

# AirFlow-ANN

November 24, 2021

```
[2]: import pandas as pd
import glob

# Load the data from csv files
data = pd.concat([pd.read_csv(f) for f in glob.glob('s*.csv')])

print("----- DATA -----")
print(data.head())

print("----- DATA DIMENSIONS -----")
print(data.shape)

print("----- DATA COLUMNS -----")
print(data.columns)

print("----- DATA NAN -----")
print(data.isnull().sum())
```

```
----- DATA -----
   Unnamed: 0  Vx1  Vy1  Vz1  P1  P2  P3 \
0           0  7.107226  2.857086  0.000000  25.015300  218.231600  29.496900
1           1  7.566645  1.580754  0.000000  24.615769  217.759186  26.045025
2           2  8.232815  0.908911 -0.885180  24.720733  215.520299  27.895377
3           3  7.764773  0.857239 -0.354739  23.817487  214.758024  29.329764
4           4  7.738041  0.717836  0.871687  21.422593  216.613663  27.234682
```

```
      P4  flowrate
0  5.227500  1726.176870
1  0.615990  1682.764884
2  3.873287  1667.562648
3  4.891262  1690.969685
4  4.121431  1660.076591
```

```
----- DATA DIMENSIONS -----
(53091, 9)
```

```
----- DATA COLUMNS -----
```

```
Index(['Unnamed: 0', 'Vx1', 'Vy1', 'Vz1', 'P1', 'P2', 'P3', 'P4', 'flowrate'],
      dtype='object')
```

```
----- DATA NAN -----
```

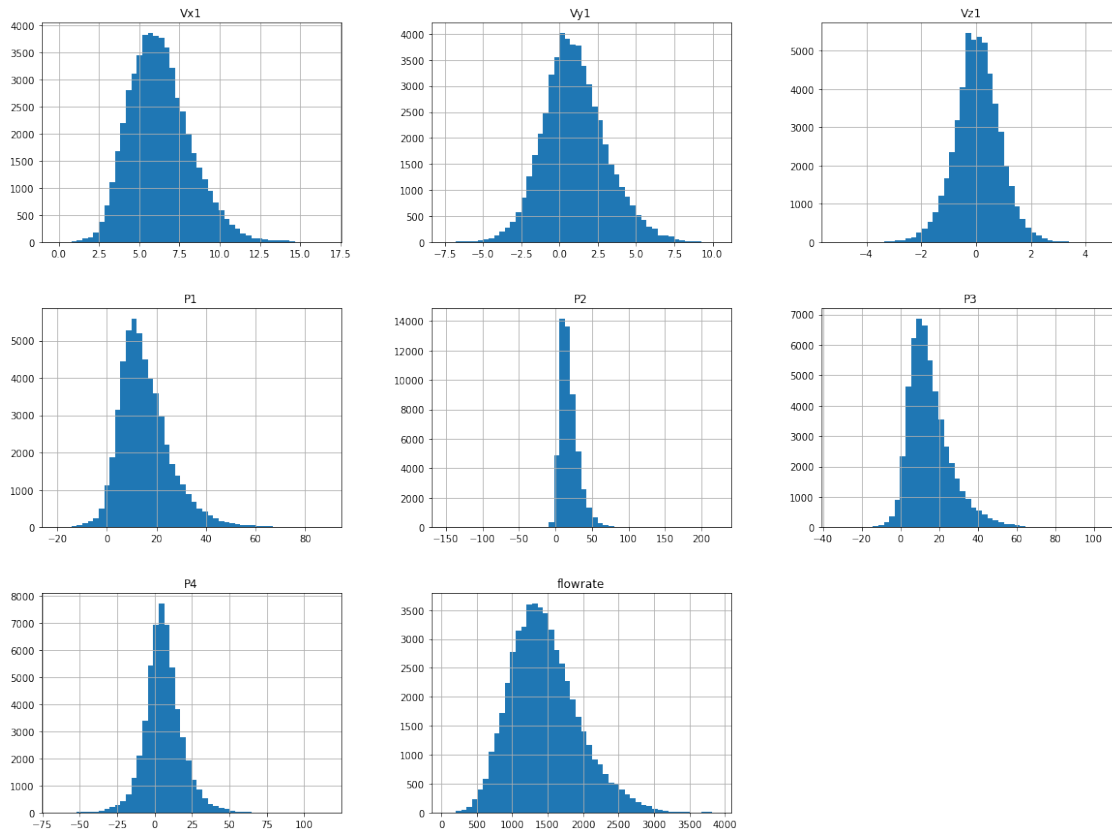
```
Unnamed: 0      0
Vx1             0
Vy1             0
Vz1             0
P1              0
P2              0
P3              0
P4              0
flowrate       22
dtype: int64
```

```
[3]: # Drop the index column
data = data.drop(['Unnamed: 0'], axis=1)

import matplotlib.pyplot as plt

# Perfrom Exploratory Data Analysis
data.hist(bins=50, figsize=(20,15))
```

```
[3]: array([[<AxesSubplot:title={'center':'Vx1'}>,
          <AxesSubplot:title={'center':'Vy1'}>,
          <AxesSubplot:title={'center':'Vz1'}>],
          [<AxesSubplot:title={'center':'P1'}>,
          <AxesSubplot:title={'center':'P2'}>,
          <AxesSubplot:title={'center':'P3'}>],
          [<AxesSubplot:title={'center':'P4'}>,
          <AxesSubplot:title={'center':'flowrate'}>, <AxesSubplot:>]],
      dtype=object)
```



```
[4]: data['flowrate'].describe(percentiles=[.1, .2, .3, .4, .5, .6, .7, .8, .9])
```

```
[4]: count    53069.000000
     mean      1459.690101
     std       482.004263
     min       48.613896
     10%       885.894840
     20%      1050.576868
     30%      1179.695183
     40%      1296.517302
     50%      1409.731610
     60%      1528.363684
     70%      1667.297100
     80%      1839.901344
     90%      2104.497622
     max      3896.046748
     Name: flowrate, dtype: float64
```

```
[5]: import numpy as np

     data["flowrate_cat"] = pd.cut(data["flowrate"],
```

```

        bins=[885.89, 1050.58, 1179.70, 1296.52, 1409.73,
↪1528.36, 1667.3, 1839.9, 2104.5, np.inf],
        right=False)

