Welcome back is project mein. Ab aapke data ko distribution complete karne ka samay aaya hai, jismein hum models ko build aur train karenge. Is se pehle, hamein yeh jaanna zaroori hai ki classification problems mein accuracy ek commonly used metric hai. Ab accuracy ko compute karne ke liye kaun sa data istemal karna chahiye? Is par sochiye. Hum yahan asal mein yeh jaan'na chahte hain ki hamara model naye data par kitna acha perform karega? Yani, samples jo algorithm ne pehle kabhi nahi dekhe hain. Jaise ki ek test set. Aap classifier ko fit karne ke liye istemal kiye gaye data ka accuracy compute kar sakte hain. Lekin, kyunki yeh data classifier ko train karne ke liye istemal kiya gaya tha, iska performance yeh nahi indicate karega ki ham kaise achhe se generalize kar sakte hain unseen data ko, isliye. Aam taur par, apne data ko do sets mein divide karna ek common practice hai - ek train set aur ek test set - apne project mein. Hamare paas pehle se training aur test data alag-alag CSV files mein available hai. Shayad aap khud se soch rahe honge ki yeh sab kyun zaroori hai? Thik hai, hum yeh split sirf hamare train dataset mein hi karenge, taki aap practice kar sakein. Kyunki yeh ek important step hai har implementation mein jo aap future mein karne waale hain. Sabse pehle, hamein sklearn model selection se train test split ko import karna hoga jab hum train test split function istemal karenge. Ham apne data ko randomly split karenge. Yahan pehla argument feature data ya training predictors hoga aur doosra target feature. Aur test size keyword argument specify karega ki original data ka kya proportion test set ke liye istemal hoga. Ant mein, random state keyword ek seed set karega random number generated ke liye jo data ko training aur test mein divide karega. Hum training set ka classifier fit karenge, uspe predictions banayenge jo hum val test set ke saath compare karenge known labels ke saath. Hum in predictions ka accuracy compute karenge future tasks mein. Ab chaliye isse run karte hain. Chaliye dekhte hain aapke data ka shape, aur yeh ek unique vector hai ek coordinate ke saath. Yeh hai aapka target. Survive karna hai ya nahi. Ab aap jaan gaye hain kaise train test split ka istemal karna hai model selection mein. Agli tasks mein hum apne machine learning models build karenge aur aapke predictions ka accuracy compute karenge. Agli task mein milte hain.

Welcome back to this project. Now it's time to finalize your data distribution for building and training the models. Before we proceed, it's important to note that accuracy is a commonly used metric in classification problems.

Now, which data should be used to compute accuracy? Think about it. What we are really interested in is how well our model will perform on new, unseen data. This refers to samples that the algorithm has never encountered before, similar to a test set. While you could compute the accuracy using the data used to train the classifier, it's important to understand that this data has already been used to train the classifier. Therefore, its performance won't indicate how well we can generalize to unseen data.

To address this issue, it's a common practice to split your data into two sets: a train set and a test set. In our project, we already have separate training and test data saved in two CSV files. You might be wondering why this split is necessary. Well, it's an important step in every implementation you'll do in the future.

To get started, we need to import the train-test split function from the sklearn model selection library. This function randomly divides our data into training and testing sets. The first argument specifies the feature data or training predictors, while the second specifies the target feature. Additionally, the test size argument determines the proportion of the original data used for the test set. Lastly, the random state argument sets a seed for the random number generator, ensuring consistent splits during training and testing.

Once we've split our data, we'll train the classifier on the training set and make predictions on the validation (val) test set. These predictions will then be compared with the known labels to compute the accuracy of our model's predictions.

Now, let's proceed with running this process. We'll examine the shape of our data, understanding that our target is to predict survival. With this understanding, we can effectively use the train-test split from the model selection library in our upcoming tasks to build machine learning models and evaluate their accuracy in predicting survival outcomes. See you in the next task!

Top of Form