The International Conference on Computing Innovation and Applied Physics (CONF-CIAP 2022)
DOI: 10.54254/2753-8818/2/20220183

# Python-based Model Optimization Platform

**Zhangjin Ding**

School of Hospitality and Tourism Management, University of Surrey, Guildford, England, United Kingdom, GU2 7XH

dingzhangjin1999@gmail.com

**Abstract.** With the increasing degree of informatization in today's society, the presentation of problems has become more complex, which puts forward higher requirements for people's ability to solve problems. Python is a popular language recently, and it is very popular among developers because of the many mature libraries that are encapsulated in it. People can use related libraries in Python and use open-source related libraries for algorithm research. The main purpose of this paper is to study the optimization platform of the model based on Python. This paper mainly analyzes the characteristics of the Python language and the structure of Python programming, and uses the relevant database of Python to realize the modeling work. The experiment shows that the accuracy of the decision tree model is 96.94 %, the accuracy of the KNN classification model is 89.05%.

**Keywords:** Python Language, Model Optimization, Database Sets, Python Programs.

## 1. Introduction

Over the years, Python has become the first choice for many people to perform a data analysis and model prediction. Because the Python language is extremely simple and well-known to the public, many people like to use Python. For Python, it can be said that everyone is very familiar with it. At present, Python is a modeling language that is used a lot today. Its inventor is Van Rossum. His expectation at the time was to make beginners more Convenient learning, compared to C language, it is easier to carry out a communication. In the subsequent development process, with the continuous change of technology and technology, the Python compiler was also invented [1, 2].

In their research on Python and model optimization platform design, Farina et al. introduced disropt, a Python package for distributed optimization of networks [3]. Focus on collaborative settings, where optimization problems must be solved by peer-to-peer processors (without a central coordinator) that only have access to partial knowledge of the entire problem. A simple syntax is designed to allow easy modeling of the problem. Ullah proposed a Biogeography-Based Optimization (BBO) method to optimize the current coefficients of Constructive Cost Model（COCOMO-II） to better estimate the cost or effort of a software project [4]. Experiments are performed on two standard datasets: NASA-93 and the Turkish Industrial Software Project. The performance of the proposed algorithm called BBO-COCOMO-II is evaluated.

This paper mainly focuses on Python, and conducts related research on the model optimization platform. This paper mainly introduces the advantages of the Python language; analyzes the Python program structure diagram and from the perspective of top-level design, we can divide Python into three

DOI: 10.54254/2753-8818/2/20220183

components; analyzes the initial feature set space feature design, and proposes related calculations; training and experimentation. This paper uses Python language design to make the model optimization platform more accurate, easier for customers to analyze and read, and lay the foundation for subsequent research.

## 2. Design and research of model optimization platform based on python

### *2.1. The advantages of the python language*

The Python language pays more attention to the high readability of the code, so that customers can analyze the meaning of the language more intuitively, and it is easier to write and modify subsequent programs. Compared with the C language, its expression is very similar, including the expression of statements, such as: if, for, while statement, etc. [5, 6]. The Python language has many advantages, as follows:

Simple and clear. The module statement is simple and easy to understand, you don't need to understand the meaning of the statement itself, you only need to pay attention to the content of the written statement.

Open source. It is one of the open source software, and users can re-package and modify the code into other software as needed.

Scalability. If the publisher does not wish to open source the code, it can be packaged in C or C++, and finally in a Python language module.

Embeddability. The Python language can be embedded in C or C++ so that its scripting capabilities can be applied.

Rich standard library. The Python language standard library is powerful, can handle all kinds of work, meet more requirements, and has complete functions.

Code specification. The Python language does not need to be compiled to binary code.

As mentioned above, the Python language has a powerful standard library, and it also provides a large number of application modules, which users can directly call and apply according to their own needs. Python language can be mature and widely used in website data crawling, data operation, image system, and Web system development. The Python language has many advantages, and we will not enumerate them one by one here, but it also has certain limitations. For example, compared with the C language, the running speed is slower, because it translates each line of code into the system CPU. A computer language that can be read and therefore runs at a lower speed. Of course, as an open source language, it also means that the language cannot be encrypted [7, 8].