```

```
data["flowrate_cat"].value_counts()
```

```

[5]: [1528.36, 1667.3)      5308
      [2104.5, inf)       5307
      [1839.9, 2104.5)    5307
      [1667.3, 1839.9)    5307
      [1179.7, 1296.52)   5307
      [1050.58, 1179.7)   5307
      [885.89, 1050.58)   5307
      [1409.73, 1528.36)  5306
      [1296.52, 1409.73)  5306
      Name: flowrate_cat, dtype: int64

```

```

[6]: # Drop null values
data = data.dropna()

from sklearn.model_selection import train_test_split

# Split the data into training and test sets with stratification
X_train, X_test, y_train, y_test = train_test_split(data.drop(['flowrate',
↪"flowrate_cat"], axis=1), data['flowrate'], test_size=0.3, random_state=42,
↪stratify=data['flowrate_cat'])

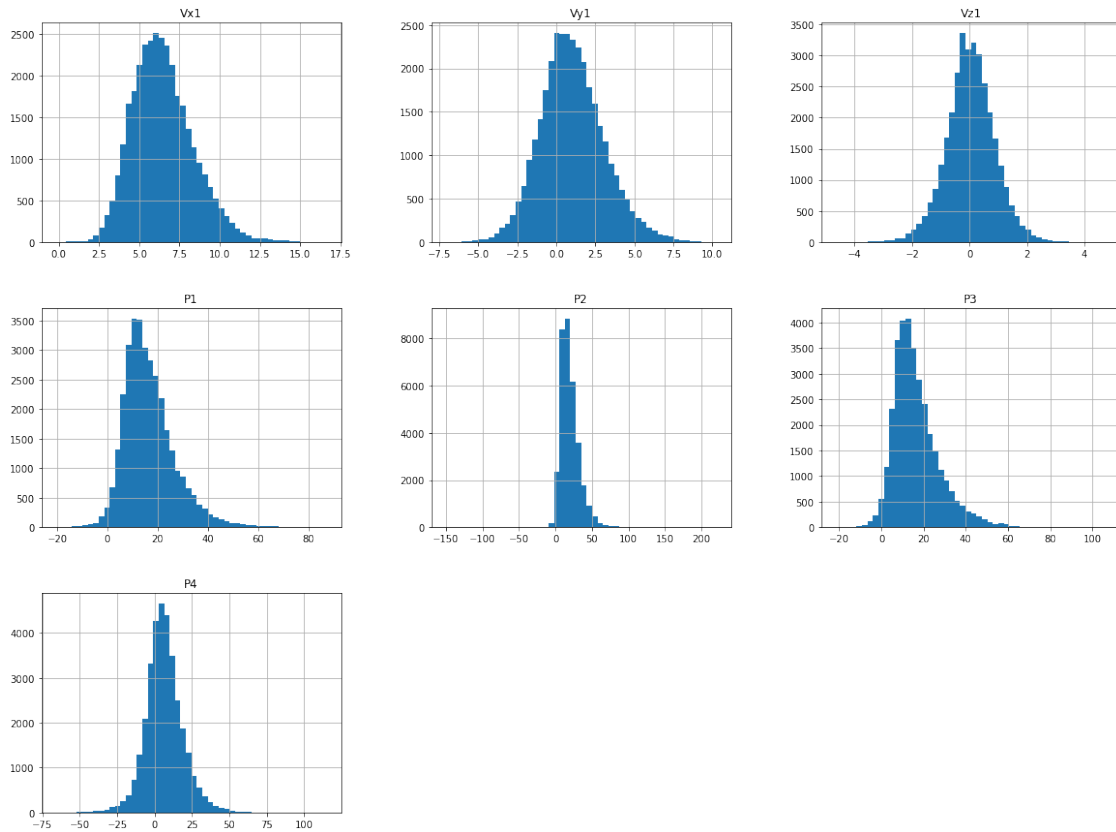
```

```
[7]: X_train.hist(bins=50, figsize=(20,15))
```

```

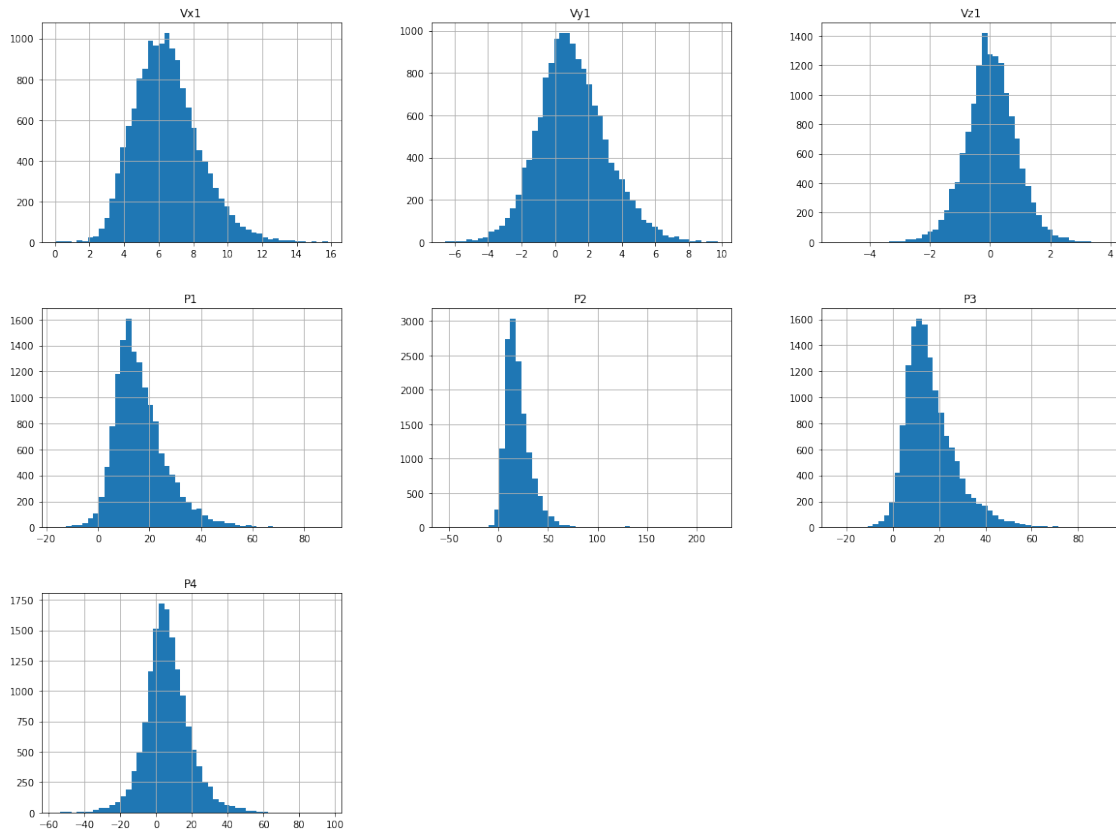
[7]: array([[<AxesSubplot:title={'center': 'Vx1'}>,
          <AxesSubplot:title={'center': 'Vy1'}>,
          <AxesSubplot:title={'center': 'Vz1'}>],
        [<AxesSubplot:title={'center': 'P1'}>,
          <AxesSubplot:title={'center': 'P2'}>,
          <AxesSubplot:title={'center': 'P3'}>],
        [<AxesSubplot:title={'center': 'P4'}>, <AxesSubplot:>,
          <AxesSubplot:>]], dtype=object)

