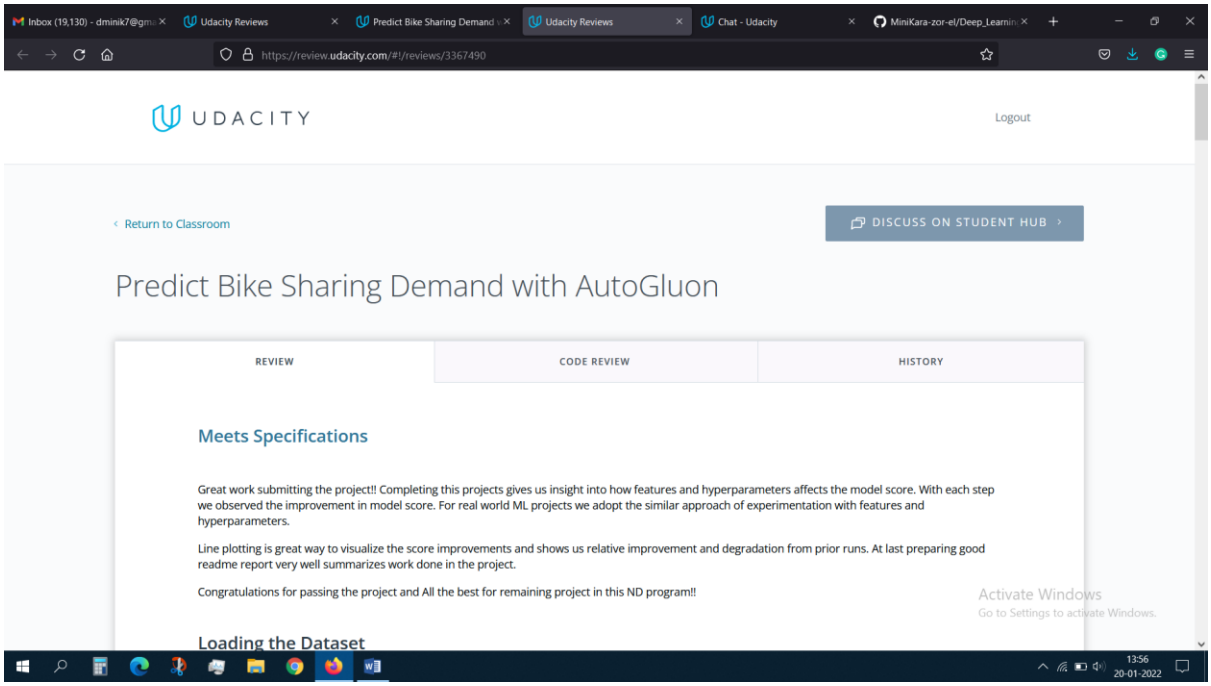
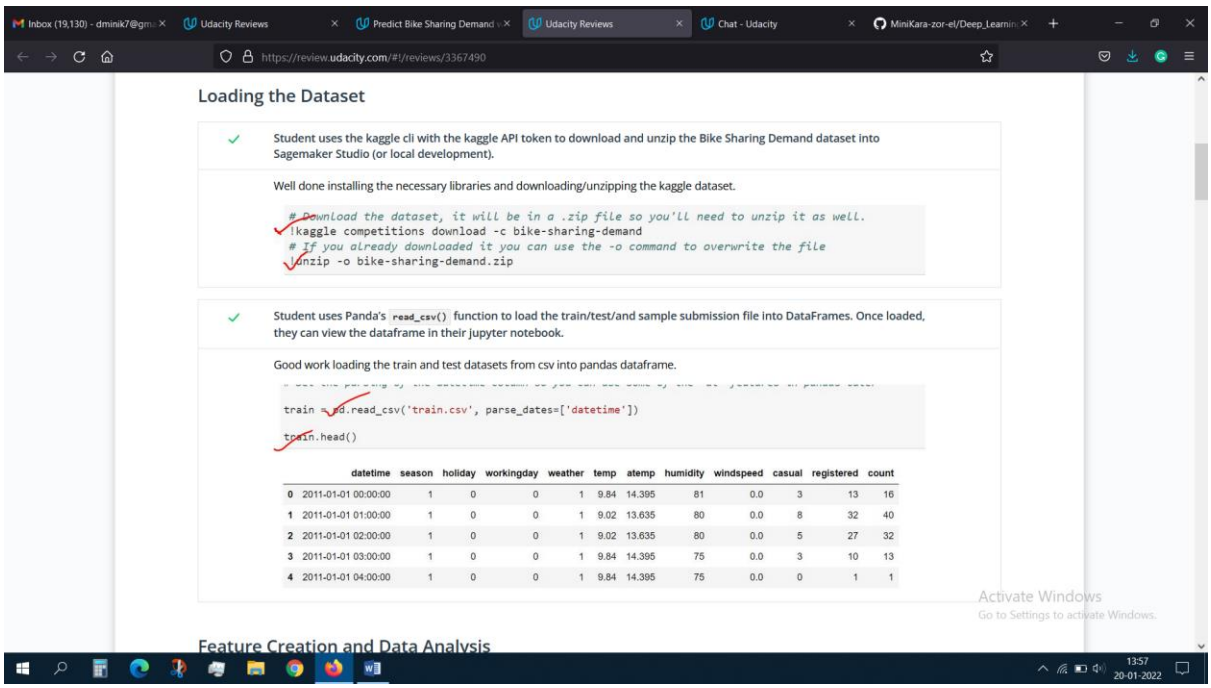

