IC50_Prediction_exclude_cells_Submission

May 13, 2025

1 Simple feed forward NN to predict IC50 based on patient demographic and molecular drug information (EXCLUDES CELL LINE FEATURE FROM TRAINING)

[2]:
```
from CHEM277B_functions import *
```

[4]:
```
valid_IC50s = pd.read_csv("data/valid_IC50s_within_range.csv")
merged_df = pd.read_csv("data/final_merged_imputed_race_CSV.csv")
lucas_df = pd.read_csv("data/Raw_Mapping_Imputed_Race.csv")
cell_lines_df = pd.read_csv("data/HarvardCellLines.csv")
```

```
/var/folders/s8/ghqk1l4n7n9_w17t7hx21g2m0000gn/T/ipykernel_13056/1684730604.py:2
: DtypeWarning: Columns (3,4,5,6,7,9,12,15,16,17,18,22,24,26,28,29,30,67) have
mixed types. Specify dtype option on import or set low_memory=False.
  merged_df = pd.read_csv("final_merged_imputed_race_CSV.csv")
```

[6]:
```
valid_IC50s.drop(columns = ['Unnamed: 0', 'N Points'], inplace = True)
```

[8]:
```
cell_lines_df.columns
```

[8]:
```
Index(['HMS LINCS Batch ID', 'HMS LINCS ID', 'Name', 'Alternative Names',
       'LINCS ID', 'Alternative ID', 'Reference Source', 'Organism', 'Organ',
       'Tissue', 'Cell Type', 'Details of Cell Type', 'Donor Sex', 'Donor Age',
       'Donor Ethnicity', 'Donor Health Status', 'Disease', 'Unnamed: 17',
       'Details of Disease', 'Production Details', 'Genetic Modification(s)',
       'Known Mutations', 'Citation Information for Mutations',
       'Verification Reference Profile', 'Growth Properties',
       'Recommended Culture Conditions', 'Relevant Citations', 'Usage Note',
       'Comments', 'Provider', 'Provider Catalog ID', 'Provider Batch ID',
       'Source Information', 'Date Received', 'HMS QC Outcome',
       'Transient Modification(s)', 'Date Publicly Available',
       'Most Recent Update', 'T Stage'],
      dtype='object')
```

[10]:
```
columns = ["HMS LINCS Batch ID", "Name", "T Stage"]
cell_lines_df = cell_lines_df[columns]
```

```
[12]: merged_df['T_stage_by_size'] = merged_df.apply(lambda row: row['T Stage'] if pd.
      ↪notnull(row['T Stage']) else T_stage_by_size(row['Tumor Size']), axis=1)
```

2 USE LUCAS' DATASET STARTING HERE

```
[15]: columns = ['Age', 'Race', 'T_stage_by_size']
      patients_df = merged_df[columns]
```

```
[19]: patients_df["cell_lines"] = lucas_df["Matched Cell Line ID"]
```

```
/var/folders/s8/ghqk114n7n9_w17t7hx21g2m0000gn/T/ipykernel_13056/3231217547.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  patients_df["cell_lines"] = lucas_df["Matched Cell Line ID"]
```

```
[21]: cell_lines_df_dict = cell_lines_df.set_index("HMS LINCS Batch ID").
      ↪to_dict()["Name"]
```

```
[27]: patients_df["Cell Name"] = patients_df["cell_lines"].apply(mapping, dictionary␣
      ↪= cell_lines_df_dict)
```

```
/var/folders/s8/ghqk114n7n9_w17t7hx21g2m0000gn/T/ipykernel_13056/1850949227.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  patients_df["Cell Name"] = patients_df["cell_lines"].apply(mapping, dictionary
= cell_lines_df_dict)
```

```
[31]: patients_df.dropna(inplace=True)
      patients_df.isna().sum()
```

```
/var/folders/s8/ghqk114n7n9_w17t7hx21g2m0000gn/T/ipykernel_13056/2483854490.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  patients_df.dropna(inplace=True)
```

```
[31]: Age                0
      Race               0
      T_stage_by_size    0
      cell_lines         0
      Cell Name          0
      dtype: int64
```