### *2.2. The programming structure of python*

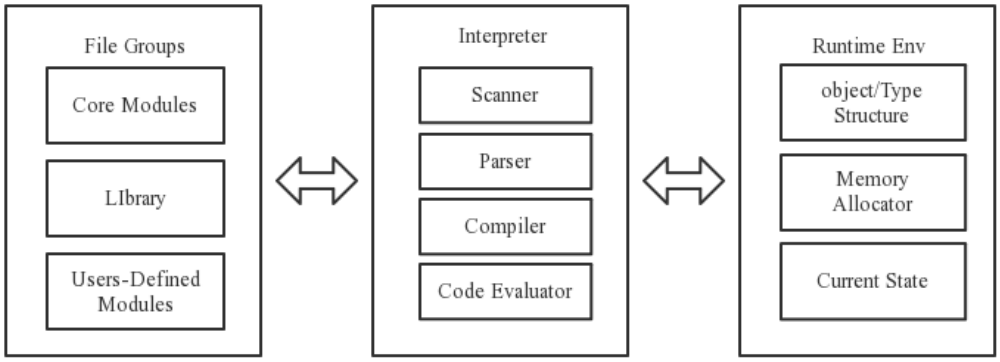


**Figure 1.** Python program structure diagram.

From a top-level design perspective, Python can be divided into three components:

The code file File Groups generally includes libraries and modules, in addition to user-defined modules.

The runtime environment Runtime Env contains basic type structures and runtime objects, as well as

DOI: 10.54254/2753-8818/2/20220183

memory allocators and real-time runtime status information.

File Groups is also our code. In the process of writing Python code, we will have code modules written by ourselves, as well as modules generated by the dependent core and third-party modules and library files (so library under Linux, And under Windows is the dll library).

Here datetime belongs to the core module, and requests is a third-party module. This code originally belongs to the user-defined code module.

Scanner is a work responsible for lexical analysis. It can separate tokens line by line. Parser is responsible for syntax analysis, condensing tokens into abstract syntax trees, and Compiler transforming syntax trees into instruction sets. The bytecode stream can finally be used by Code Evaluator for enterprises to execute our data bytecode.

Object and Type Structure, the former is the object that appears in the running process of the program, and the latter is a built-in object in Python, such as int, List, dict, etc.

However, the Memory Allocator is responsible for applying for the memory required to create an object, which itself is to encapsulate the malloc() function in the C language.

Current State is responsible for maintaining various types of state information that appear during operation, so that in the process of program execution, if the state changes (normal state and abnormal state), it can still run normally [9, 10].

### *2.3. Characteristics of data selection*

The selection of data should have the following characteristics:

Uniqueness. The collected data must be unique, and the duplicate data can be filtered to a certain extent in the preliminary data screening stage. For example, there may be multiple time-sensitive recruitment data for the same job recruitment information of the same company, so the latest recruitment data will be taken as the main, and the original old data will be covered. In this way, the preliminary data uniqueness rules are achieved.

Accuracy. The collected data will have multiple dimensions, and the dimension information that can accurately describe the data is selected in the data collection stage. If there is less relevant or obviously irrelevant dimension information, it can be removed at this stage [11, 12].

Correlation. The collected data needs to be related in one dimension or several dimensions.

### *2.4. Initial feature set spatial feature design*

Spatial features are intended to reflect the spatial structure of the code and the properties of the information level. Among the spatial features, the number of program branches, the depth of code blocks and the McCabe complexity reflect the structural characteristics of the code, the Halstead index reflects the capacity complexity of the code, and the function cohesion and code entropy reflect the cohesion level and information content of the code. The code features reflected by the spatial features are deeper than the text-level features, and are an important complement and part of the feature set.

(1) Function cohesion

The lack of cohesion of functions negatively affects source code quality and correlates with the number of defects. In the code of beginners, it is easy to appear complex functions. This system is based on Python code, takes function cohesion into consideration, and uses the degree of overlap of words between different statement blocks in the function as the basis for judgment. The algorithm for calculating the cohesion of a function is shown in Algorithm 1. The formulas for the degree of lexical overlap and functional cohesion are as follows:

$$\text{degree of lexical overlap}_{[B1,B2]} = \frac{|B1[Token] \cap B2[Token]|}{|B1[Token] \cup B2[Token]|} \tag{1}$$

$$\text{functional cohesion} = \max(\text{degree of lexical overlap}[Bi, Bj])$$

(2) Code entropy

Code Entropy (Information Entropy): Entropy is often viewed as the complexity, degree of disorder, or amount of information in a signal or dataset. Calculate the entropy of a piece of code with the following

DOI: 10.54254/2753-8818/2/20220183

formula, where xi is a Token in the code fragment, and count(xi) is the number of occurrences of xi.

