enemy Pokémon.
- **Defense:** Base defense of the Pokémon. The bigger it is, the less damage it will receive when being hit by a physical attack.
- **Sp_Atk:** Base special attack of the Pokémon. The bigger it is, the more damage its special attacks will deal to the enemy Pokémon.
- **Sp_Def:** Base special defense of the Pokémon. The bigger it is, the less damage it will receive when being hit by a special attack.
- **Speed:** Base speed of the Pokémon. The bigger it is, the more times the Pokémon will be able to attack to the enemy.
- **Generation:** The generation where the Pokémon was released. It is an integer between 1 and 6.
- **Legendary:** Boolean indicating whether the Pokémon is legendary or not, the target value of this problem.

3.2 Decision Trees

In machine learning, decision tree is a prediction model which represents the mapping relation between object features and object values. Decision tree model is supervised learning which can be used to solve classification and regression problems. Moreover, decision tree can be applied to ensemble learning such as random forest. A complete decision tree contains root node, non-leaf node, branches and leaf node. The root node is the first question for features, and non-leaf nodes are other questions, branches

represent test results and leaf nodes are classification mark after classification.

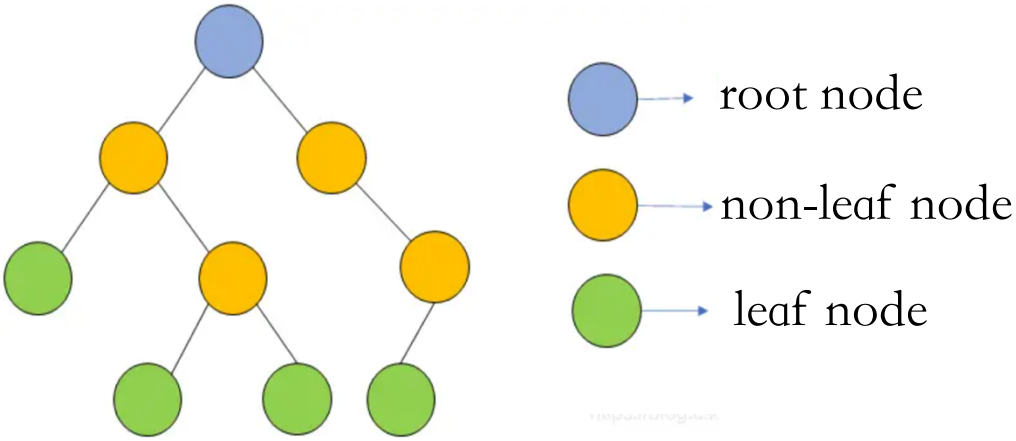


Fig1. Structure of decision tree model

Modeling the decision tree usually use the top-down approach, select the best feature to split in each step, different algorithms use different criterion to define "best", but the same aim is to make the train set of child node as pure as possible. There are usually three decision tree algorithms[5]:

- **ID3:** the classification criterion of ID3 is information gain, which represents the degree to which the uncertainly of the subset is reduced by known the information of feature A.

  Information entropy:

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|}$$

  Where C_k represents the subset of samples in the set D belonging to the k-th class.

  For feature A, the conditional entropy of dataset D H(D | A):

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i)$$
$$= -\sum_{i=1}^{n} \frac{|D_i|}{|D|} (\sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} log_2 \frac{|D_{ik}|}{|D_i|})$$

  Where D_i represents the subset of feature A in D takes the i-th value,

D_{ik} represents the subset of samples belonging to the k-th class in D_i.
Information gain = information entropy − conditional entropy:

$$Gain(D, A) = H(D) - H(D|A)$$

ID3 algorithm do not have pruning strategy, so it is easy to overfitting, and information gain criterion has a preference for features with a larger number of possible values.

- **C4.5:** The biggest feature of the C4.5 algorithm is that it introduces the information gain rate as the classification criterion, overcomes the shortcoming of ID3's emphasis on the number of features.
  Information gain rate:

$$Gain_{ratio}(D, A) = \frac{Gain(D, A)}{H_A(D)}$$

$$H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} log_2 \frac{|D_i|}{|D|}$$

  C4.5 algorithm has pruning (pre-pruning and post-pruning) strategy which can prevent decision tree model from overfitting.

- **CART:** the CART algorithm can solve both classification and regression problems, and it use the Gini index as the classification criterion, which simplifying the entropy model but retains its advantages. The Gini index represents the impurity of the model. The smaller the Gini index the better the features, this is the opposite of information gain rate. The Gini index:

$$Gini(D) = \sum_{k=1}^{K} \frac{|C_k|}{|D|}(1 - \frac{|C_k|}{|D|})$$

$$= 1 - \sum_{k=1}^{K} (\frac{|C_k|}{|D|})^2$$

$$Gini(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} Gini(D_i)$$

  Where the k represents the class.