Inbox (19,130) - dminik7@gm
Udacity Reviews
Predict Bike Sharing Demand
Udacity Reviews
Chat - Udacity
MiniKara-zor-el/Deep_Learnin
https://review.udacity.com/#!/reviews/3367490
UDACITY
Logout
Return to Classroom
DISCUSS ON STUDENT HUB
Predict Bike Sharing Demand with AutoGluon
REVIEW
CODE REVIEW
HISTORY
Meets Specifications
Great work submitting the project!! Completing this projects gives us insight into how features and hyperparameters affects the model score. With each step we observed the improvement in model score. For real world ML projects we adopt the similar approach of experimentation with features and hyperparameters.
Line plotting is great way to visualize the score improvements and shows us relative improvement and degradation from prior runs. At last preparing good readme report very well summarizes work done in the project.
Congratulations for passing the project and All the best for remaining project in this ND program!!
Activate Windows
Go to Settings to activate Windows.
Loading the Dataset
13:56
20-01-2022


Inbox (19,130) - dminik7@gm
Udacity Reviews
Predict Bike Sharing Demand
Udacity Reviews
Chat - Udacity
MiniKara-zor-el/Deep_Learnin
https://review.udacity.com/#!/reviews/3367490
Loading the Dataset
Student uses the kaggle cli with the kaggle API token to download and unzip the Bike Sharing Demand dataset into Sagemaker Studio (or local development).
Well done installing the necessary libraries and downloading/unzipping the kaggle dataset.
Download the dataset, it will be in a .zip file so you'll need to unzip it as well.
!kaggle competitions download -c bike-sharing-demand
If you already downloaded it you can use the -o command to overwrite the file
!unzip -o bike-sharing-demand.zip
Student uses Panda's read_csv() function to load the train/test/and sample submission file into DataFrames. Once loaded, they can view the dataframe in their jupyter notebook.
Good work loading the train and test datasets from csv into pandas dataframe.
train = pd.read_csv('train.csv', parse_dates=['datetime'])
train.head()
datetime season holiday workingday weather temp atemp humidity windspeed casual registered count
0 2011-01-01 00:00:00 1 0 0 1 9.84 14.395 81 0.0 3 13 16
1 2011-01-01 01:00:00 1 0 0 1 9.02 13.635 80 0.0 8 32 40
2 2011-01-01 02:00:00 1 0 0 1 9.02 13.635 80 0.0 5 27 32
3 2011-01-01 03:00:00 1 0 0 1 9.84 14.395 75 0.0 3 10 13
4 2011-01-01 04:00:00 1 0 0 1 9.84 14.395 75 0.0 0 1 1
Activate Windows
Go to Settings to activate Windows.
Feature Creation and Data Analysis
13:57
20-01-2022

Feature Creation and Data Analysis

✓ Student uses data from one feature column and extract data from it to use in a new feature column.

