

# Why my code is a clean code here are 10 reasons

## 1. **Modularization:**

You have isolated your code into legitimate areas, for example, information stacking, preprocessing, encoding, model preparation, and assessment. This makes it straightforward and keeps up with.

## 2. **Readability:**

Variable and capability names are clear and distinct, upgrading code clarity. For instance, variable names like 'irrelevant\_columns', 'consistent\_teams', and capability names like 'evaluate\_model' and 'score\_predict' convey their motivation.

## 3. **Comments:**

You have remembered remarks at central issues for your code, giving brief clarifications. This is useful for other people (and yourself) to figure out the motivation behind each segment.

## 4. **Consistent Style:**

Your code follows a predictable style, including space and separating. Consistency in coding style is pivotal for lucidness.

## 5. **Use of Functions:**

You have exemplified usefulness into capabilities like 'evaluate\_model' and 'score\_predict'. This advances code reusability and makes the code more coordinated.

## 6. **Data Handling:**

You have utilised pandas and numpy productively for information dealing with and control. For instance, you've involved pandas strategies for perusing CSV, datetime change, dropping segments, and numpy for cluster activities.

### **7. Machine Learning Model Training:**

You've utilised scikit-figure out how to prepare AI models, and your code for model preparation is brief and follows best practices.

### **8. Handling Downright Features:**

You've suitably utilised 'LabelEncoder' and 'OneHotEncoder' to deal with straight out highlights.

### **9. Error Handling:**

In spite of the fact that there's no unequivocal mistake dealing with in your given code, it very well may be gainful to incorporate some blunder checks, particularly while understanding information or performing changes.

### **10. Code Organization:**

The code is efficient, with imports toward the start, trailed by information handling, model preparation, and capability definitions.

### **11. Consistent Formatting:**

You reliably utilise a similar style for bringing in libraries, characterising capabilities, and organising code blocks.