Machine Learning using tf.estimator

In this notebook, we will create a machine learning model using tf.estimator and evaluate its performance. The dataset is rather small (7700 samples), so we can do it all in-memory. We will also simply pass the raw data in as-is.

In [3]:

```
import tensorflow as tf
import pandas as pd
import numpy as np
import shutil

print(tf.__version__)
```

```
1.15.0
```

Read data created in the previous chapter.

In [4]:

```
# In CSV, label is the first column, after the features, followed by the key
CSV_COLUMNS = ['fare_amount', 'pickuplon','pickuplat','dropofflon','dropofflat','passengers', 'key']
FEATURES = CSV_COLUMNS[1:len(CSV_COLUMNS) - 1]
LABEL = CSV_COLUMNS[0]

df_train = pd.read_csv('./taxi-train.csv', header = None, names = CSV_COLUMNS)
df_valid = pd.read_csv('./taxi-valid.csv', header = None, names = CSV_COLUMNS)
```

Input function to read from Pandas Dataframe into tf.constant

In [5]:

```
def make_input_fn(df, num_epochs):
  return tf.estimator.inputs.pandas_input_fn(
    x = df,
    y = df[LABEL],
    batch_size = 128,
    num_epochs = num_epochs,
    shuffle = True,
    queue_capacity = 1000,
    num_threads = 1
  )
```

Create feature columns for estimator

In [6]:

```
def make_feature_cols():
  input_columns = [tf.feature_column.numeric_column(k) for k in FEATURES]
  return input_columns
```

Linear Regression with tf.Estimator framework

In [7]:

```
tf.logging.set_verbosity(tf.logging.INFO)

OUTDIR = 'taxi_trained'
shutil.rmtree(OUTDIR, ignore_errors = True) # start fresh each time

model = tf.estimator.LinearRegressor(
      feature_columns = make_feature_cols(), model_dir = OUTDIR)

model.train(input_fn = make_input_fn(df_train, num_epochs = 10))
```

```
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_task_id': 0, '_experimental_max_worker_delay_secs': None, '_num_ps_
replicas': 0, '_keep_checkpoint_every_n_hours': 10000, '_eval_distribute': None, '_master': '', '_tf
_random_seed': None, '_experimental_distribute': None, '_protocol': None, '_train_distribute': None,
'_global_id_in_cluster': 0, '_task_type': 'worker', '_cluster_spec': <tensorflow.python.training.ser
ver_lib.ClusterSpec object at 0x7f0a32950240>, '_num_worker_replicas': 1, '_session_creation_timeout
_secs': 7200, '_log_step_count_steps': 100, '_device_fn': None, '_model_dir': 'taxi_trained', '_keep
_checkpoint_max': 5, '_evaluation_master': '', '_save_checkpoints_steps': None, '_save_checkpoints_s
ecs': 600, '_save_summary_steps': 100, '_session_config': allow_soft_placement: true
graph_options {
  rewrite_options {
    meta_optimizer_iterations: ONE
  }
}
, '_service': None, '_is_chief': True}
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/training/train
ing_util.py:236: Variable.initialized_value (from tensorflow.python.ops.variables) is deprecated and
will be removed in a future version.
Instructions for updating:
Use Variable.read_value. Variables in 2.X are initialized automatically both in eager and graph (ins
ide tf.defun) contexts.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_estimator/python/estimato
r/inputs/queues/feeding_queue_runner.py:62: QueueRunner.__init__ (from tensorflow.python.training.qu
eue_runner_impl) is deprecated and will be removed in a future version.
Instructions for updating:
To construct input pipelines, use the `tf.data` module.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_estimator/python/estimato
r/inputs/queues/feeding_functions.py:500: add_queue_runner (from tensorflow.python.training.queue_ru
nner_impl) is deprecated and will be removed in a future version.
Instructions for updating:
To construct input pipelines, use the `tf.data` module.
INFO:tensorflow:Calling model_fn.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/feature_colum
n/feature_column_v2.py:305: Layer.add_variable (from tensorflow.python.keras.engine.base_layer) is d
eprecated and will be removed in a future version.
Instructions for updating:
Please use `layer.add_weight` method instead.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/ops/resource_v
ariable_ops.py:1630: calling BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_vari
able_ops) with constraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_estimator/python/estimato
```

```
r/canned/linear.py:308: to_float (from tensorflow.python.ops.math_ops) is deprecated and will be rem
oved in a future version.
Instructions for updating:
Use `tf.cast` instead.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/ops/array_ops.
py:1475: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/training/monit
ored_session.py:882: start_queue_runners (from tensorflow.python.training.queue_runner_impl) is depr
ecated and will be removed in a future version.
Instructions for updating:
To construct input pipelines, use the `tf.data` module.
INFO:tensorflow:Saving checkpoints for 0 into taxi_trained/model.ckpt.
INFO:tensorflow:loss = 28658.043, step = 1
INFO:tensorflow:global_step/sec: 118.548
INFO:tensorflow:loss = 6466.0728, step = 101 (0.849 sec)
INFO:tensorflow:global_step/sec: 155.07
INFO:tensorflow:loss = 13970.383, step = 201 (0.644 sec)
INFO:tensorflow:global_step/sec: 167.306
INFO:tensorflow:loss = 11700.387, step = 301 (0.599 sec)
INFO:tensorflow:global_step/sec: 147.752
INFO:tensorflow:loss = 10473.381, step = 401 (0.674 sec)
WARNING:tensorflow:It seems that global step (tf.train.get_global_step) has not been increased. Curr
ent value (could be stable): 449 vs previous value: 449. You could increase the global step by passi
ng tf.train.get_global_step() to Optimizer.apply_gradients or Optimizer.minimize.
INFO:tensorflow:global_step/sec: 184.121
INFO:tensorflow:loss = 5168.2344, step = 501 (0.545 sec)
INFO:tensorflow:Saving checkpoints for 573 into taxi_trained/model.ckpt.
INFO:tensorflow:Loss for final step: 9433.614.
```