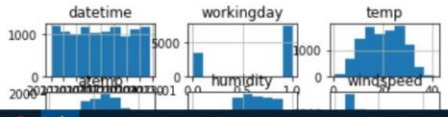
Excellent work splitting the datetime column to extract date, month and year features.

```
# create a new feature
train['hour'] = train['datetime'].dt.hour
train['day'] = train['datetime'].dt.day
train['month'] = train['datetime'].dt.month


test['hour'] = test['datetime'].dt.hour
test['day'] = test['datetime'].dt.day
test['month'] = test['datetime'].dt.month
```

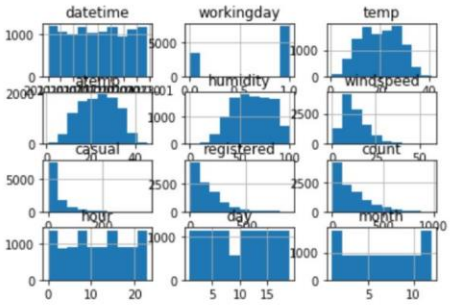
✓ Student creates a matplotlib image showing histograms of each feature column in the train dataframe.

Well done!! A histogram allows you to see the frequency distribution of a data set. It offers an "at a glance" picture of a distribution pattern, charted in specific categories. Histograms are one of the most frequently used methods for charting historical data.



✓ Student creates a matplotlib image showing histograms of each feature column in the train dataframe.

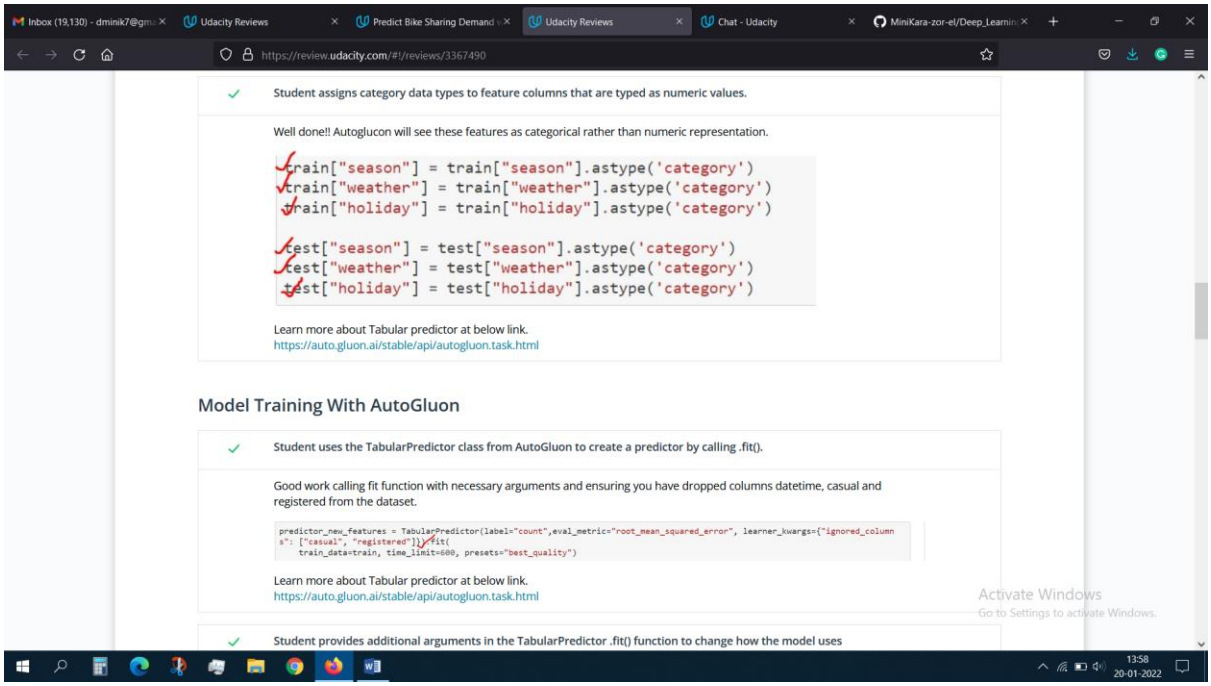
Well done!! A histogram allows you to see the frequency distribution of a data set. It offers an "at a glance" picture of a distribution pattern, charted in specific categories. Histograms are one of the most frequently used methods for charting historical data.



