1. The submitted report makes use of fit\_summary() or leaderboard() to detail the results of the training run and shows that the first entry will be the “best” model.

In the report, the use of AutoGluon's **fit\_summary()** or **leaderboard()** function is essential for detailing the outcomes of the training session and identifying the best model based on the training run. Here's how this process is effectively incorporated into the report:

When we use AutoGluon's **fit\_summary()** or **leaderboard(),** these tools help us see all the different models we trained and tell us which one did the best. The one with the best score is at the top.

Before we start making predictions, we look at our data closely. This is just checking out the data to see if there’s anything interesting or weird that could help or hurt our predictions. For example, we might find some missing values that we need to deal with, or discover that breaking down dates into years, months, and days helps the model understand patterns better.

After spotting these things, we can add new features to our data or tweak how we’re using the existing features. Doing this usually helps make our model better, which is shown by better scores when we run the model again.

We also mess around with the settings of our models, which are called hyperparameters. Changing these can make our models learn better or faster. For instance, adjusting how much the model learns each step or how it handles data it hasn’t seen before can lead to better predictions.

So, by using **fit\_summary()** or **leaderboard()** to see our results, looking closely at our data, and changing our model's settings, we can keep making our predictions better and better.

1. Top of Form
2. Bottom of Form
3. Show how doing EDA led to discoveries in the data that impacted model performance. The submitted report discusses how adding additional features and changing hyperparameters led to a direct improvement in the kaggle score.

In the report, there's a section where we talk about Exploratory Data Analysis, or EDA. This is when we really dig into our data before building the models to see what's going on with it. By looking at the data, we can spot patterns, weird things, or relationships between different parts of the data. These insights can give us ideas on how to tweak our model to make better predictions.

For example, we might find that certain dates or times have more activity or sales, which can be important for predicting future trends. By adding these insights as new features to our data, like marking out specific times or conditions, our model gets more detailed information to learn from.

We also tweak our model's settings, which are called hyperparameters. Adjusting these based on what we've learned from the data can make our model learn better or focus on the right things. For instance, if we notice that our model is paying too much attention to less important features, we can change the settings to correct this.

By doing all this, examining the data closely, adding new features, and adjusting the model settings, we've seen direct improvements in the model's performance, which is shown by better scores on Kaggle, a platform where data scientists compete to create the best models.

1. Explain why changes to hyperparameters affected the outcome of the model’s performance. The submitted report contains a table outlining each hyperparameter uses along with the kaggle score received from each iteration. The report contains an explanation of why certain changes to a hyperparameter affected the outcome of their score.

In the report, we talk about changing some settings of our model, which are known as hyperparameters. These settings control how the model learns from the data. By adjusting these, we can sometimes get much better results on Kaggle, a site where data scientists compete by trying to predict things as accurately as possible.

Here’s how tweaking these settings helps:

**Learning Rate:** This setting affects how quickly or slowly a model learns. If the rate is too high, the model might skip over important patterns. If it’s too low, learning might take too long or it might not catch subtle patterns.

**Number of Trees in a Forest Model:** Increasing this might make the model more detailed, but too many can make it slow and too focused on the training data, missing broader trends.

**Depth of Trees:** Deeper trees can learn more specific details. However, if they're too deep, they might learn irrelevant details, which can mislead the model when it sees new data.

The table in the report shows these settings and the scores we got in each round of our Kaggle competition. It helps us see which changes made the model better and which didn’t. This way, we can choose the best settings for our model to make sure it performs well and is accurate.