3.3 Random forest

Random forest is an ensemble classifier which is constructed by combining many decision trees. There are three ensemble

algorithms: Bagging, Boosting and stacking, random forest is a representative model of bagging, and the decision tree is the base estimator of random forest [6]. The main idea of Bagging is to build multiple independent evaluators, and then use the average or majority voting principle to determine the result of the ensemble estimator.

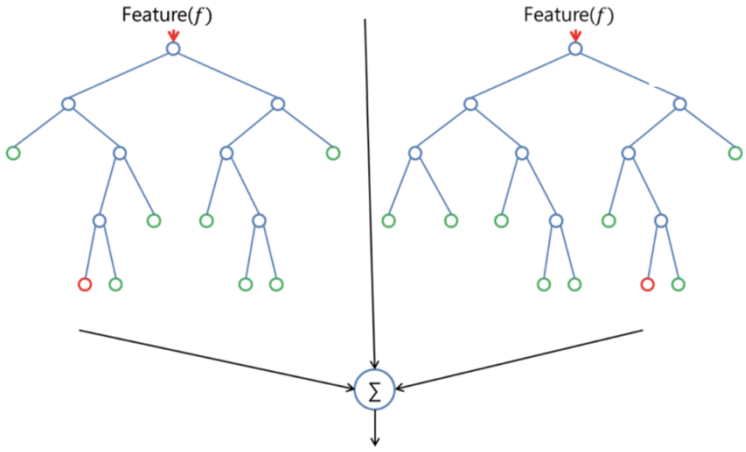


Fig2. Example random forest constructed from 2 trees

Since the random decision tress generation process uses the Boostrap, not all samples are used, the unused samples are about 37% on average and called out of bag (OOB) samples. The OOB score can be used to evaluate the accuracy of this tree, and the mean value of all trees is the accuracy of this random forest. Generally speaking, the OOB error is larger than the cross-validation error, because estimating OOB only use some of the trees in the random forest, instead of using the complete model, which limits the performance of the model. However, OBB eliminates the need of multiple training steps in cross-validation, so it is more efficient.

4 Experiments

This project uses the decision tree classifier and the random forest classifier which are provided by scikit-learn package to perform experiments and evaluate differences of these two models.

4.1 Explore the dataset

4.1.1 Missing value and label data encode

In this dataset we used, Type_1 and Type_2 are label data, Fig3 and Fig4 indicated the count of 18 different labels in Type_1 and Type_2. Most of Pokémon only has Type_1, so there are 386 missing values in Type_2 column, we filled these missing values as "NONE". Since both decision tree and random forest algorithm cannot operate label data, we will use one-hot encoding to process these label data into binary form before training.