✓ Student assigns category data types to feature columns that are typed as numeric values.

Well done!! Autoglucon will see these features as categorical rather than numeric representation.

```
train["season"] = train["season"].astype('category')
train["weather"] = train["weather"].astype('category')
```



Student assigns category data types to feature columns that are typed as numeric values.

Well done!! Autoglucon will see these features as categorical rather than numeric representation.

```
train["season"] = train["season"].astype('category')
train["weather"] = train["weather"].astype('category')
train["holiday"] = train["holiday"].astype('category')

test["season"] = test["season"].astype('category')
test["weather"] = test["weather"].astype('category')
test["holiday"] = test["holiday"].astype('category')
```

Learn more about Tabular predictor at below link.
https://auto.gluon.ai/stable/api/autogluon.task.html

Model Training With AutoGluon

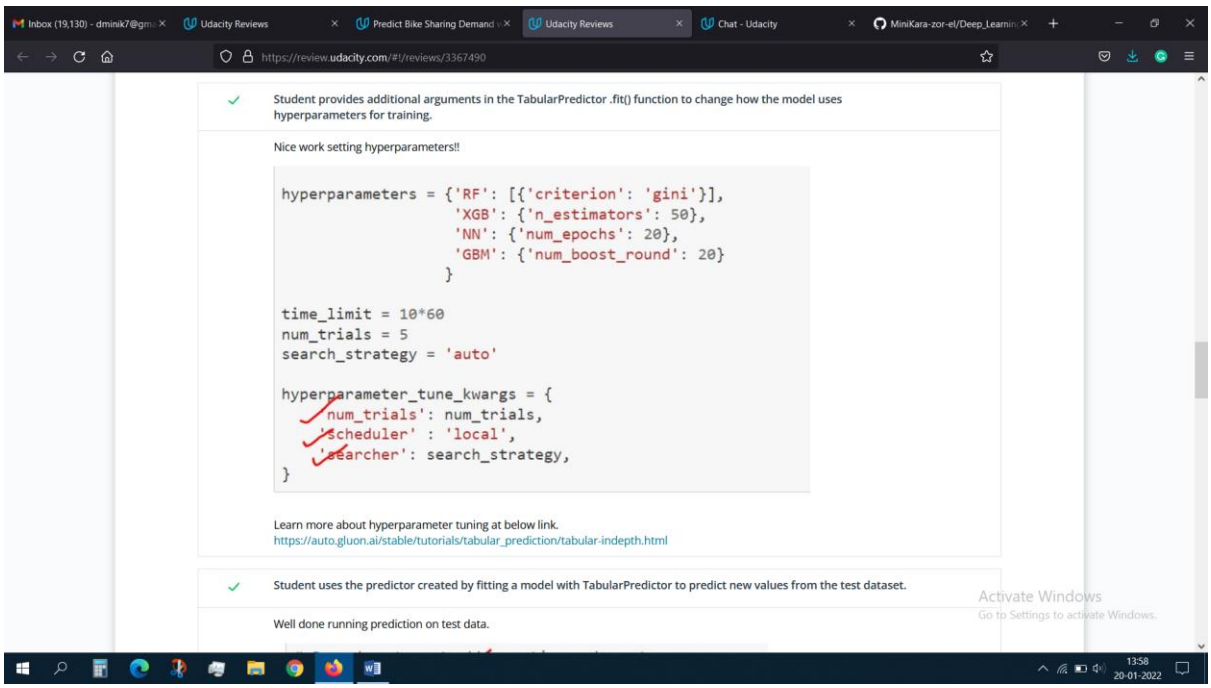
Student uses the TabularPredictor class from AutoGluon to create a predictor by calling .fit().