3 Add Small Molecule descriptors for each row.

```
[33]: drugs_df = pd.read_csv("Descriptors_Small_Molecules.csv")
      drugs_df = drugs_df[['Name', 'Molecular Mass', 'LogP', 'NumHDonors',␣
       ↪'NumHAcceptors', 'TPSA']]
      drugs_df.rename(columns={'Name': 'Small Molecule Name'}, inplace=True)
      drugs_df.head()
```

```
[33]:   Small Molecule Name  Molecular Mass     LogP  NumHDonors  NumHAcceptors  \
      0             AZD7762          362.12  2.52660           4              4
      1           Neratinib          556.20  5.93248           2              8
      2           Dasatinib          487.16  3.31354           3              9
      3         Saracatinib          541.21  3.93950           1             10
      4          Pictilisib          513.16  2.14840           1              9

           TPSA
      0   96.25
      1  112.40
      2  106.51
      3   90.44
      4  107.55
```

```
[35]: valid_IC50s = pd.merge(valid_IC50s, drugs_df, on='Small Molecule Name',␣
       ↪how='left')
```

```
[37]: patient_drug_df = pd.merge(patients_df, valid_IC50s, on='Cell Name', how='left')
```

```
[39]: patient_drug_df.drop(columns = "cell_lines", inplace=True)
```

```
[41]: patient_drug_df.to_csv("patient_drug_information_aka_final_final_final.csv")
```

```
[43]: race_dict = {0:'N/A', 1:"white", 2:"black", 3:"asian", 4:"native", 5:
       ↪"hispanic", 6:"multi", 7:"hawa", 8:"amer indian"}
```

```
[45]: patient_drug_df["Race"] = patient_drug_df["Race"].apply(mapping, dictionary =␣
       ↪race_dict)
```

3.0.1 Cell lines are proxy for patients, so drop the cell lines column.

[47]:
```
patient_drug_df.drop(columns="Cell Name", inplace=True)
```

3.0.2 Need to one-hot-encode categorical featurs

[50]:
```
categorical_cols = ['Race', 'T_stage_by_size', 'Small Molecule Name']
patient_drug_df[categorical_cols] = patient_drug_df[categorical_cols].
    ↪astype('category')
```

[54]:
```
patient_drug_df_encoded = pd.get_dummies(patient_drug_df,␣
    ↪columns=categorical_cols, drop_first=True)
```

[56]:
```
bool_cols = patient_drug_df_encoded.select_dtypes(include='bool').columns
patient_drug_df_encoded[bool_cols] = patient_drug_df_encoded[bool_cols].
    ↪astype(int)
```

3.0.3 PRE-PROCESSING DONE. ONTO MODEL TRAINING.

[58]:
```
# Separate features (X) and target (y)
X = patient_drug_df_encoded.drop(columns=['EC50 (uM)'])  # Drop EC50 and␣
    ↪non-features
y = patient_drug_df_encoded['EC50 (uM)']  # EC50 is the target

# Standardize numerical features
scaler = StandardScaler()
X[['Age', 'Molecular Mass', 'LogP', 'NumHDonors', 'NumHAcceptors', 'TPSA']] =␣
    ↪scaler.fit_transform(X[['Age', 'Molecular Mass', 'LogP', 'NumHDonors',␣
    ↪'NumHAcceptors', 'TPSA']])
```

[60]:
```
# Train/Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
    ↪random_state=42)
```