Out[7]:

```
<tensorflow_estimator.python.estimator.canned.linear.LinearRegressor at 0x7f0a329502e8>
```

Evaluate on the validation data (we should defer using the test data to after we have selected a final model).

In [8]:

```
def print_rmse(model, name, df):
  metrics = model.evaluate(input_fn = make_input_fn(df, 1))
  print('RMSE on {} dataset = {}'.format(name, np.sqrt(metrics['average_loss'])))
print_rmse(model, 'validation', df_valid)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2020-01-17T20:11:08Z
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from taxi_trained/model.ckpt-573
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2020-01-17-20:11:09
INFO:tensorflow:Saving dict for global step 573: average_loss = 86.872574, global_step = 573, label/
mean = 11.419548, loss = 10498.217, prediction/mean = 11.642365
INFO:tensorflow:Saving 'checkpoint_path' summary for global step 573: taxi_trained/model.ckpt-573
RMSE on validation dataset = 9.32054615020752
```

This is nowhere near our benchmark (RMSE of $6 or so on this data), but it serves to demonstrate what TensorFlow code looks like. Let's use this model for prediction.

In [9]:

```
import itertools
# Read saved model and use it for prediction
model = tf.estimator.LinearRegressor(
      feature_columns = make_feature_cols(), model_dir = OUTDIR)
preds_iter = model.predict(input_fn = make_input_fn(df_valid, 1))
print([pred['predictions'][0] for pred in list(itertools.islice(preds_iter, 5))])
```

```
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_task_id': 0, '_experimental_max_worker_delay_secs': None, '_num_ps_
replicas': 0, '_keep_checkpoint_every_n_hours': 10000, '_eval_distribute': None, '_master': '', '_tf
_random_seed': None, '_experimental_distribute': None, '_protocol': None, '_train_distribute': None,
'_global_id_in_cluster': 0, '_task_type': 'worker', '_cluster_spec': <tensorflow.python.training.ser
ver_lib.ClusterSpec object at 0x7f0a30072ef0>, '_num_worker_replicas': 1, '_session_creation_timeout
_secs': 7200, '_log_step_count_steps': 100, '_device_fn': None, '_model_dir': 'taxi_trained', '_keep
_checkpoint_max': 5, '_evaluation_master': '', '_save_checkpoints_steps': None, '_save_checkpoints_s
ecs': 600, '_save_summary_steps': 100, '_session_config': allow_soft_placement: true
graph_options {
  rewrite_options {
    meta_optimizer_iterations: ONE
  }
}
, '_service': None, '_is_chief': True}
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from taxi_trained/model.ckpt-573
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
[11.880379, 11.521134, 11.702424, 11.525503, 11.5230665]
```

This explains why the RMSE was so high -- the model essentially predicts the same amount for every trip. Would a more complex model help? Let's try using a deep neural network. The code to do this is quite straightforward as well.