Good work calling fit function with necessary arguments and ensuring you have dropped columns datetime, casual and registered from the dataset.

```
predictor_new_features = TabularPredictor(label="count",eval_metric="root_mean_squared_error", learner_kwargs={"ignored_column
s": ["casual", "registered"]}).fit(
    train_data=train, time_limit=600, presets="best_quality")
```

Learn more about Tabular predictor at below link.
https://auto.gluon.ai/stable/api/autogluon.task.html

Student provides additional arguments in the TabularPredictor .fit() function to change how the model uses



Student provides additional arguments in the TabularPredictor .fit() function to change how the model uses hyperparameters for training.

Nice work setting hyperparameters!!

```
hyperparameters = {'RF': [{'criterion': 'gini'}],
                   'XGB': {'n_estimators': 50},
                   'NN': {'num_epochs': 20},
                   'GBM': {'num_boost_round': 20}
                  }

time_limit = 10*60
num_trials = 5
search_strategy = 'auto'

hyperparameter_tune_kwargs = {
    'num_trials': num_trials,
    'scheduler' : 'local',
    'searcher': search_strategy,
}
```

Learn more about hyperparameter tuning at below link.
https://auto.gluon.ai/stable/tutorials/tabular_prediction/tabular-indepth.html

Student uses the predictor created by fitting a model with TabularPredictor to predict new values from the test dataset.

Well done running prediction on test data.

Activate Windows
Go to Settings to activate Windows.

Student uses the predictor created by fitting a model with TabularPredictor to predict new values from the test dataset.

Well done running prediction on test data.

```
# Remember to set all negative values to zero
predictor_new_hpo=predictor_new_hpo.predict(test)
predictor_new_hpo[predictor_new_hpo<0]=0
predictor_new_hpo.describe()
```

Compare Model Performance

Student uses the kaggle cli to submit their predictions from the trained AutoGluon Tabular Predictor to Kaggle for a public score submission.

You have successfully submitted prediction for all three trained model.

```
!kaggle competitions submit -c bike-sharing-demand -f submission_new_hpo.csv -m "new features with hyperparameters"
```

```
100%|████████████████████████████████████████| 188k/188k [00:00<00:00, 454kB/s]
Successfully submitted to Bike Sharing Demand
```

```
!kaggle competitions submissions -c bike-sharing-demand | tail -n +1 | head -n 6
```

```
fileName                    date                 description                      status    publicScore  privateScore
--------------------------  -------------------  -------------------------------  --------  -----------  ------------
submission_new_hpo.csv      2022-01-12 15:51:03  new features with hyperparameters  complete  0.46413      0.46413
submission_new_features.csv  2022-01-12 15:07:25  new features                     complete  0.48869      0.48869
submission.csv              2022-01-12 14:06:20  first raw submission             complete  1.39644      1.39644
submission_new_hpo.csv      2022-01-11 18:12:01  new features with hyperparameters  complete  0.51059      0.51059
```

Student uses matplotlib or google sheets/excel to chart model performance metrics in a line chart. The appropriate metric will be derived from the either `fit_summary()` or `leaderboard()` of the predictor. Y axis is the metric number and X axis is each model iteration.