[62]:
```
# Data
# Select drug descriptor columns
drug_feature_cols = ['Molecular Mass', 'LogP', 'NumHDonors', 'NumHAcceptors',␣
    ↪'TPSA']
drug_features = X_train[drug_feature_cols].to_numpy()  # (n_samples, 5)

# Drop drug features to get patient features
patient_features = X_train.drop(columns=drug_feature_cols).to_numpy()

# Convert to tensors
drug_tensor = torch.tensor(drug_features).float()
patient_tensor = torch.tensor(patient_features).float()
ic50_tensor = torch.tensor(y_train.to_numpy()).float()
```

```python
model = DrugResponsePredictor(drug_input_dim=5,␣
 ↪patient_input_dim=patient_tensor.shape[1])
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
loss_fn = nn.MSELoss()


# Training loop
model.train()
start_time = time.time()

my_epochs = []
my_losses = []

for epoch in range(300):
    optimizer.zero_grad()
    preds = model(drug_tensor, patient_tensor)
    loss = loss_fn(preds, ic50_tensor)
    loss.backward()
    optimizer.step()
    end_time = time.time()
    elapsed = end_time - start_time
    print(f"Epoch {epoch} | Loss: {loss.item():.4f}")
    print(f"Time elapsed: {elapsed:.2f} seconds")
    my_epochs.append(epoch)
    my_losses.append(loss.item())
```

```
Epoch 0 | Loss: 2.9137
Time elapsed: 0.06 seconds
Epoch 1 | Loss: 2.8801
Time elapsed: 0.09 seconds
Epoch 2 | Loss: 2.8484
Time elapsed: 0.12 seconds
Epoch 3 | Loss: 2.8184
Time elapsed: 0.15 seconds
Epoch 4 | Loss: 2.7895
Time elapsed: 0.18 seconds
Epoch 5 | Loss: 2.7615
Time elapsed: 0.21 seconds
Epoch 6 | Loss: 2.7342
Time elapsed: 0.24 seconds
Epoch 7 | Loss: 2.7071
Time elapsed: 0.27 seconds
Epoch 8 | Loss: 2.6792
Time elapsed: 0.30 seconds
Epoch 9 | Loss: 2.6506
Time elapsed: 0.33 seconds
```

```
Epoch 10 | Loss: 2.6215
Time elapsed: 0.35 seconds
Epoch 11 | Loss: 2.5919
Time elapsed: 0.38 seconds
Epoch 12 | Loss: 2.5618
Time elapsed: 0.41 seconds
Epoch 13 | Loss: 2.5315
Time elapsed: 0.44 seconds
Epoch 14 | Loss: 2.5010
Time elapsed: 0.47 seconds
Epoch 15 | Loss: 2.4702
Time elapsed: 0.50 seconds
Epoch 16 | Loss: 2.4388
Time elapsed: 0.53 seconds
Epoch 17 | Loss: 2.4071
Time elapsed: 0.56 seconds
Epoch 18 | Loss: 2.3752
Time elapsed: 0.59 seconds
Epoch 19 | Loss: 2.3433
Time elapsed: 0.62 seconds
Epoch 20 | Loss: 2.3113
Time elapsed: 0.65 seconds
Epoch 21 | Loss: 2.2792
Time elapsed: 0.68 seconds
Epoch 22 | Loss: 2.2474
Time elapsed: 0.71 seconds
Epoch 23 | Loss: 2.2158
Time elapsed: 0.74 seconds
Epoch 24 | Loss: 2.1849
Time elapsed: 0.77 seconds
Epoch 25 | Loss: 2.1552
Time elapsed: 0.80 seconds
Epoch 26 | Loss: 2.1274
Time elapsed: 0.83 seconds
Epoch 27 | Loss: 2.1022
Time elapsed: 0.85 seconds
Epoch 28 | Loss: 2.0798
Time elapsed: 0.88 seconds
Epoch 29 | Loss: 2.0607
Time elapsed: 0.91 seconds
Epoch 30 | Loss: 2.0448
Time elapsed: 0.94 seconds
Epoch 31 | Loss: 2.0324
Time elapsed: 0.97 seconds