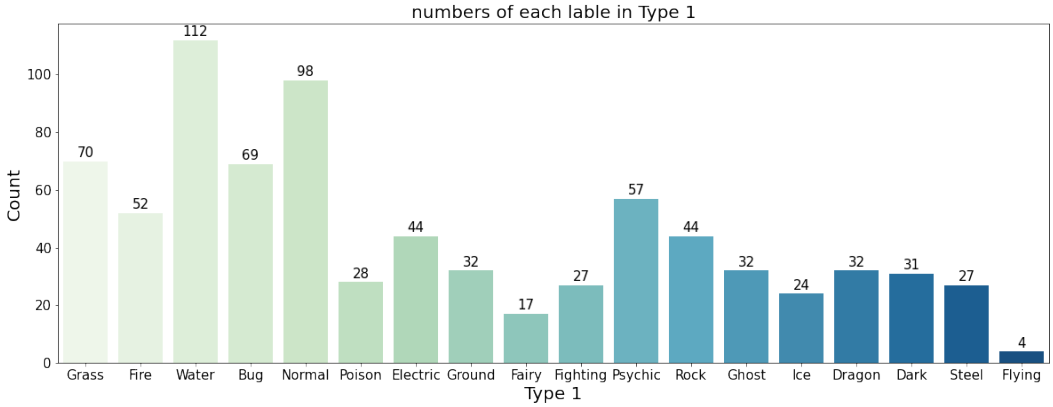


Fig3. Count of different labels in Type 1

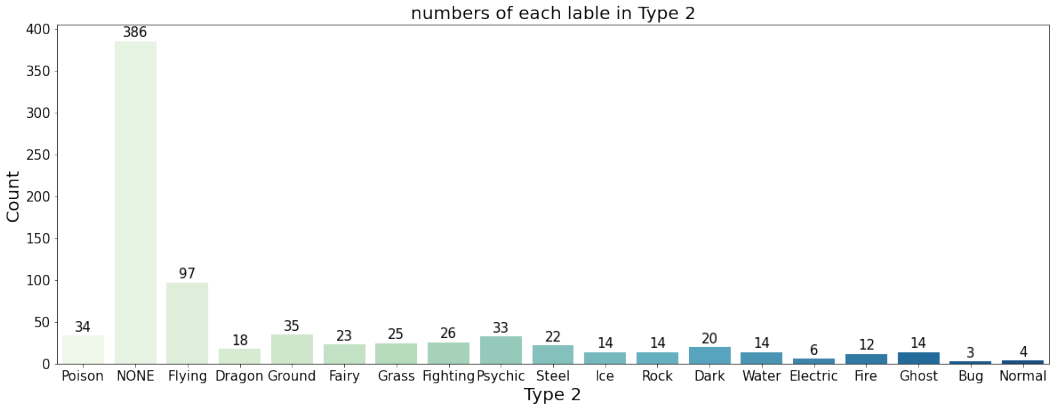


Fig4. Count of different labels in Type 2

4.1.2 Numerical data

There are 8 features (Total, HP, Attack, Defense, Sp_Atk, Sp_Def, Speed, Generation)of Pokémon are described in numerical form,

Fig5 illustrated distribution density of each numerical data, Fig6 is the visualized correlation coefficient matrix of these data.