Well done taking best score from fit_summary for each trained model and plotting line chart. We can see improvement in score with featuring engineering and hyperparamter tuning.

```
fig = pd.DataFrame(
    {
        "model": ["initial", "add_features", "hpo"],
        "score": [-115.031935, -35.229910, -41.281192]
    }
).plot(x="model", y="score", figsize=(8, 6)).get_figure()
fig.savefig('model_train_score.png')
```

Student uses matplotlib or google sheets/excel to chart changes to the competition score. Y axis is the kaggle score and X axis is each model iteration.

Good work on competition scores!!

```
fig = pd.DataFrame(
    {
        "test_eval": ["initial", "add_features", "hpo"],
        "score": [1.38964, 0.48869 , 0.46413]
    }
).plot(x="test_eval", y="score", figsize=(8, 6)).get_figure()
fig.savefig('model_test_score.png')
```

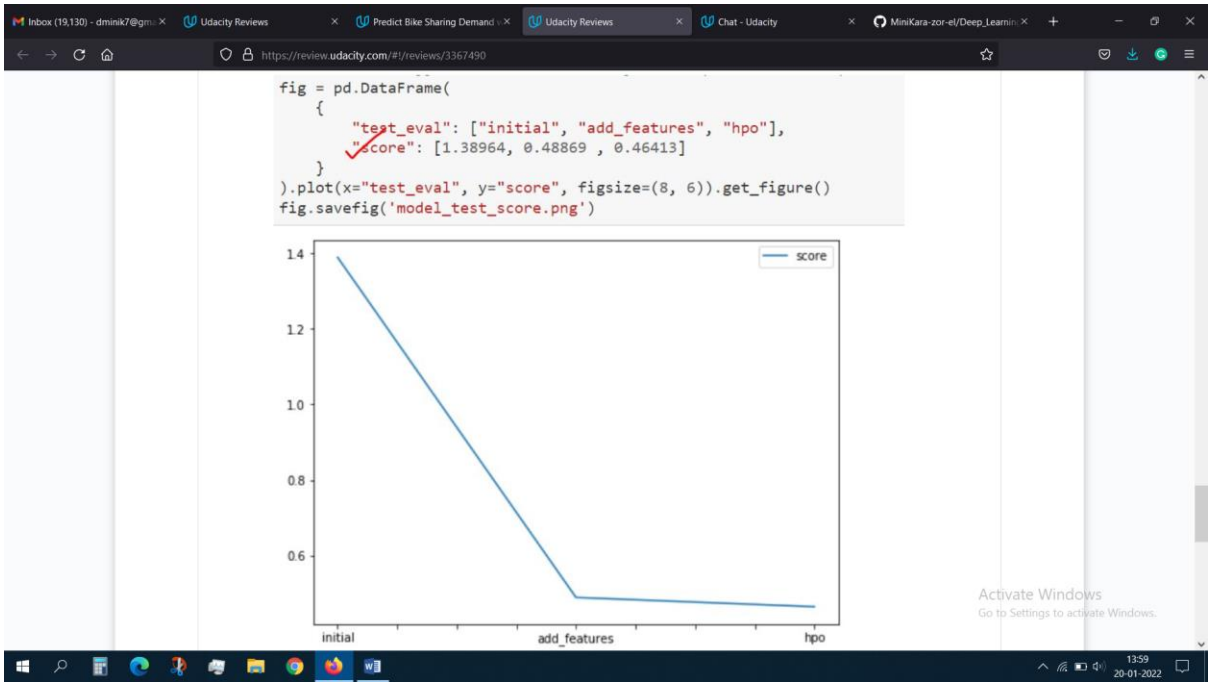
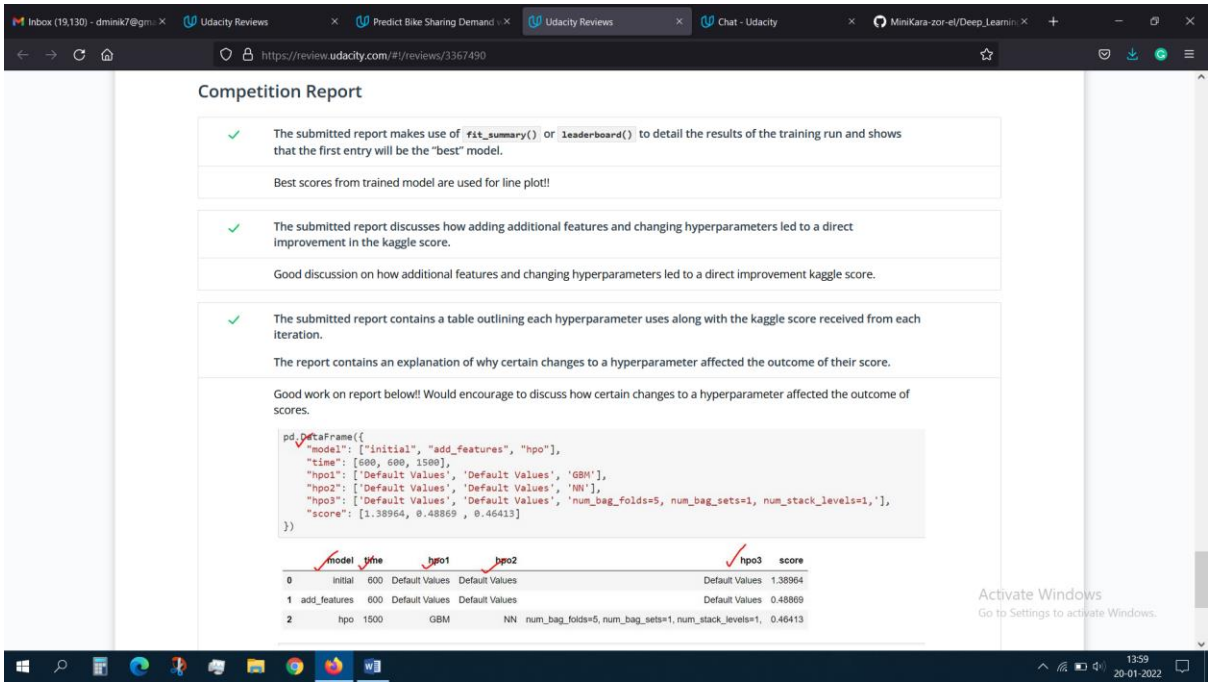
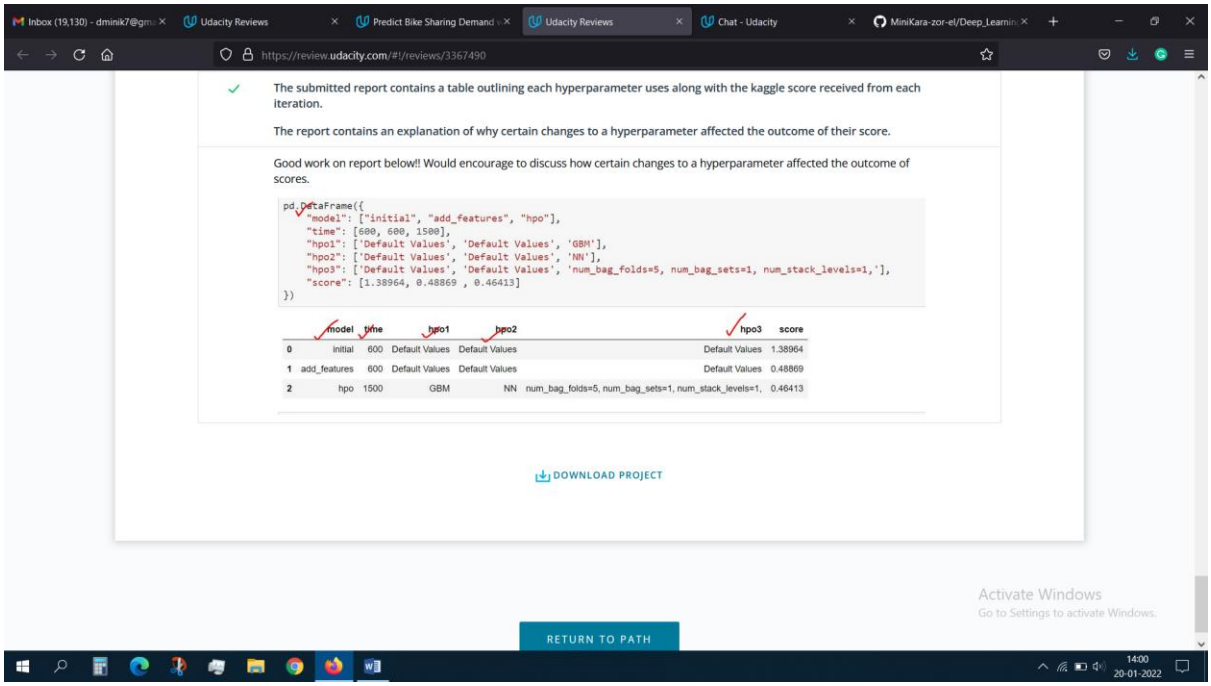