Deep Neural Network regression

In [10]:

```
tf.logging.set_verbosity(tf.logging.INFO)
shutil.rmtree(OUTDIR, ignore_errors = True) # start fresh each time
model = tf.estimator.DNNRegressor(hidden_units = [32, 8, 2],
      feature_columns = make_feature_cols(), model_dir = OUTDIR)
model.train(input_fn = make_input_fn(df_train, num_epochs = 100));
print_rmse(model, 'validation', df_valid)
```

```
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_task_id': 0, '_experimental_max_worker_delay_secs': None, '_num_ps_
replicas': 0, '_keep_checkpoint_every_n_hours': 10000, '_eval_distribute': None, '_master': '', '_tf
_random_seed': None, '_experimental_distribute': None, '_protocol': None, '_train_distribute': None,
'_global_id_in_cluster': 0, '_task_type': 'worker', '_cluster_spec': <tensorflow.python.training.ser
ver_lib.ClusterSpec object at 0x7f0a2af05898>, '_num_worker_replicas': 1, '_session_creation_timeout
_secs': 7200, '_log_step_count_steps': 100, '_device_fn': None, '_model_dir': 'taxi_trained', '_keep
_checkpoint_max': 5, '_evaluation_master': '', '_save_checkpoints_steps': None, '_save_checkpoints_s
ecs': 600, '_save_summary_steps': 100, '_session_config': allow_soft_placement: true
graph_options {
  rewrite_options {
    meta_optimizer_iterations: ONE
  }
}
, '_service': None, '_is_chief': True}
INFO:tensorflow:Calling model_fn.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/training/adagr
ad.py:76: calling Constant.__init__ (from tensorflow.python.ops.init_ops) with dtype is deprecated a
nd will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 0 into taxi_trained/model.ckpt.
INFO:tensorflow:loss = 15690.048, step = 1
INFO:tensorflow:global_step/sec: 107.839
INFO:tensorflow:loss = 34985.18, step = 101 (0.929 sec)
INFO:tensorflow:global_step/sec: 142.02
INFO:tensorflow:loss = 25564.314, step = 201 (0.704 sec)
INFO:tensorflow:global_step/sec: 140.093
INFO:tensorflow:loss = 19112.988, step = 301 (0.713 sec)
INFO:tensorflow:global_step/sec: 159.384
INFO:tensorflow:loss = 32680.238, step = 401 (0.634 sec)
INFO:tensorflow:global_step/sec: 161.491
INFO:tensorflow:loss = 25273.988, step = 501 (0.619 sec)
INFO:tensorflow:global_step/sec: 165.849
INFO:tensorflow:loss = 16857.686, step = 601 (0.601 sec)
INFO:tensorflow:global_step/sec: 150.834
INFO:tensorflow:loss = 42681.035, step = 701 (0.665 sec)
INFO:tensorflow:global_step/sec: 168.038
```

```
INFO:tensorflow:loss = 24162.627, step = 801 (0.592 sec)
INFO:tensorflow:global_step/sec: 139.69
INFO:tensorflow:loss = 17818.34, step = 901 (0.716 sec)
INFO:tensorflow:global_step/sec: 140.699
INFO:tensorflow:loss = 24854.977, step = 1001 (0.710 sec)
INFO:tensorflow:global_step/sec: 141.325
INFO:tensorflow:loss = 21016.688, step = 1101 (0.705 sec)
INFO:tensorflow:global_step/sec: 134.018
INFO:tensorflow:loss = 12343.434, step = 1201 (0.746 sec)
INFO:tensorflow:global_step/sec: 142.148
INFO:tensorflow:loss = 20845.258, step = 1301 (0.704 sec)
INFO:tensorflow:global_step/sec: 136.097
INFO:tensorflow:loss = 31104.668, step = 1401 (0.735 sec)