$$p(x_i) = \frac{count(x_i)}{\sum_{j=1}^{n} count(x_j)} \quad (2)$$

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \quad (3)$$

(3) Halstead complexity

Halstead complexity uses two indicators, Program Vocabulary and Program Volume, and the formulas are:

Program Vocabulary： $n = n1 + n2$

Program Volumn : $V = (N1 + N2) * \log_2(n)$ （4）

(4) McCabe complexity

The McCabe complexity is calculated as follows:

$V(G) = e - n + 2$ （5）

## 3. Experimental research on model optimization platform based on python

### *3.1. Python and database interface*

Python needs to connect to a database during operation. Whether it is MySQL, SQL Server or PostgreSQL, cursors can be used, so learning to use Python DB-API is a must. Because DB-API can provide the same interface for different databases, it makes the following operations and porting code very simple. Various databases for Python are as follows:

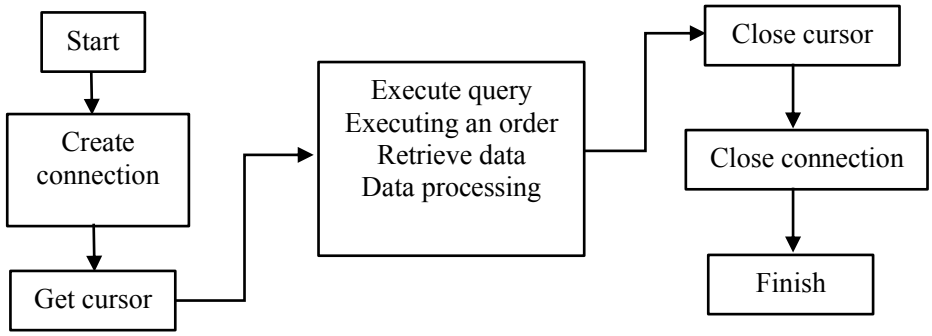


**Figure 2.** Python database interface process.

Various different types of databases:

IBM DB2: The management system of this database was invented by IBM. The advantage of this database is that it can be used by multiple users to perform a query process.

Firebird (and Interbase): This database is very convenient and fast, and it has good applicability for some companies that do not have very large requirements.

Informix: This is also invented by IBM, this is a fairly large database, similar to IBM DB2, is used to process IBM transactions online.

Ingres: It is an ancient database management system. The code adopted is the BSD license. Now some mainstream databases such as Sybase and Server are all developed and generated based on it, so this is a A database that has a huge impact on the history of computer development.

MySQL: The invention of this library comes from Oracle Corporation. It is the most popular database in the world. It is very convenient and easy to use. It can be seen from small personal computers to large companies.

Oracle: This is the most mature library about Oracle, and it is also a library that is widely used by everyone. It is very convenient. For our users, the porting performance is also very good, and it can be used in various computers. surroundings.

DOI: 10.54254/2753-8818/2/20220183

PostgreSQL: The original name of PostgreSQL is POSTGRES. As a database management system, it is relational. It can support most SQL standards and also give some other modern features, such as foreign keys, triggers, MVCC and Views, etc., in addition to this data management system can also be extended.

### *3.2. Model iterative training*

Abstract the training process into an automatic iterative training process:

(1) Feature set preparation: extract the feature values of the code fragments in the training set; remove the outliers in the feature values; format the data according to the data structure required by the XGBoost model training set.

(2) Model training: The feature set will change each time, so adjust the model input; during model training, pay attention to overfitting prevention; in each iteration process, use ten-fold cross-check to calculate the prediction accuracy of the model.

(3) Evaluation model: Evaluate the accuracy of the model and determine whether to continue the iteration. When the accuracy of the model has a large gap (>4%) or the size of the feature set reaches the expected (10-15), the iteration is stopped, and the feature set and the model at this time are output.

(4) Evaluation feature set: obtain the influence of features on readability; remove the features with the lowest weight to form a new feature set.

## 4. Experimental analysis of model optimization platform based on python

### *4.1. Model feature analysis*

The model is output when the iteration stops, and the model is considered to be the optimal model at this time. At this time, the feature set and contribution are as follows:

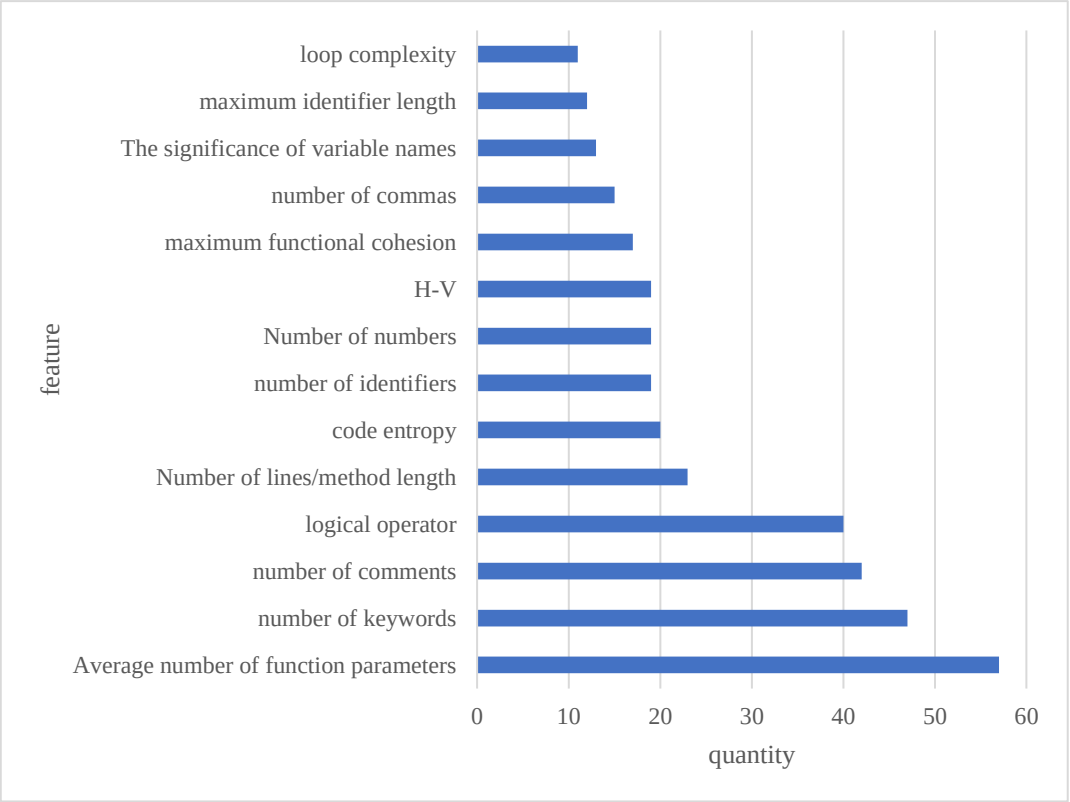


**Figure 3.** Optimal feature set and contribution.

As can be seen from Figure 3, spatial features and Code-Smell features have an important impact on the readability judgment, and the addition of these two features is an important enrichment to the classic

DOI: 10.54254/2753-8818/2/20220183

readability feature set.

*4.2. Model evaluation*

(1) Decision tree

1) Modeling based on decision tree algorithm

Use python's sklearn library to prepare training and test sets during the modeling phase. The training set generates the data model, and the test set tests the generated model. How to generate an effective dataset is a key step in the modeling process, and decision tree modeling is done through sklearn's DecisionTreeClassifier. .

2) Decision tree model evaluation

Perform prediction operations on the x_test, y_test test sets. pre is the prediction result generated by the test set. This method has been encapsulated in sklearn. The following code is used to generate a confusion matrix and evaluate the classification results of the model.

# Generate confusion matrix results using metrics' confusion_matrix

The results are as follows:

**Table 1.** Decision tree model confusion matrix table.

| Predicted value/Actual value | 0 (lower) | 1 (suitable) | 2 (higher) |
|---|---|---|---|
| 0 (lower) | 358 | 11 | 0 |
| 1 (suitable) | 1 | 570 | 0 |
| 2 (higher) | 7 | 12 | 55 |

Through the confusion matrix, the recognition rate and misclassification rate of the model can be obtained, as shown in the following table:

**Table 2.** Decision tree model accuracy and error rate.

| Accuracy | Error rate |
|---|---|
| 96.94% | 3.06% |

(2) KNN algorithm

1) Modeling based on KNN algorithm

The KNN algorithm is similar to the decision tree classifier generation process. It is also necessary to separate and extract two sets of data sets (training set and test set). The KNN algorithm is modeled by the KNeighborsClassifier method of sklearn.