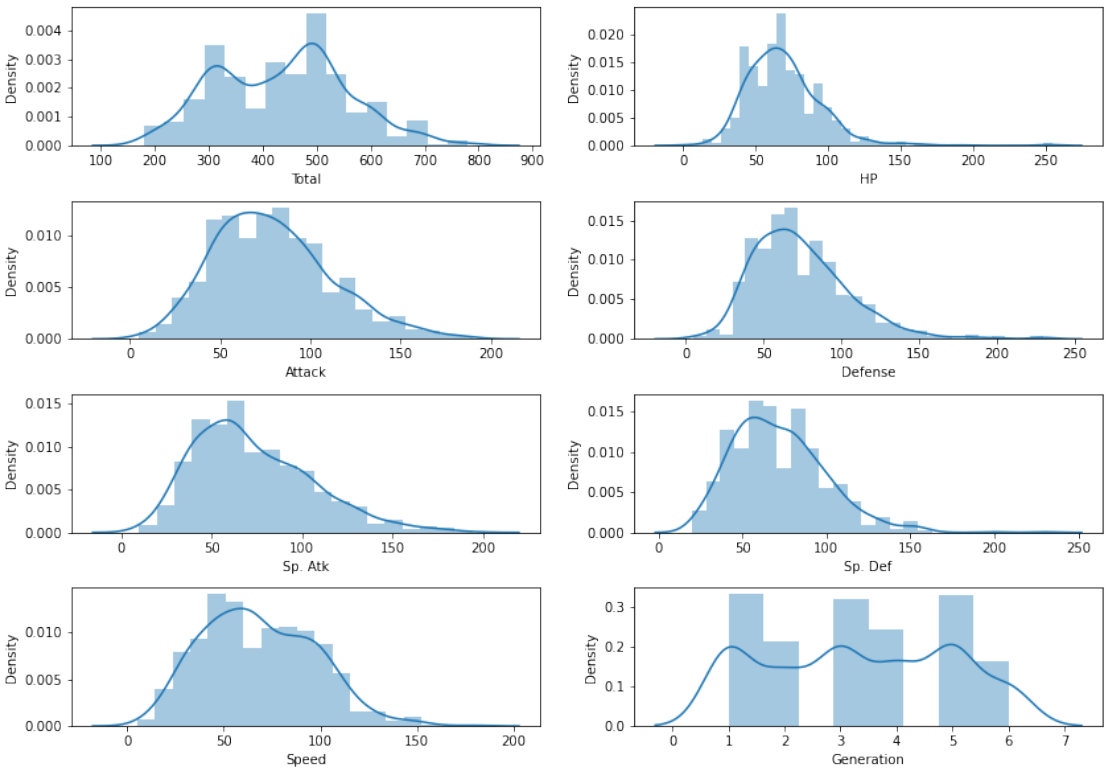


Fig5. Distribution density

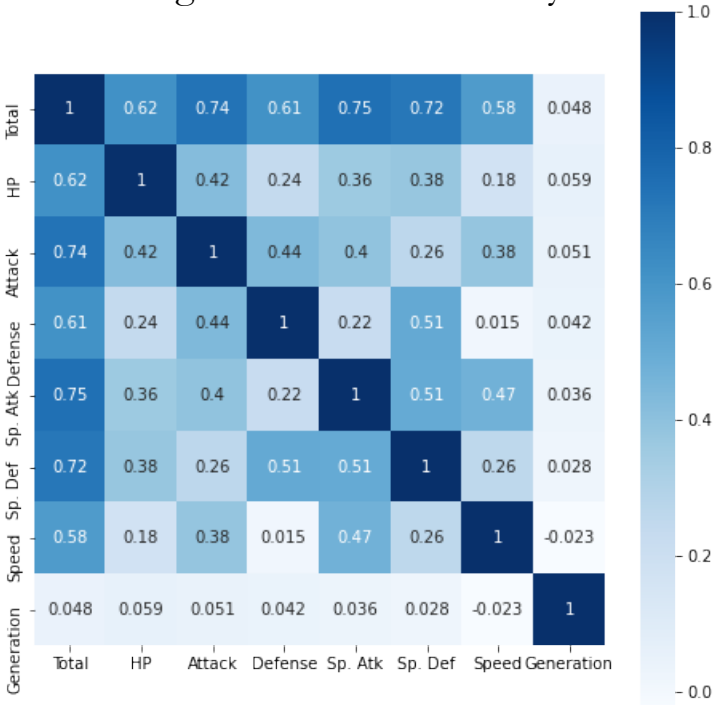


Fig6. Correlation

4.1.3 Imbalanced data

The "Legendary" Pokémon are very rare, Fig7 compared the numbers of "Legendary" Pokémon and normal Pokémon. As we can see in the figure, this dataset is very imbalanced, there are only 65 Pokémon in 800 samples that are "Legendary", only count 7% of the dataset. In order to eliminate the impact of data imbalance on model training, we use the SMOTE (Synthetic Minority Oversampling Technique) algorithm to over-sample the minority class (Legendary). As a result, we increased the number of legendary Pokémon, produced balanced training data. In order to eliminate the effect of creating the synthesized data to test the model, the oversampling method only applied to the training set, then use the trained model predicts in the test set and produce the score.

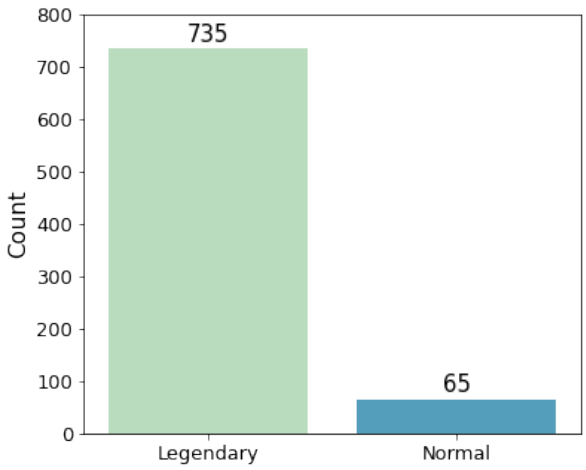


Fig7. The number of legendary and normal Pokémon

4.2 Random forest classifier

According to describe above, we build the random forest classifier to identity the legendary Pokémon. Firstly, splitting the dataset into training set and testing set at a ratio of 7 to 3, the SMOTE algorithm is applied to the training set and create a new training set which has the same numbers of legendary and normal. After we fit the random forest model in new training set and test it in testing set.

4.2.1 Tune the hyperparameters of RF

Before evaluating the performance of random forest model, we use SMOTE in training set and produce the accuracy score in testing set to tune hyperparameters (n_estimators, max_depth, max_features, min_samples_split). The result is the best hyperparameters combination is n_estimators=95, max_depth=14, max_features=6, min_samples_split=5.

4.2.2 Experiment result of RF

The accuracy of test set of this model is 94.167%, Fig8 shows the confusion matrix for the test set. Its OOB score in relatively higher which is 98.07%, but the OOB score cannot be used as a criterion for evaluation because it is based on the new train set which is processed by SMOTE and influenced by effect of create the synthesized data. So, we use 10-fold cross validation to produce a score which and can better reflect the performance of the model. In each fold of cross validation, we use SMOTE in training set and produce accuracy score in test set, finally use the mean score of 10 folds as the accuracy of the model. which is 93.375%, a little low than before, but more precise.

