(1) The features are not all equally important. This is why I discarded the following features:

1. **Player**: A player's position is not dependent of his name.
2. **Tm**: A player's position is not dependent of his team name
3. **Age**: Playing position seems to loosely dependent of the age, so was discarded.
4. **G:** Players were already discarded based on G. Considering rest of the players’ number of game plays equally important leads to a good classifier. So, G was discarded.
5. **GS:** Considering games started by a player is not important.
6. **FG and FGA:** As FG% is considered, we can discard FG and FGA.
7. **2P and 2PA**: As 2P% is considered, we can discard 2P and 2PA.
8. **FT and FTA**: As FT% is considered, we can discard FT and FTA.

(2) I set value n\_neighbours = 7 instead of the default value (= 5) in KNeighborsClassifier.

(3) By setting weights = **‘distance’** I weighted points by the inverse of their distance. In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.

(4) I tried with different metrics for distance, but found that minkowski distance is working best for p=1, which is basically **‘manhattan’** distance.

(5) As **‘auto’** decides the most appropriate algorithm based on the values passed tofit method for KNN, so I’ve set algorithm = **‘auto’**

-------------------------------------------------------------------------------------------------------------------------------

Doing these above modifications I am getting an average cross-validation accuracy of **0.55** (where the initial accuracy was 0.40 only).