Epoch 32 | Loss: 2.0233
Time elapsed: 1.00 seconds
Epoch 33 | Loss: 2.0173
Time elapsed: 1.03 seconds
```

```
Epoch 34 | Loss: 2.0139
Time elapsed: 1.06 seconds
Epoch 35 | Loss: 2.0119
Time elapsed: 1.09 seconds
Epoch 36 | Loss: 2.0106
Time elapsed: 1.12 seconds
Epoch 37 | Loss: 2.0088
Time elapsed: 1.15 seconds
Epoch 38 | Loss: 2.0056
Time elapsed: 1.18 seconds
Epoch 39 | Loss: 2.0005
Time elapsed: 1.21 seconds
Epoch 40 | Loss: 1.9935
Time elapsed: 1.24 seconds
Epoch 41 | Loss: 1.9847
Time elapsed: 1.26 seconds
Epoch 42 | Loss: 1.9747
Time elapsed: 1.29 seconds
Epoch 43 | Loss: 1.9638
Time elapsed: 1.32 seconds
Epoch 44 | Loss: 1.9525
Time elapsed: 1.35 seconds
Epoch 45 | Loss: 1.9412
Time elapsed: 1.38 seconds
Epoch 46 | Loss: 1.9303
Time elapsed: 1.41 seconds
Epoch 47 | Loss: 1.9200
Time elapsed: 1.44 seconds
Epoch 48 | Loss: 1.9102
Time elapsed: 1.47 seconds
Epoch 49 | Loss: 1.9010
Time elapsed: 1.50 seconds
Epoch 50 | Loss: 1.8922
Time elapsed: 1.53 seconds
Epoch 51 | Loss: 1.8836
Time elapsed: 1.56 seconds
Epoch 52 | Loss: 1.8751
Time elapsed: 1.58 seconds
Epoch 53 | Loss: 1.8665
Time elapsed: 1.62 seconds
Epoch 54 | Loss: 1.8576
Time elapsed: 1.64 seconds
Epoch 55 | Loss: 1.8483
Time elapsed: 1.67 seconds
Epoch 56 | Loss: 1.8384
Time elapsed: 1.70 seconds
Epoch 57 | Loss: 1.8280
Time elapsed: 1.73 seconds
```

```
Epoch 58 | Loss: 1.8171
Time elapsed: 1.76 seconds
Epoch 59 | Loss: 1.8055
Time elapsed: 1.79 seconds
Epoch 60 | Loss: 1.7933
Time elapsed: 1.82 seconds
Epoch 61 | Loss: 1.7807
Time elapsed: 1.85 seconds
Epoch 62 | Loss: 1.7674
Time elapsed: 1.88 seconds
Epoch 63 | Loss: 1.7538
Time elapsed: 1.91 seconds
Epoch 64 | Loss: 1.7398
Time elapsed: 1.94 seconds
Epoch 65 | Loss: 1.7254
Time elapsed: 1.96 seconds
Epoch 66 | Loss: 1.7107
Time elapsed: 1.99 seconds
Epoch 67 | Loss: 1.6954
Time elapsed: 2.02 seconds
Epoch 68 | Loss: 1.6797
Time elapsed: 2.05 seconds
Epoch 69 | Loss: 1.6632
Time elapsed: 2.08 seconds
Epoch 70 | Loss: 1.6460
Time elapsed: 2.11 seconds
Epoch 71 | Loss: 1.6279
Time elapsed: 2.14 seconds
Epoch 72 | Loss: 1.6088
Time elapsed: 2.17 seconds
Epoch 73 | Loss: 1.5887
Time elapsed: 2.20 seconds
Epoch 74 | Loss: 1.5676
Time elapsed: 2.23 seconds
Epoch 75 | Loss: 1.5456
Time elapsed: 2.26 seconds
Epoch 76 | Loss: 1.5226
Time elapsed: 2.29 seconds
Epoch 77 | Loss: 1.4988
Time elapsed: 2.32 seconds
Epoch 78 | Loss: 1.4741
Time elapsed: 2.34 seconds
Epoch 79 | Loss: 1.4488
Time elapsed: 2.37 seconds
Epoch 80 | Loss: 1.4226
Time elapsed: 2.40 seconds
Epoch 81 | Loss: 1.3957
Time elapsed: 2.43 seconds
```

```
Epoch 82 | Loss: 1.3679
Time elapsed: 2.46 seconds
Epoch 83 | Loss: 1.3396
Time elapsed: 2.49 seconds
Epoch 84 | Loss: 1.3106
Time elapsed: 2.52 seconds
Epoch 85 | Loss: 1.2811
Time elapsed: 2.55 seconds
Epoch 86 | Loss: 1.2514
Time elapsed: 2.58 seconds
Epoch 87 | Loss: 1.2215