INFO:tensorflow:global_step/sec: 139.858
INFO:tensorflow:loss = 14530.141, step = 1501 (0.715 sec)
INFO:tensorflow:global_step/sec: 127.916
INFO:tensorflow:loss = 15158.748, step = 1601 (0.782 sec)
INFO:tensorflow:global_step/sec: 165.325
INFO:tensorflow:loss = 16550.809, step = 1701 (0.607 sec)
INFO:tensorflow:global_step/sec: 170.554
INFO:tensorflow:loss = 31012.924, step = 1801 (0.584 sec)
INFO:tensorflow:global_step/sec: 164.496
INFO:tensorflow:loss = 34722.72, step = 1901 (0.614 sec)
INFO:tensorflow:global_step/sec: 168.844
INFO:tensorflow:loss = 16918.83, step = 2001 (0.585 sec)
INFO:tensorflow:global_step/sec: 173.896
INFO:tensorflow:loss = 18876.768, step = 2101 (0.581 sec)
INFO:tensorflow:global_step/sec: 146.181
INFO:tensorflow:loss = 5199.866, step = 2201 (0.681 sec)
INFO:tensorflow:global_step/sec: 145.643
INFO:tensorflow:loss = 20283.453, step = 2301 (0.690 sec)
INFO:tensorflow:global_step/sec: 143.079
INFO:tensorflow:loss = 16087.745, step = 2401 (0.696 sec)
INFO:tensorflow:global_step/sec: 141.464
INFO:tensorflow:loss = 17800.98, step = 2501 (0.709 sec)
INFO:tensorflow:global_step/sec: 144.355
INFO:tensorflow:loss = 17845.398, step = 2601 (0.691 sec)
INFO:tensorflow:global_step/sec: 138.014
INFO:tensorflow:loss = 43504.67, step = 2701 (0.725 sec)
INFO:tensorflow:global_step/sec: 164.913
INFO:tensorflow:loss = 17780.941, step = 2801 (0.609 sec)
INFO:tensorflow:global_step/sec: 167.249
INFO:tensorflow:loss = 11847.445, step = 2901 (0.595 sec)
```

```
INFO:tensorflow:global_step/sec: 178.095
INFO:tensorflow:loss = 13236.715, step = 3001 (0.562 sec)
INFO:tensorflow:global_step/sec: 161.422
INFO:tensorflow:loss = 16014.54, step = 3101 (0.619 sec)
INFO:tensorflow:global_step/sec: 139.308
INFO:tensorflow:loss = 10231.878, step = 3201 (0.718 sec)
INFO:tensorflow:global_step/sec: 145.256
INFO:tensorflow:loss = 14271.457, step = 3301 (0.688 sec)
WARNING:tensorflow:It seems that global step (tf.train.get_global_step) has not been increased. Curr
ent value (could be stable): 3358 vs previous value: 3358. You could increase the global step by pas
sing tf.train.get_global_step() to Optimizer.apply_gradients or Optimizer.minimize.
INFO:tensorflow:global_step/sec: 106.433
INFO:tensorflow:loss = 19755.09, step = 3401 (0.944 sec)
INFO:tensorflow:global_step/sec: 114.053
INFO:tensorflow:loss = 18197.113, step = 3501 (0.873 sec)
INFO:tensorflow:global_step/sec: 114.993
INFO:tensorflow:loss = 23105.23, step = 3601 (0.870 sec)
INFO:tensorflow:global_step/sec: 155.164
INFO:tensorflow:loss = 14897.652, step = 3701 (0.647 sec)
WARNING:tensorflow:It seems that global step (tf.train.get_global_step) has not been increased. Curr
ent value (could be stable): 3721 vs previous value: 3721. You could increase the global step by pas
sing tf.train.get_global_step() to Optimizer.apply_gradients or Optimizer.minimize.
WARNING:tensorflow:It seems that global step (tf.train.get_global_step) has not been increased. Curr
ent value (could be stable): 3789 vs previous value: 3789. You could increase the global step by pas