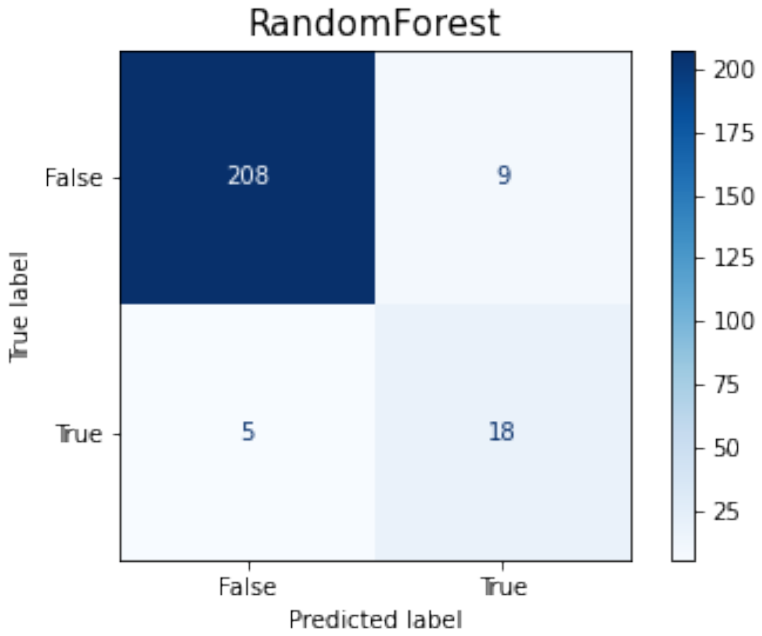
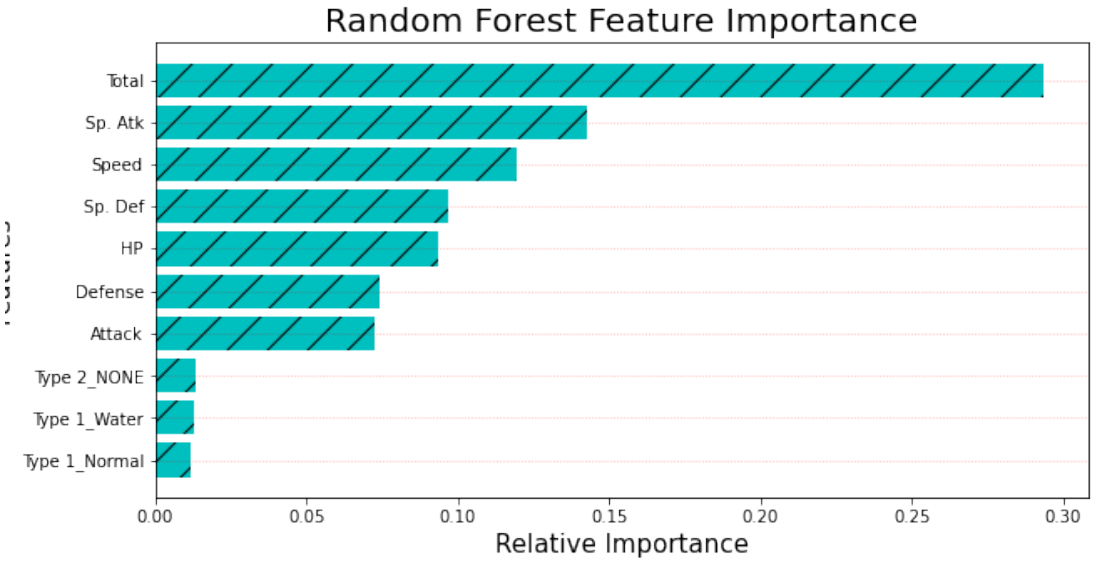


Fig8. Confusion matrix of RF

Using the feature_importance provided in Sk-learn, we can see which features have an important influence on whether a Pokémon is legendary, which is measured by how much the entropy of the feature is reduced in the model on average and sum of all importances is equal to 1, so we can use percent to describe each importance. Fig9 shows top 10 important features, we can see the "Total" is most important feature, which importance is about 29.36%. The next features are "Sp_Atk", "Speed", "Sp_Def", "HP", "Defense" and "Attack", importance of each different types almost the same.



Total	Sp. Atk	Speed	Sp. Def	HP	Defense	Attack	Type 2_NONE	Type 1_Water	Type 1_Normal
0.293622	0.142584	0.119388	0.0965012	0.0935114	0.0738401	0.0721877	0.0130434	0.0124685	0.0113544

Fig9. Top 10 importance

4.3 Decision tree classifier

We choose a single decision tree as the control group, use the same split training set and testing set, applied SMOTE to training set, use accuracy score which produced by test set to tune hyperparameters (max_features and max_depth).

Finally, the best parameters combination of decision tree model is max_features=7, max_depth=15. The accuracy of this model is 92.5%. Fig10 is the confusion matrix for the model on the test set.