Time elapsed: 2.61 seconds
Epoch 88 | Loss: 1.1919
Time elapsed: 2.64 seconds
Epoch 89 | Loss: 1.1626
Time elapsed: 2.67 seconds
Epoch 90 | Loss: 1.1338
Time elapsed: 2.70 seconds
Epoch 91 | Loss: 1.1057
Time elapsed: 2.73 seconds
Epoch 92 | Loss: 1.0785
Time elapsed: 2.76 seconds
Epoch 93 | Loss: 1.0522
Time elapsed: 2.79 seconds
Epoch 94 | Loss: 1.0273
Time elapsed: 2.82 seconds
Epoch 95 | Loss: 1.0037
Time elapsed: 2.84 seconds
Epoch 96 | Loss: 0.9816
Time elapsed: 2.87 seconds
Epoch 97 | Loss: 0.9610
Time elapsed: 2.90 seconds
Epoch 98 | Loss: 0.9418
Time elapsed: 2.93 seconds
Epoch 99 | Loss: 0.9242
Time elapsed: 2.96 seconds
Epoch 100 | Loss: 0.9079
Time elapsed: 2.99 seconds
Epoch 101 | Loss: 0.8930
Time elapsed: 3.02 seconds
Epoch 102 | Loss: 0.8792
Time elapsed: 3.05 seconds
Epoch 103 | Loss: 0.8666
Time elapsed: 3.08 seconds
Epoch 104 | Loss: 0.8548
Time elapsed: 3.11 seconds
Epoch 105 | Loss: 0.8438
Time elapsed: 3.14 seconds
```

```
Epoch 106 | Loss: 0.8334
Time elapsed: 3.17 seconds
Epoch 107 | Loss: 0.8234
Time elapsed: 3.19 seconds
Epoch 108 | Loss: 0.8137
Time elapsed: 3.22 seconds
Epoch 109 | Loss: 0.8042
Time elapsed: 3.25 seconds
Epoch 110 | Loss: 0.7948
Time elapsed: 3.28 seconds
Epoch 111 | Loss: 0.7857
Time elapsed: 3.31 seconds
Epoch 112 | Loss: 0.7768
Time elapsed: 3.34 seconds
Epoch 113 | Loss: 0.7681
Time elapsed: 3.37 seconds
Epoch 114 | Loss: 0.7595
Time elapsed: 3.40 seconds
Epoch 115 | Loss: 0.7512
Time elapsed: 3.43 seconds
Epoch 116 | Loss: 0.7430
Time elapsed: 3.46 seconds
Epoch 117 | Loss: 0.7350
Time elapsed: 3.49 seconds
Epoch 118 | Loss: 0.7272
Time elapsed: 3.52 seconds
Epoch 119 | Loss: 0.7197
Time elapsed: 3.55 seconds
Epoch 120 | Loss: 0.7123
Time elapsed: 3.58 seconds
Epoch 121 | Loss: 0.7051
Time elapsed: 3.61 seconds
Epoch 122 | Loss: 0.6982
Time elapsed: 3.63 seconds
Epoch 123 | Loss: 0.6914
Time elapsed: 3.66 seconds
Epoch 124 | Loss: 0.6847
Time elapsed: 3.69 seconds
Epoch 125 | Loss: 0.6782
Time elapsed: 3.72 seconds
Epoch 126 | Loss: 0.6718
Time elapsed: 3.75 seconds
Epoch 127 | Loss: 0.6654
Time elapsed: 3.78 seconds
Epoch 128 | Loss: 0.6591
Time elapsed: 3.81 seconds
Epoch 129 | Loss: 0.6529
Time elapsed: 3.84 seconds
```

```
Epoch 130 | Loss: 0.6468
Time elapsed: 3.87 seconds
Epoch 131 | Loss: 0.6408
Time elapsed: 3.90 seconds
Epoch 132 | Loss: 0.6349
Time elapsed: 3.93 seconds
Epoch 133 | Loss: 0.6290
Time elapsed: 3.95 seconds
Epoch 134 | Loss: 0.6233
Time elapsed: 3.98 seconds
Epoch 135 | Loss: 0.6176
Time elapsed: 4.01 seconds
Epoch 136 | Loss: 0.6119
Time elapsed: 4.04 seconds
Epoch 137 | Loss: 0.6063
Time elapsed: 4.07 seconds
Epoch 138 | Loss: 0.6008
Time elapsed: 4.10 seconds
Epoch 139 | Loss: 0.5953
Time elapsed: 4.13 seconds
Epoch 140 | Loss: 0.5899
Time elapsed: 4.16 seconds
Epoch 141 | Loss: 0.5845
Time elapsed: 4.19 seconds