```



```
[8]: X_test.hist(bins=50, figsize=(20,15))
```

```
[8]: array([[<AxesSubplot:title={'center':'Vx1'}>,
  <AxesSubplot:title={'center':'Vy1'}>,
  <AxesSubplot:title={'center':'Vz1'}>],
  [<AxesSubplot:title={'center':'P1'}>,
  <AxesSubplot:title={'center':'P2'}>,
  <AxesSubplot:title={'center':'P3'}>],
  [<AxesSubplot:title={'center':'P4'}>, <AxesSubplot:>,
  <AxesSubplot:>]], dtype=object)
```



```
[9]: # Artificial Neural Network with tensorflow
import tensorflow as tf
import math
from sklearn.model_selection import GridSearchCV
from keras.wrappers.scikit_learn import KerasRegressor

# Build the ANN model to predict the flowrate with normalization
def ANN(n_layers=2, n_neurons=16, optimizer='adam', epochs=50,
    ↪activation='relu', loss='mse'):
    model = tf.keras.models.Sequential()

    # Add the input layer and the first hidden layer
    model.add(tf.keras.layers.Dense(units=n_neurons, input_dim=X_train.
    ↪shape[1], activation=activation))

    # Add the rest of the hidden layers
    for i in range(1, n_layers):
        model.add(tf.keras.layers.Dense(units=n_neurons,
    ↪activation=activation))

    # Output layer
```

```

    model.add(tf.keras.layers.Dense(units=1, activation='linear'))
    model.compile(optimizer=optimizer, loss=loss)
    return model

# Wrap model into scikit-learn
model = KerasRegressor(build_fn=ANN, verbose = False)

# Use GridSearchCV to find the best parameters
param_grid = {
    'n_layers': [2, 4, 8],
    'n_neurons': [16, 32],
    'optimizer': ['adam'],
    'epochs': [50],
    'activation': ['relu'],
    'loss': ['mse']
}
ann_gs = GridSearchCV(estimator = model, param_grid = param_grid, cv=3,
    →scoring='neg_mean_squared_error')
ann_gs.fit(X_train, y_train, verbose=0)

# Evaluate the model
print("Best parameters set found on development set:")
print()
print(ann_gs.best_params_)
print()
print("Grid scores on development set:")
print()
means = ann_gs.cv_results_['mean_test_score']
stds = ann_gs.cv_results_['std_test_score']
for mean, std, params in zip(means, stds, ann_gs.cv_results_['params']):
    print("%0.3f (+/-%0.03f) for %r"
        % (mean, std * 2, params))
print()

```

```

2021-11-24 05:10:44.874264: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcudart.so.10.1
2021-11-24 05:10:48.047918: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcuda.so.1
2021-11-24 05:10:48.088023: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:48.088572: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1716] Found device 0 with
properties:

```

```

pciBusID: 0000:02:00.0 name: GeForce GTX 1060 6GB computeCapability: 6.1
coreClock: 1.7845GHz coreCount: 10 deviceMemorySize: 5.93GiB
deviceMemoryBandwidth: 178.99GiB/s
2021-11-24 05:10:48.088627: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcudart.so.10.1
2021-11-24 05:10:48.180379: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcublas.so.10
2021-11-24 05:10:48.203928: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcufft.so.10
2021-11-24 05:10:48.212692: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcurand.so.10
2021-11-24 05:10:48.246883: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcusolver.so.10
2021-11-24 05:10:48.259005: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcusparsparse.so.10
2021-11-24 05:10:48.407138: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcudnn.so.7
2021-11-24 05:10:48.407436: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:48.408190: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:48.408762: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1858] Adding visible gpu
devices: 0
2021-11-24 05:10:48.445121: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:104] CPU Frequency:
2194695000 Hz
2021-11-24 05:10:48.448290: I tensorflow/compiler/xla/service/service.cc:168]
XLA service 0x55ff33b3b5f0 initialized for platform Host (this does not
guarantee that XLA will be used). Devices:
2021-11-24 05:10:48.448346: I tensorflow/compiler/xla/service/service.cc:176]
StreamExecutor device (0): Host, Default Version
2021-11-24 05:10:48.708110: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:48.708771: I tensorflow/compiler/xla/service/service.cc:168]