sing tf.train.get_global_step() to Optimizer.apply_gradients or Optimizer.minimize.
INFO:tensorflow:global_step/sec: 121.4
INFO:tensorflow:loss = 15060.063, step = 3801 (0.821 sec)
WARNING:tensorflow:It seems that global step (tf.train.get_global_step) has not been increased. Curr
ent value (could be stable): 3881 vs previous value: 3881. You could increase the global step by pas
sing tf.train.get_global_step() to Optimizer.apply_gradients or Optimizer.minimize.
INFO:tensorflow:global_step/sec: 153.907
INFO:tensorflow:loss = 13353.113, step = 3901 (0.650 sec)
INFO:tensorflow:global_step/sec: 132.7
INFO:tensorflow:loss = 12578.818, step = 4001 (0.755 sec)
INFO:tensorflow:global_step/sec: 144.527
INFO:tensorflow:loss = 18975.535, step = 4101 (0.691 sec)
INFO:tensorflow:global_step/sec: 145.02
INFO:tensorflow:loss = 20018.605, step = 4201 (0.691 sec)
INFO:tensorflow:global_step/sec: 131.874
INFO:tensorflow:loss = 14638.527, step = 4301 (0.758 sec)
INFO:tensorflow:global_step/sec: 126.033
INFO:tensorflow:loss = 13563.173, step = 4401 (0.793 sec)
INFO:tensorflow:global_step/sec: 146.503
```

```
INFO:tensorflow:loss = 12628.152, step = 4501 (0.682 sec)
INFO:tensorflow:global_step/sec: 152.361
INFO:tensorflow:loss = 23083.922, step = 4601 (0.653 sec)
INFO:tensorflow:global_step/sec: 147.515
INFO:tensorflow:loss = 13198.551, step = 4701 (0.684 sec)
INFO:tensorflow:global_step/sec: 135.702
INFO:tensorflow:loss = 11634.069, step = 4801 (0.736 sec)
INFO:tensorflow:global_step/sec: 171.794
INFO:tensorflow:loss = 16247.746, step = 4901 (0.581 sec)
INFO:tensorflow:global_step/sec: 146.593
INFO:tensorflow:loss = 21979.781, step = 5001 (0.684 sec)
INFO:tensorflow:global_step/sec: 162.095
INFO:tensorflow:loss = 15464.899, step = 5101 (0.620 sec)
INFO:tensorflow:global_step/sec: 125.914
INFO:tensorflow:loss = 12855.646, step = 5201 (0.791 sec)
INFO:tensorflow:global_step/sec: 118.769
INFO:tensorflow:loss = 28153.973, step = 5301 (0.839 sec)
INFO:tensorflow:global_step/sec: 148.57
INFO:tensorflow:loss = 9092.629, step = 5401 (0.675 sec)
INFO:tensorflow:global_step/sec: 142.668
INFO:tensorflow:loss = 11223.282, step = 5501 (0.700 sec)
INFO:tensorflow:global_step/sec: 147.53
INFO:tensorflow:loss = 14686.924, step = 5601 (0.678 sec)
INFO:tensorflow:global_step/sec: 112.501
INFO:tensorflow:loss = 22257.61, step = 5701 (0.889 sec)
INFO:tensorflow:Saving checkpoints for 5729 into taxi_trained/model.ckpt.
INFO:tensorflow:Loss for final step: 16199.312.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2020-01-17T20:13:10Z
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from taxi_trained/model.ckpt-5729
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2020-01-17-20:13:11
INFO:tensorflow:Saving dict for global step 5729: average_loss = 111.65242, global_step = 5729, labe
l/mean = 11.419548, loss = 13492.766, prediction/mean = 6.428124
INFO:tensorflow:Saving 'checkpoint_path' summary for global step 5729: taxi_trained/model.ckpt-5729
RMSE on validation dataset = 10.566571235656738
```