fig = pd.DataFrame(
{
"test_eval": ["initial", "add_features", "hpo"],
"score": [1.38964, 0.48869 , 0.46413]
}
).plot(x="test_eval", y="score", figsize=(8, 6)).get_figure()
fig.savefig('model_test_score.png')
score
1.4
1.2
1.0
0.8
0.6
initial
add_features
hpo
Activate Windows
Go to Settings to activate Windows.


Competition Report
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.
Best scores from trained model are used for line plot!!
The submitted report discusses how adding additional features and changing hyperparameters led to a direct improvement in the kaggle score.
Good discussion on how additional features and changing hyperparameters led to a direct improvement kaggle score.
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.
Good work on report below!! Would encourage to discuss how certain changes to a hyperparameter affected the outcome of scores.
pd.DataFrame({
"model": ["initial", "add_features", "hpo"],
"time": [600, 600, 1500],
"hpo1": ['Default Values', 'Default Values', 'GBM'],
"hpo2": ['Default Values', 'Default Values', 'NN'],
"hpo3": ['Default Values', 'Default Values', 'num_bag_folds=5, num_bag_sets=1, num_stack_levels=1,'],
"score": [1.38964, 0.48869 , 0.46413]
})
model time hpo1 hpo2 hpo3 score
0 initial 600 Default Values Default Values Default Values 1.38964
1 add_features 600 Default Values Default Values Default Values 0.48869
2 hpo 1500 GBM NN num_bag_folds=5, num_bag_sets=1, num_stack_levels=1, 0.46413
Activate Windows
Go to Settings to activate Windows.


Inbox (19,130) - dminik7@gm
Udacity Reviews
Predict Bike Sharing Demand
Udacity Reviews
Chat - Udacity
MiniKara-zor-el/Deep_Learnin
https://review.udacity.com/#!/reviews/3367490
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.
Good work on report below!! Would encourage to discuss how certain changes to a hyperparameter affected the outcome of scores.
pd.DataFrame({
"model": ["initial", "add_features", "hpo"],
"time": [600, 600, 1500],
"hpo1": ['Default Values', 'Default Values', 'GBM'],
"hpo2": ['Default Values', 'Default Values', 'NN'],
"hpo3": ['Default Values', 'Default Values', 'num_bag_folds=5, num_bag_sets=1, num_stack_levels=1,'],
"score": [1.38964, 0.48869 , 0.46413]
})
model time hpo1 hpo2 hpo3 score
0 initial 600 Default Values Default Values Default Values 1.38964
1 add_features 600 Default Values Default Values Default Values 0.48869
2 hpo 1500 GBM NN num_bag_folds=5, num_bag_sets=1, num_stack_levels=1, 0.46413
DOWNLOAD PROJECT
Activate Windows
Go to Settings to activate Windows.
RETURN TO PATH
14:00
20-01-2022