2) KNN classification model evaluation:

As in the previous section, make predictions on the model through the test set and analyze the predicted results. The results are as follows:

**Table 3.** Knn model confusion matrix table.

| Predicted value/Actual value | 0 (lower) | 1 (suitable) | 2 (higher) |
|---|---|---|---|
| 0 (lower) | 312 | 41 | 16 |
| 1 (suitable) | 11 | 556 | 4 |
| 2 (higher) | 18 | 21 | 35 |

Through the confusion matrix, the recognition rate and misclassification rate of the model can be obtained, as shown in the following table:

**Table 4.** Knn model recognition rate and misclassification rate table.

| Accuracy | Error rate |
|---|---|
| 89.05% | 10.95% |

According to the process design, use the python language and its common libraries to complete the whole process. Through the sklearn library, the model establishment of decision tree and KNN nearest neighbor algorithm is realized. The collected datasets were tested separately. The respective accuracy and error rates are derived from the data presented by the confusion matrix generated by the test results.

Among them, the decision tree model has an accuracy rate of 96.94% and an error rate of 3.06%. The KNN classification model has an accuracy of 89.05% and an error of 10.95%.

DOI: 10.54254/2753-8818/2/20220183

## 5. Conclusions

This paper mainly analyzes the characteristics of the Python language and the structure of Python programming, and uses the relevant database of Python to realize the modeling work. The accuracy rate of the decision tree model reaches 96.94%. However, in the process of iterative training of the model, the accuracy of the evaluation model needs to be improved, and further research and continuous learning will be conducted in the follow-up.

Python is widely used in many different fields, such as big data analysis, establishment of predictive models, website maintenance, and third-party data crawling, etc. In these respects, Python has shown great usefulness and is widely used. Because the invention of Python is based on the convenience and inspiration of C, to a certain extent, we can use Python as a bridge to implement various commonly used libraries in C language, so that the library that can be used in C language, Python can also be achieved. In this case, when Python has just entered its infancy, it has many mature core data types, as well as some mature modules for an effective expansion.

## References

[1] Rao C S , Karunakara K . Efficient Detection and Classification of Brain Tumor using Kernel based SVM for MRI[J]. Multimedia Tools and Applications, 2022, 81(5):7393-7417.

[2] Alonso D H , Rodriguez L , Silva E . Flexible framework for fluid topology optimization with OpenFOAMand finite element-based high-level discrete adjoint method (FEniCS/dolfin-adjoint)[J]. Structural and Multidisciplinary Optimization, 2021, 64(6):4409-4440.

[3] Farina F , Camisa A , Testa A , et al. DISROPT: a Python Framework for Distributed Optimization[J]. IFAC-PapersOnLine, 2020, 53(2):2666-2671.

[4] Ullah A , Wang B , Sheng J , et al. Optimization of software cost estimation model based on biogeography-based optimization algorithm[J]. Intelligent Decision Technologies, 2020, 14(4):1-8.

[5] Jammalamadaka K , Parveen N . Testing coverage criteria for optimized deep belief network with search and rescue[J]. Journal of Big Data, 2021, 8(1):1-20.

[6] Cragnolini T , Sahota H , Joseph A P , et al. TEMPy 2: a Python library with improved 3D electron microscopy density-fitting and validation workflows[J]. Acta Crystallographica Section D Structural Biology, 2021, 77(1):41-47.

[7] R Sreenivasulu, Chaitanya G , Kumar G V , et al. Inverse Kinematic Solution for Five bar Parallel Linkage Planar Manipulator using PYTHON and Optimization by Taguchi Method[J]. International Journal of Engineering Trends and Technology, 2021, 69(5):94-100.

[8] Rao A , Rao C S , Cheruku D R . Differentiating digital image forensics and tampering localization by a novel hybrid approach[J]. Multimedia Tools and Applications, 2022, 81(13):18693-18713.

[9] Minquiz G M , Borja V , M López-Parra, et al. Machining Parameters and Toolpath Productivity Optimization Using a Factorial Design and Fit Regression Model in Face Milling and Drilling Operations[J]. Mathematical Problems in Engineering, 2020, 2020(5):1-13.

[10] Fearn S , Trembath D . Southern distribution limits and a translocated population of the scrub python Morelia kinghorni (Serpentes: Pythonidae) in tropical Queensland[J]. Herpetofauna, 2020, 36(2):85-87.

[11] Rao C S , Karunakara K . Efficient Detection and Classification of Brain Tumor using Kernel based SVM for MRI[J]. Multimedia Tools and Applications, 2022, 81(5):7393-7417.

[12] Melingi S B , Mojjada R K , Tamizhselvan C , et al. A self-adaptive monarch butterfly optimization (MBO) algorithm based improved deep forest neural network model for detecting and classifying brain stroke lesions[J]. Research on Biomedical Engineering, 2022, 38(2):647-660.