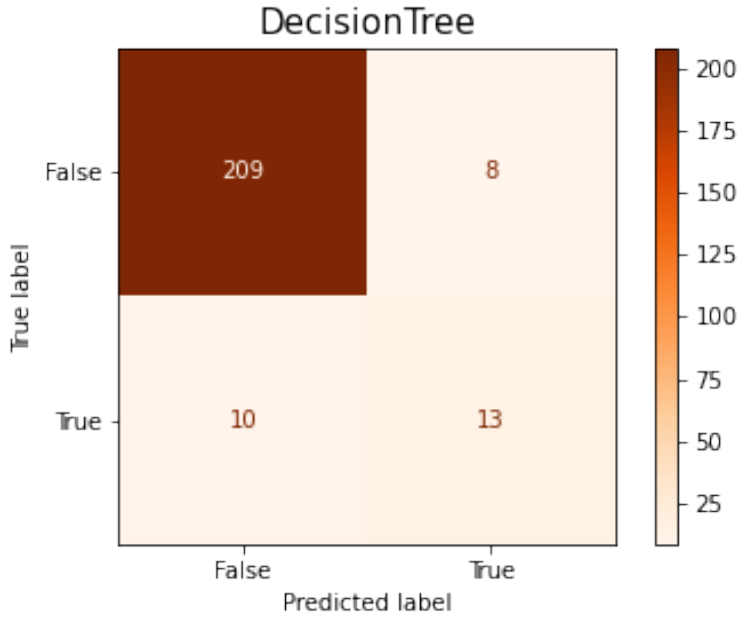
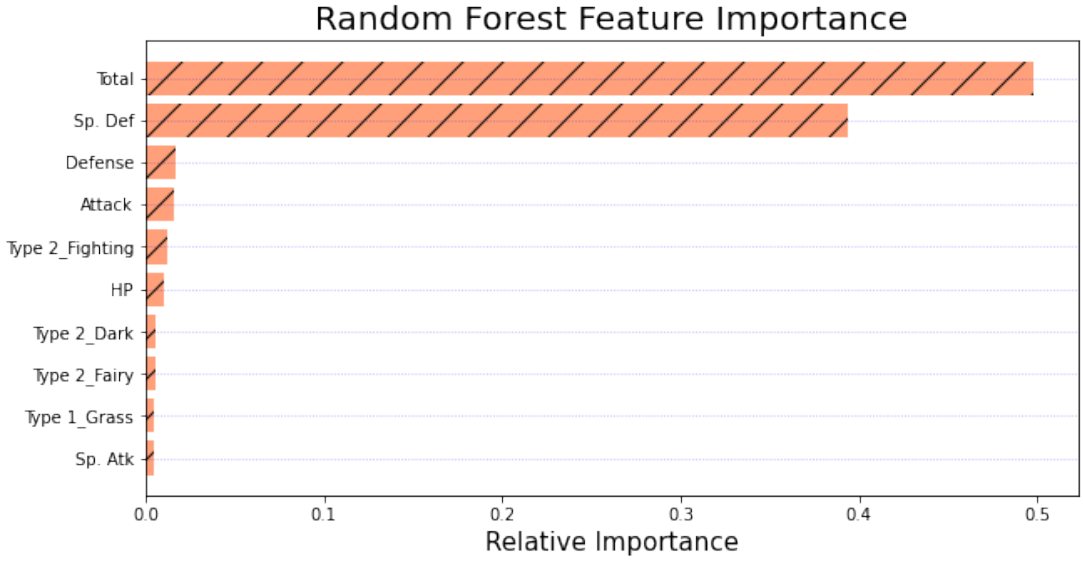


Fig9. Confusion matrix of DT

As same as the random forest model, we also use 10-fold cross validation with the SMOTE in the decision tree model, the mean accuracy of cross validation is 91.25%, lower than random forest model.



Total	Sp. Def	Defense	Attack	Type 2_Fighting	HP	Type 2_Dark	Type 2_Fairy	Type 1_Grass	Sp. Atk
0.498213	0.393685	0.0167672	0.0154918	0.0126044	0.00988179	0.00573138	0.00536268	0.00477795	0.0045939

Fig10. Top 10 importance

In the term of feature importance that is shown in Fig10, although "Total" still the most important feature, the second important feature changes to "Sp_Def", these top 2 importances is 49.82% and 39.37% respectively. Interestingly, except for the two features mentioned above, the importances of other features are very low and similar, which is completely different from the random forest model.

4.5 Compare random forest and decision tree

According to above discussion we know that the mean accuracy of random forest is relative higher than decision tree. Fig11 illustrates the accuracy of each fold in cross validation of these two models. Obviously, the random forest model gets the higher score in each fold than decision tree, the performance of two models in each fold shows the same change tendency.