Epoch 142 | Loss: 0.5793
Time elapsed: 4.22 seconds
Epoch 143 | Loss: 0.5741
Time elapsed: 4.25 seconds
Epoch 144 | Loss: 0.5690
Time elapsed: 4.28 seconds
Epoch 145 | Loss: 0.5640
Time elapsed: 4.31 seconds
Epoch 146 | Loss: 0.5591
Time elapsed: 4.34 seconds
Epoch 147 | Loss: 0.5542
Time elapsed: 4.37 seconds
Epoch 148 | Loss: 0.5495
Time elapsed: 4.40 seconds
Epoch 149 | Loss: 0.5448
Time elapsed: 4.43 seconds
Epoch 150 | Loss: 0.5403
Time elapsed: 4.45 seconds
Epoch 151 | Loss: 0.5359
Time elapsed: 4.48 seconds
Epoch 152 | Loss: 0.5316
Time elapsed: 4.51 seconds
Epoch 153 | Loss: 0.5274
Time elapsed: 4.54 seconds
```

```
Epoch 154 | Loss: 0.5234
Time elapsed: 4.57 seconds
Epoch 155 | Loss: 0.5195
Time elapsed: 4.60 seconds
Epoch 156 | Loss: 0.5156
Time elapsed: 4.63 seconds
Epoch 157 | Loss: 0.5119
Time elapsed: 4.66 seconds
Epoch 158 | Loss: 0.5082
Time elapsed: 4.69 seconds
Epoch 159 | Loss: 0.5046
Time elapsed: 4.72 seconds
Epoch 160 | Loss: 0.5011
Time elapsed: 4.75 seconds
Epoch 161 | Loss: 0.4977
Time elapsed: 4.78 seconds
Epoch 162 | Loss: 0.4944
Time elapsed: 4.80 seconds
Epoch 163 | Loss: 0.4912
Time elapsed: 4.83 seconds
Epoch 164 | Loss: 0.4881
Time elapsed: 4.86 seconds
Epoch 165 | Loss: 0.4851
Time elapsed: 4.89 seconds
Epoch 166 | Loss: 0.4822
Time elapsed: 4.92 seconds
Epoch 167 | Loss: 0.4793
Time elapsed: 4.95 seconds
Epoch 168 | Loss: 0.4765
Time elapsed: 4.98 seconds
Epoch 169 | Loss: 0.4738
Time elapsed: 5.01 seconds
Epoch 170 | Loss: 0.4712
Time elapsed: 5.04 seconds
Epoch 171 | Loss: 0.4687
Time elapsed: 5.07 seconds
Epoch 172 | Loss: 0.4662
Time elapsed: 5.10 seconds
Epoch 173 | Loss: 0.4638
Time elapsed: 5.13 seconds
Epoch 174 | Loss: 0.4614
Time elapsed: 5.16 seconds
Epoch 175 | Loss: 0.4591
Time elapsed: 5.19 seconds
Epoch 176 | Loss: 0.4568
Time elapsed: 5.22 seconds
Epoch 177 | Loss: 0.4545
Time elapsed: 5.24 seconds
```

```
Epoch 178 | Loss: 0.4523
Time elapsed: 5.27 seconds
Epoch 179 | Loss: 0.4501
Time elapsed: 5.30 seconds
Epoch 180 | Loss: 0.4480
Time elapsed: 5.33 seconds
Epoch 181 | Loss: 0.4458
Time elapsed: 5.36 seconds
Epoch 182 | Loss: 0.4437
Time elapsed: 5.39 seconds
Epoch 183 | Loss: 0.4416
Time elapsed: 5.42 seconds
Epoch 184 | Loss: 0.4396
Time elapsed: 5.45 seconds
Epoch 185 | Loss: 0.4375
Time elapsed: 5.48 seconds
Epoch 186 | Loss: 0.4355
Time elapsed: 5.51 seconds
Epoch 187 | Loss: 0.4336
Time elapsed: 5.54 seconds
Epoch 188 | Loss: 0.4316
Time elapsed: 5.57 seconds
Epoch 189 | Loss: 0.4296
Time elapsed: 5.60 seconds
Epoch 190 | Loss: 0.4276
Time elapsed: 5.62 seconds
Epoch 191 | Loss: 0.4255
Time elapsed: 5.65 seconds
Epoch 192 | Loss: 0.4235
Time elapsed: 5.68 seconds
Epoch 193 | Loss: 0.4215
Time elapsed: 5.71 seconds
Epoch 194 | Loss: 0.4197
Time elapsed: 5.74 seconds
Epoch 195 | Loss: 0.4178
Time elapsed: 5.77 seconds
Epoch 196 | Loss: 0.4159
