



## #innominions\_challenge\_one #knn\_from\_scratch\_in\_9\_hours



**Challenge starts** at 1:00PM, 12th Feb, 2021 🕒

**Challenge ends** at 6:00PM, 13th Feb, 2021. 🕒

Follow the instructions and implement the KNN algorithm from scratch for the dataset given below. 🧑🏻♀️🧑🏻♂️

### How to Participate?

Comment with an emoji in the LinkedIn comment section of this post.

### How to submit?

Upload the Jupyter Notebook on Github and paste the link in the comment section.

Write a post on LinkedIn about the experience and don't forget to use the hashtag mentioned below:

#innominions

#knn\_from\_scratch\_in\_9\_hours

### What's in it for you?

1. Explore, Compete and Learn 🧠
2. One lucky participant will get a pizza party 🍕
3. One lucky winner will receive a movie ticket 🎟️
4. One lucky winner will get a chance to interview with us 🧑🏻🔬



**Caution** - This is the first challenge in the series of followup challenges.

**Dataset** - Click here to download the dataset: [diamonds.csv](#)

**Data Description** - [description.txt](#)

**Task** - Predict the diamond price. 💎

Can you write the KNN code from scratch in 9 hours and make it work on the given dataset ?

Hints: You don't have to perform complete EDA, just do the data preparation and train the model.

**Step - 1:** Load the data

**Step - 2:** Handle Categorical Columns (i.e. convert them to numerical representation) - Wisely choose between LabelEncoding and OneHotEncoding

**Step - 3:** Normalise the data

**Step - 4:** Split the data - Test and Train (recommended 75:25 split)

**Step - 5:** Build KNN Algorithm from scratch and do the predictions for test data

**Step - 6:** Evaluate your model

**Step - 7:** Train a model using sklearn KNN Algorithm and compare it with your scratch implementation

**Dos -**

1. You can use sklearn for standardising and splitting the dataset
2. You can use library functions for LabelEncoder or OneHotEncoder (Choose wisely)
3. Use Jupyter Notebook to implement your work
4. Eat, Sleep, Code repeat

**Don't -**

1. Using the sklearn for training and prediction
2. Procrastination