```



```

XLA service 0x55ff3322ff30 initialized for platform CUDA (this does not
guarantee that XLA will be used). Devices:
2021-11-24 05:10:48.708806: I tensorflow/compiler/xla/service/service.cc:176]
StreamExecutor device (0): GeForce GTX 1060 6GB, Compute Capability 6.1
2021-11-24 05:10:48.710684: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:48.711681: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1716] Found device 0 with
properties:
pciBusID: 0000:02:00.0 name: GeForce GTX 1060 6GB computeCapability: 6.1
coreClock: 1.7845GHz coreCount: 10 deviceMemorySize: 5.93GiB
deviceMemoryBandwidth: 178.99GiB/s
2021-11-24 05:10:48.711758: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcudart.so.10.1
2021-11-24 05:10:48.711803: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcublas.so.10
2021-11-24 05:10:48.711843: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcufft.so.10
2021-11-24 05:10:48.711883: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcurand.so.10
2021-11-24 05:10:48.711927: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcusolver.so.10
2021-11-24 05:10:48.711968: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcusparsesparse.so.10
2021-11-24 05:10:48.712008: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcudnn.so.7
2021-11-24 05:10:48.712136: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:48.712705: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:48.713177: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1858] Adding visible gpu
devices: 0
2021-11-24 05:10:48.714090: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully

```

```

opened dynamic library libcudart.so.10.1
2021-11-24 05:10:50.526654: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1257] Device interconnect
StreamExecutor with strength 1 edge matrix:
2021-11-24 05:10:50.526722: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1263]      0
2021-11-24 05:10:50.526738: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1276] 0:   N
2021-11-24 05:10:50.529797: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:50.530430: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:982] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2021-11-24 05:10:50.530955: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1402] Created TensorFlow device
(/job:localhost/replica:0/task:0/device:GPU:0 with 3630 MB memory) -> physical
GPU (device: 0, name: GeForce GTX 1060 6GB, pci bus id: 0000:02:00.0, compute
capability: 6.1)
2021-11-24 05:10:51.729211: I
tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully
opened dynamic library libcublas.so.10

```

Best parameters set found on development set:

```
{'activation': 'relu', 'epochs': 50, 'loss': 'mse', 'n_layers': 4, 'n_neutrons':
32, 'optimizer': 'adam'}
```

Grid scores on development set:

```

-79492.971 (+/-2272.782) for {'activation': 'relu', 'epochs': 50, 'loss': 'mse',
'n_layers': 2, 'n_neutrons': 16, 'optimizer': 'adam'}
-77531.668 (+/-1864.953) for {'activation': 'relu', 'epochs': 50, 'loss': 'mse',
'n_layers': 2, 'n_neutrons': 32, 'optimizer': 'adam'}
-78054.932 (+/-1631.513) for {'activation': 'relu', 'epochs': 50, 'loss': 'mse',
'n_layers': 4, 'n_neutrons': 16, 'optimizer': 'adam'}
-75286.096 (+/-977.435) for {'activation': 'relu', 'epochs': 50, 'loss': 'mse',
'n_layers': 4, 'n_neutrons': 32, 'optimizer': 'adam'}
-77169.922 (+/-2892.412) for {'activation': 'relu', 'epochs': 50, 'loss': 'mse',
'n_layers': 8, 'n_neutrons': 16, 'optimizer': 'adam'}
-76202.422 (+/-1071.164) for {'activation': 'relu', 'epochs': 50, 'loss': 'mse',
'n_layers': 8, 'n_neutrons': 32, 'optimizer': 'adam'}

```

```
[12]: # Get the predictions
y_pred = ann_gs.predict(X_test)
```

```
# Plot the predictions and actual values
plt.figure(figsize=(20,10))
plt.plot(range(200), y_test[:200], color='blue', label='Actual')
plt.plot(range(200), y_pred[:200], color='red', label='Predicted')
plt.legend()
plt.show()
```