Time elapsed: 5.80 seconds
Epoch 197 | Loss: 0.4141
Time elapsed: 5.83 seconds
Epoch 198 | Loss: 0.4122
Time elapsed: 5.86 seconds
Epoch 199 | Loss: 0.4104
Time elapsed: 5.89 seconds
Epoch 200 | Loss: 0.4086
Time elapsed: 5.92 seconds
Epoch 201 | Loss: 0.4067
Time elapsed: 5.95 seconds
```

```
Epoch 202 | Loss: 0.4049
Time elapsed: 5.98 seconds
Epoch 203 | Loss: 0.4032
Time elapsed: 6.01 seconds
Epoch 204 | Loss: 0.4014
Time elapsed: 6.03 seconds
Epoch 205 | Loss: 0.3996
Time elapsed: 6.06 seconds
Epoch 206 | Loss: 0.3978
Time elapsed: 6.09 seconds
Epoch 207 | Loss: 0.3961
Time elapsed: 6.12 seconds
Epoch 208 | Loss: 0.3943
Time elapsed: 6.15 seconds
Epoch 209 | Loss: 0.3926
Time elapsed: 6.18 seconds
Epoch 210 | Loss: 0.3908
Time elapsed: 6.21 seconds
Epoch 211 | Loss: 0.3891
Time elapsed: 6.24 seconds
Epoch 212 | Loss: 0.3873
Time elapsed: 6.27 seconds
Epoch 213 | Loss: 0.3856
Time elapsed: 6.30 seconds
Epoch 214 | Loss: 0.3839
Time elapsed: 6.33 seconds
Epoch 215 | Loss: 0.3822
Time elapsed: 6.36 seconds
Epoch 216 | Loss: 0.3805
Time elapsed: 6.39 seconds
Epoch 217 | Loss: 0.3789
Time elapsed: 6.42 seconds
Epoch 218 | Loss: 0.3773
Time elapsed: 6.45 seconds
Epoch 219 | Loss: 0.3757
Time elapsed: 6.48 seconds
Epoch 220 | Loss: 0.3741
Time elapsed: 6.51 seconds
Epoch 221 | Loss: 0.3725
Time elapsed: 6.53 seconds
Epoch 222 | Loss: 0.3709
Time elapsed: 6.56 seconds
Epoch 223 | Loss: 0.3694
Time elapsed: 6.59 seconds
Epoch 224 | Loss: 0.3679
Time elapsed: 6.62 seconds
Epoch 225 | Loss: 0.3663
Time elapsed: 6.65 seconds
```

```
Epoch 226 | Loss: 0.3648
Time elapsed: 6.68 seconds
Epoch 227 | Loss: 0.3634
Time elapsed: 6.71 seconds
Epoch 228 | Loss: 0.3619
Time elapsed: 6.74 seconds
Epoch 229 | Loss: 0.3603
Time elapsed: 6.77 seconds
Epoch 230 | Loss: 0.3588
Time elapsed: 6.80 seconds
Epoch 231 | Loss: 0.3573
Time elapsed: 6.83 seconds
Epoch 232 | Loss: 0.3559
Time elapsed: 6.86 seconds
Epoch 233 | Loss: 0.3545
Time elapsed: 6.89 seconds
Epoch 234 | Loss: 0.3531
Time elapsed: 6.91 seconds
Epoch 235 | Loss: 0.3516
Time elapsed: 6.94 seconds
Epoch 236 | Loss: 0.3502
Time elapsed: 6.97 seconds
Epoch 237 | Loss: 0.3488
Time elapsed: 7.00 seconds
Epoch 238 | Loss: 0.3474
Time elapsed: 7.03 seconds
Epoch 239 | Loss: 0.3460
Time elapsed: 7.06 seconds
Epoch 240 | Loss: 0.3446
Time elapsed: 7.09 seconds
Epoch 241 | Loss: 0.3433
Time elapsed: 7.12 seconds
Epoch 242 | Loss: 0.3420
Time elapsed: 7.15 seconds
Epoch 243 | Loss: 0.3406
Time elapsed: 7.18 seconds
Epoch 244 | Loss: 0.3394
Time elapsed: 7.21 seconds
Epoch 245 | Loss: 0.3381
Time elapsed: 7.23 seconds
Epoch 246 | Loss: 0.3368
Time elapsed: 7.26 seconds
Epoch 247 | Loss: 0.3355
Time elapsed: 7.29 seconds
Epoch 248 | Loss: 0.3343
Time elapsed: 7.32 seconds
Epoch 249 | Loss: 0.3330
Time elapsed: 7.35 seconds
```

```
Epoch 250 | Loss: 0.3318