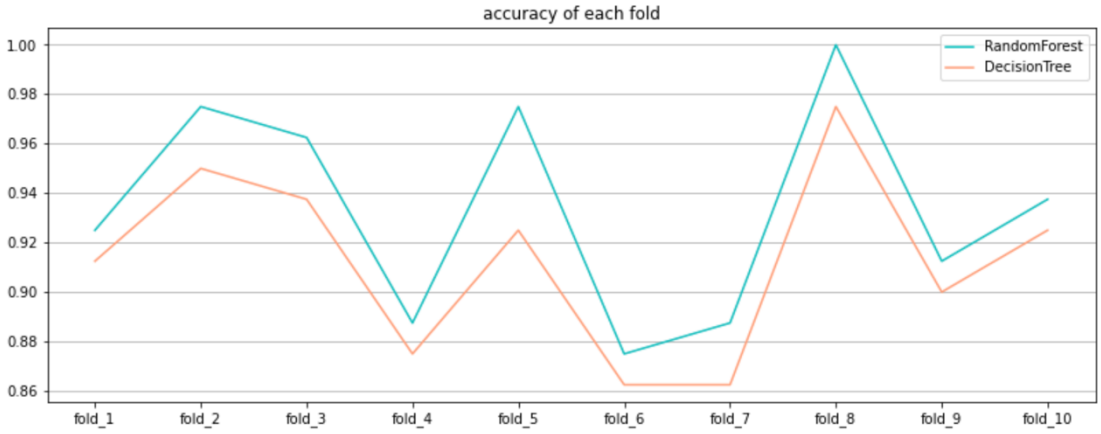


Fig11. 10-fold cross validation

Fig12 compares two model's 10-fold accuracy in the box plot. It can be clearly seen from the box plot that the five important points (Minimun, Maximun, Median, First quartile, Third quartile) of the random forest model are higher than those of the decision tree model, but the gap between the first quantile and the third quantile is also greater, indicating that the data has changed more. In general, the two sets of data have no outliers, and the random forest performs better.

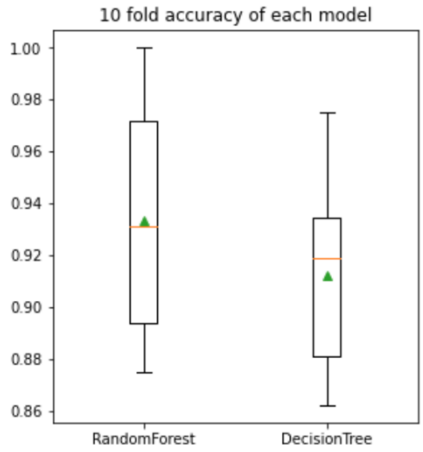


Fig12. Box plot of 10-fold cross validation

In order to further evaluate and compare the performance of the two models, we also use ROC (Receiver Operating Characteristic) curve and AUC (Area under Curve) [7].

The x-axis of the ROC curve is FPR (False positive rate), and the y-axis is TPR (True positive rate). A set of FPR and TPR is obtained by setting different thresholds to draw the ROC curve. In Sk-learn package, we can use ".predict_proba" produce the probability of each sample being classified as each label by the classifier, in this project, we can get the probability that each Pokémon is predicted to be "Normal" or "Legendary" by classifier. Then we use "roc_curve" which also provided by Sk-learn, by true values and probability of predicted to be positive, we can get a set of FPR, TPR and thresholds.

AUC represents the area under the ROC curve, which is mainly used to measure the generalization performance of the model. AUC indicates the probability that a positive example is ranked before a negative example, so the value of AUC ranges between 0 and 1, closer to 1, the better. The reason why AUC is used to evaluate is mainly because the ROC curve itself cannot intuitively explain the performance of a classifier, and the AUC value as a quantitative value is comparable and can be quantitatively compared. Use ".roc_auc" provided by Sk-learn, we can easily get AUC.

Fig13 is ROC curves of random forest model and decision tree model, the ideal target for the model to draw the ROC curve: TPR = 1, FPR = 0, that is, the point (0,1) in the figure, so the closer the ROC curve is to the point (0,1), the deviation from the 45-degree diagonal the larger the better, the greater the sensitivity and specificity, the better, this also means that the larger the AUC.

