```
import pandas as pd

#%%
# Read data
train = pd.read_csv('train.csv')
destinations = pd.read_csv('destinations.csv')

#%%
# Get preliminary info on data
train.shape
train.info()
train['hotel_cluster'].value_counts()
train.isna().sum()

#%%
# Convert object type columns to date time and split into separate columns
train['date_time'] = pd.to_datetime(train['date_time'], errors='coerce')
train['srch_ci'] = pd.to_datetime(train['srch_ci'], errors='coerce')
train['srch_co'] = pd.to_datetime(train['srch_co'], errors='coerce')

train['dt_year'] = train['date_time'].dt.year
train['dt_month'] = train['date_time'].dt.month
train['dt_day'] = train['date_time'].dt.day
train['dt_hour'] = train['date_time'].dt.hour
train['dt_day_of_week'] = train['date_time'].dt.dayofweek

train['ci_year'] = train['srch_ci'].dt.year
train['ci_month'] = train['srch_ci'].dt.month
train['ci_day'] = train['srch_ci'].dt.day
train['ci_day_of_week'] = train['srch_ci'].dt.dayofweek

train['co_year'] = train['srch_co'].dt.year
train['co_month'] = train['srch_co'].dt.month
train['co_day'] = train['srch_co'].dt.day
train['co_day_of_week'] = train['srch_co'].dt.dayofweek

#%%
# Create features for length of stay and number of days between booking the trip and the start
of trip
train['length_of_stay'] = (train['srch_co'] - train['srch_ci']).astype('timedelta64[D]')
train['no_of_days_bet_booking'] = (train['srch_ci'] - train['date_time']).astype('timedelta64[D]')

#%%
```

```
# Fill missing values with mean, but create column that denotes this value was missing
train['orig_destination_distance_present'] =
(~train['orig_destination_distance'].isnull()).astype(int)
train['orig_destination_distance'].fillna(train['orig_destination_distance'].mean(), inplace=True)

#%%
# Remove redundant columns
train.drop(['date_time', 'srch_ci', 'srch_co'], axis = 1, inplace = True)
train.info()

#%%
# Use PCA to reduce the number of columns in destinations while preserving data
from sklearn.decomposition import PCA

pca = PCA(n_components=5)
dest_small = pca.fit_transform(destinations[["d{0}".format(i + 1) for i in range(149)]])
dest_small = pd.DataFrame(dest_small)
dest_small["srch_destination_id"] = destinations["srch_destination_id"]

#%%
# Join train with destination PCA df
train_plus = train.join(dest_small, on="srch_destination_id", how='left', rsuffix="dest")
train_plus.drop("srch_destination_iddest", axis=1, inplace = True)

#%%
# Fill missing values with -1
train_plus.fillna(-1, inplace = True)

#%%
# Save file
train_plus.to_hdf('train.h5', key='train', mode='w') # Size 37,670,293 x 42

#%%
# Read and process test data according to how I processed train data
test = pd.read_csv('test.csv')
test.info()

test['date_time'] = pd.to_datetime(test['date_time'], errors='coerce')
test['srch_ci'] = pd.to_datetime(test['srch_ci'], errors='coerce')
test['srch_co'] = pd.to_datetime(test['srch_co'], errors='coerce')

test['dt_year'] = test['date_time'].dt.year
test['dt_month'] = test['date_time'].dt.month
```

```
test['dt_day'] = test['date_time'].dt.day
test['dt_hour'] = test['date_time'].dt.hour
test['dt_day_of_week'] = test['date_time'].dt.dayofweek

test['ci_year'] = test['srch_ci'].dt.year
test['ci_month'] = test['srch_ci'].dt.month
test['ci_day'] = test['srch_ci'].dt.day
test['ci_day_of_week'] = test['srch_ci'].dt.dayofweek

test['co_year'] = test['srch_co'].dt.year
test['co_month'] = test['srch_co'].dt.month
test['co_day'] = test['srch_co'].dt.day
test['co_day_of_week'] = test['srch_co'].dt.dayofweek

test['length_of_stay'] = (test['srch_co'] - test['srch_ci']).astype('timedelta64[D]')
test['no_of_days_bet_booking'] = (test['srch_ci'] - test['date_time']).astype('timedelta64[D]')

test['orig_destination_distance_present'] = (~test['orig_destination_distance'].isnull()).astype(int)
# Fill na with TRAIN's average value, not test's
test['orig_destination_distance'].fillna(train['orig_destination_distance'].mean(), inplace=True)

test.drop(['date_time', 'srch_ci', 'srch_co'], axis = 1, inplace = True)
test_plus = test.join(dest_small, on="srch_destination_id", how='left', rsuffix="dest")
test_plus.drop("srch_destination_iddest", axis=1, inplace = True)

test_plus.fillna(-1, inplace = True)
test_plus.to_hdf('test.h5', key='train', mode='w') # Size 2,528,243 x 40
```