We are not beating our benchmark with either model ... what's up? Well, we may be using TensorFlow for Machine Learning, but we are not yet using it well. That's what the rest of this course is about!

But, for the record, let's say we had to choose between the two models. We'd choose the one with the lower validation error. Finally, we'd measure the RMSE on the test data with this chosen model.

Benchmark dataset

Let's do this on the benchmark dataset.

In [11]:

```
import datalab.bigquery as bq
import numpy as np
import pandas as pd

def create_query(phase, EVERY_N):
  """
  phase: 1 = train 2 = valid
  """
  base_query = """
SELECT
  (tolls_amount + fare_amount) AS fare_amount,
  CONCAT(STRING(pickup_datetime), STRING(pickup_longitude), STRING(pickup_latitude), STRING(dropoff_latitude),
 STRING(dropoff_longitude)) AS key,
  DAYOFWEEK(pickup_datetime)*1.0 AS dayofweek,
  HOUR(pickup_datetime)*1.0 AS hourofday,
  pickup_longitude AS pickuplon,
  pickup_latitude AS pickuplat,
  dropoff_longitude AS dropofflon,
  dropoff_latitude AS dropofflat,
  passenger_count*1.0 AS passengers,
FROM
  [nyc-tlc:yellow.trips]
WHERE
  trip_distance > 0
  AND fare_amount >= 2.5
  AND pickup_longitude > -78
  AND pickup_longitude < -70
  AND dropoff_longitude > -78
  AND dropoff_longitude < -70
  AND pickup_latitude > 37
  AND pickup_latitude < 45
  AND dropoff_latitude > 37
  AND dropoff_latitude < 45
  AND passenger_count > 0
  """

  if EVERY_N == None:
    if phase < 2:
      # Training
      query = "{0} AND ABS(HASH(pickup_datetime)) % 4 < 2".format(base_query)
    else:
      # Validation
      query = "{0} AND ABS(HASH(pickup_datetime)) % 4 == {1}".format(base_query, phase)
  else:
    query = "{0} AND ABS(HASH(pickup_datetime)) % {1} == {2}".format(base_query, EVERY_N, phase)

  return query

query = create_query(2, 100000)
df = bq.Query(query).to_dataframe()
```

In [12]:

```
print_rmse(model, 'benchmark', df)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2020-01-17T20:16:04Z
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from taxi_trained/model.ckpt-5729
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2020-01-17-20:16:05
INFO:tensorflow:Saving dict for global step 5729: average_loss = 112.60278, global_step = 5729, labe
l/mean = 11.333684, loss = 14312.103, prediction/mean = 6.428129
INFO:tensorflow:Saving 'checkpoint_path' summary for global step 5729: taxi_trained/model.ckpt-5729
RMSE on benchmark dataset = 10.611445426940918
```

RMSE on benchmark dataset is **9.41** (your results will vary because of random seeds).

This is not only way more than our original benchmark of 6.00, but it doesn't even beat our distance-based rule's RMSE of 8.02.

Fear not -- you have learned how to write a TensorFlow model, but not to do all the things that you will have to do to your ML model performant. We will do this in the next chapters. In this chapter though, we will get our TensorFlow model ready for these improvements.

In a software sense, the rest of the labs in this chapter will be about refactoring the code so that we can improve it.