

This notebook generates predictions on the test dataset

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

import torch
import torch.nn as nn

/home/candy/.local/lib/python3.10/site-packages/tqdm/auto.py:21:
TqdmWarning: IPProgress not found. Please update jupyter and
ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm

class Network(nn.Module):
    def __init__(self):
        super().__init__()
        self.layers = nn.Sequential(
            nn.Linear(30, 64),
            nn.ReLU(),
            nn.Dropout(0.15),
            nn.Linear(64, 128),
            nn.ReLU(),
            nn.Dropout(0.15),
            nn.Linear(128, 256),
            nn.ReLU(),
            nn.Dropout(0.15),
            nn.Linear(256, 128),
            nn.ReLU(),
            nn.Dropout(0.15),
            nn.Linear(128, 64),
            nn.ReLU(),
            nn.Dropout(0.15),
            nn.Linear(64, 32),
            nn.ReLU(),
            nn.Dropout(0.15),
            nn.Linear(32, 1),
            nn.Sigmoid()
        )

    def forward(self, x):
        return self.layers(x)

device = torch.device("cuda")
```

```

network = Network()
network.load_state_dict(torch.load('weights/final_weights.pth'))
network.to(device)

Network(
  (layers): Sequential(
    (0): Linear(in_features=30, out_features=64, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.15, inplace=False)
    (3): Linear(in_features=64, out_features=128, bias=True)
    (4): ReLU()
    (5): Dropout(p=0.15, inplace=False)
    (6): Linear(in_features=128, out_features=256, bias=True)
    (7): ReLU()
    (8): Dropout(p=0.15, inplace=False)
    (9): Linear(in_features=256, out_features=128, bias=True)
    (10): ReLU()
    (11): Dropout(p=0.15, inplace=False)
    (12): Linear(in_features=128, out_features=64, bias=True)
    (13): ReLU()
    (14): Dropout(p=0.15, inplace=False)
    (15): Linear(in_features=64, out_features=32, bias=True)
    (16): ReLU()
    (17): Dropout(p=0.15, inplace=False)
    (18): Linear(in_features=32, out_features=1, bias=True)
    (19): Sigmoid()
  )
)

train_data = pd.read_csv('csv_files/train.csv')
test_data = pd.read_csv('csv_files/test.csv')

train_data = train_data.drop(['Target'], axis=1)
test_data = test_data.drop(['ID'], axis=1)

scaler = StandardScaler()
scaler.fit(train_data)
test_data = scaler.transform(test_data)

test_data = torch.from_numpy(test_data).float().to(device)

preds = network(test_data)
preds = preds.round()
preds = preds.cpu().detach().numpy().astype(int).reshape(-1)

submission = pd.DataFrame({'ID': np.arange(1, 100001), 'Target':
preds})

# file saved to csv_files/submission.csv
submission.to_csv('csv_files/submission.csv', index=False)

```