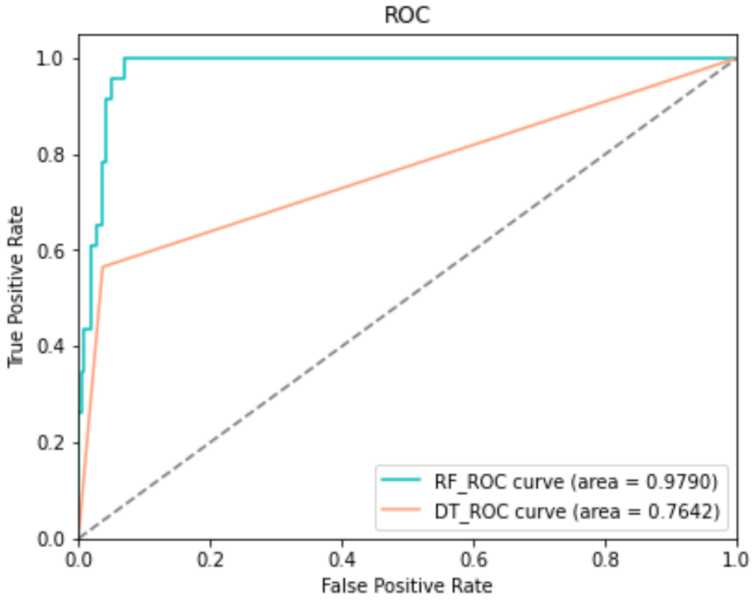


Fig13. ROC curve of two models

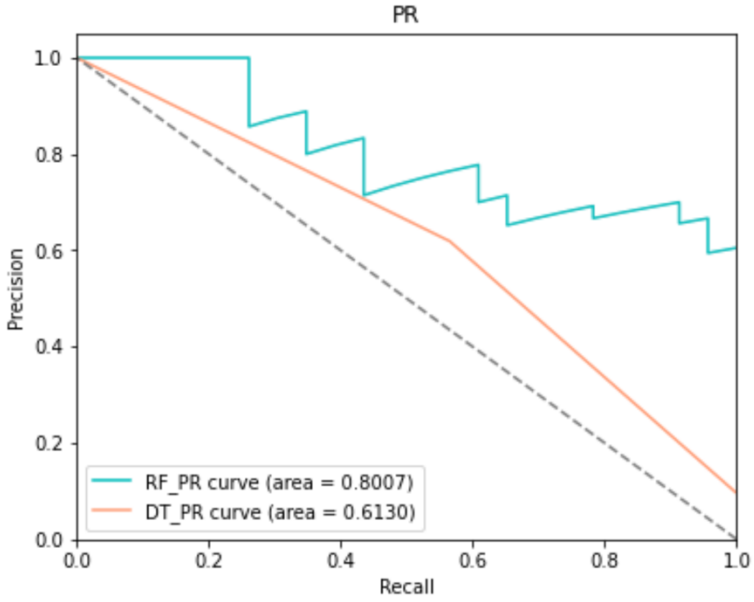


Fig14. PR curve of two models

Correspondingly, if we use Recall (same as TPR) as the x-axis, and Precision (TP/(TP + FP)) as the y-axis, we can get the PR curve, Both indicators of the PR curve focus on positive examples. In the category imbalance problem [8], since the main concern is positive examples, the PR curve is widely regarded as the ROC curve in this case. Similarly, the pr curve also has AUC, and Sk-learn also provide "precision_recall_curve", we can draw PR curves of two models as Fig14.

As a result of above experiments, although from the accuracy of the 10-fold cross validation, the performance gap between the random forest and the single decision tree on this problem is not large, but by drawing the ROC curve and the PR curve analysis, the curves of the two classifiers do not cross, but the curve of the random forest "wraps" the curve of the decision tree, which means the performance of random forest is significantly better than decision tree and has better generalization.

5 Conclusion and further work

In this project, we use random forest and decision tree to identity the "legendary" Pokémon. These two classifiers have good performance on this problem and the accuracy after 10-fold cross validation all above 90%, which is good result. But through further analysis, the performance of random forest on this issue is due to the decision tree. In addition, the two classifiers are also different in feature importance, but the same point is that "Total" is the most important feature in determining whether a Pokémon is "Legendary".

The experiment may be improved in the following aspects: the first is data preprocessing. Making the numerical data dimensionless may improve the performance of the classifier; Secondly, in tuning the hyperparameters of the classifier, if use cross-validation and use the oversampling method at each fold, we may get a better combination of parameters.

References

[1] “www.pokemon.com”

[2] “The Pokémon Company International” *Pokédex*, 2015

[3] “Google buys Kaggle and its gaggle of AI geekes” *CNET*, 2018

[4] Asier López Zorrilla “Statistical analysis of Pokémon”,2017

[5] T. Menzies, Y. Hu “Data Mining For Very Busy People” *IEEE Computer,* 2003

[6] Ho, Tin Kam “Random Decision Forests (PDF)” *Proceedings of the 3rd International Conference on Document Analysis and Recognition,* 1995

[7] Fawcett, Tom “ An Introduction to ROC Analysis" (PDF)” *Pattern Recognition Letters*,2005.

[8] Zhou ZhiHua “Machine Learning”, 2016