Time elapsed: 7.38 seconds
Epoch 251 | Loss: 0.3305
Time elapsed: 7.41 seconds
Epoch 252 | Loss: 0.3293
Time elapsed: 7.44 seconds
Epoch 253 | Loss: 0.3281
Time elapsed: 7.47 seconds
Epoch 254 | Loss: 0.3269
Time elapsed: 7.50 seconds
Epoch 255 | Loss: 0.3257
Time elapsed: 7.53 seconds
Epoch 256 | Loss: 0.3245
Time elapsed: 7.56 seconds
Epoch 257 | Loss: 0.3233
Time elapsed: 7.58 seconds
Epoch 258 | Loss: 0.3222
Time elapsed: 7.61 seconds
Epoch 259 | Loss: 0.3210
Time elapsed: 7.64 seconds
Epoch 260 | Loss: 0.3199
Time elapsed: 7.67 seconds
Epoch 261 | Loss: 0.3187
Time elapsed: 7.70 seconds
Epoch 262 | Loss: 0.3176
Time elapsed: 7.73 seconds
Epoch 263 | Loss: 0.3165
Time elapsed: 7.76 seconds
Epoch 264 | Loss: 0.3154
Time elapsed: 7.79 seconds
Epoch 265 | Loss: 0.3144
Time elapsed: 7.82 seconds
Epoch 266 | Loss: 0.3133
Time elapsed: 7.85 seconds
Epoch 267 | Loss: 0.3122
Time elapsed: 7.88 seconds
Epoch 268 | Loss: 0.3112
Time elapsed: 7.91 seconds
Epoch 269 | Loss: 0.3102
Time elapsed: 7.94 seconds
Epoch 270 | Loss: 0.3091
Time elapsed: 7.96 seconds
Epoch 271 | Loss: 0.3081
Time elapsed: 7.99 seconds
Epoch 272 | Loss: 0.3071
Time elapsed: 8.02 seconds
Epoch 273 | Loss: 0.3061
Time elapsed: 8.05 seconds
```

```
Epoch 274 | Loss: 0.3051
Time elapsed: 8.08 seconds
Epoch 275 | Loss: 0.3041
Time elapsed: 8.11 seconds
Epoch 276 | Loss: 0.3030
Time elapsed: 8.14 seconds
Epoch 277 | Loss: 0.3020
Time elapsed: 8.17 seconds
Epoch 278 | Loss: 0.3010
Time elapsed: 8.20 seconds
Epoch 279 | Loss: 0.2999
Time elapsed: 8.23 seconds
Epoch 280 | Loss: 0.2989
Time elapsed: 8.26 seconds
Epoch 281 | Loss: 0.2979
Time elapsed: 8.28 seconds
Epoch 282 | Loss: 0.2969
Time elapsed: 8.31 seconds
Epoch 283 | Loss: 0.2959
Time elapsed: 8.34 seconds
Epoch 284 | Loss: 0.2949
Time elapsed: 8.37 seconds
Epoch 285 | Loss: 0.2939
Time elapsed: 8.40 seconds
Epoch 286 | Loss: 0.2929
Time elapsed: 8.43 seconds
Epoch 287 | Loss: 0.2919
Time elapsed: 8.46 seconds
Epoch 288 | Loss: 0.2909
Time elapsed: 8.49 seconds
Epoch 289 | Loss: 0.2899
Time elapsed: 8.52 seconds
Epoch 290 | Loss: 0.2889
Time elapsed: 8.55 seconds
Epoch 291 | Loss: 0.2879
Time elapsed: 8.58 seconds
Epoch 292 | Loss: 0.2869
Time elapsed: 8.61 seconds
Epoch 293 | Loss: 0.2859
Time elapsed: 8.64 seconds
Epoch 294 | Loss: 0.2849
Time elapsed: 8.67 seconds
Epoch 295 | Loss: 0.2839
Time elapsed: 8.70 seconds
Epoch 296 | Loss: 0.2829
Time elapsed: 8.73 seconds
Epoch 297 | Loss: 0.2819
Time elapsed: 8.76 seconds
```

```
Epoch 298 | Loss: 0.2808
Time elapsed: 8.78 seconds
Epoch 299 | Loss: 0.2798
Time elapsed: 8.81 seconds
```

[77]:

```
epochs_vs_loss_without_cells(my_epochs, my_losses)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[77], line 1
----> 1 epochs_vs_loss_without_cells(my_epochs, my_losses)

NameError: name 'epochs_vs_loss_without_cells' is not defined
```

[57]:

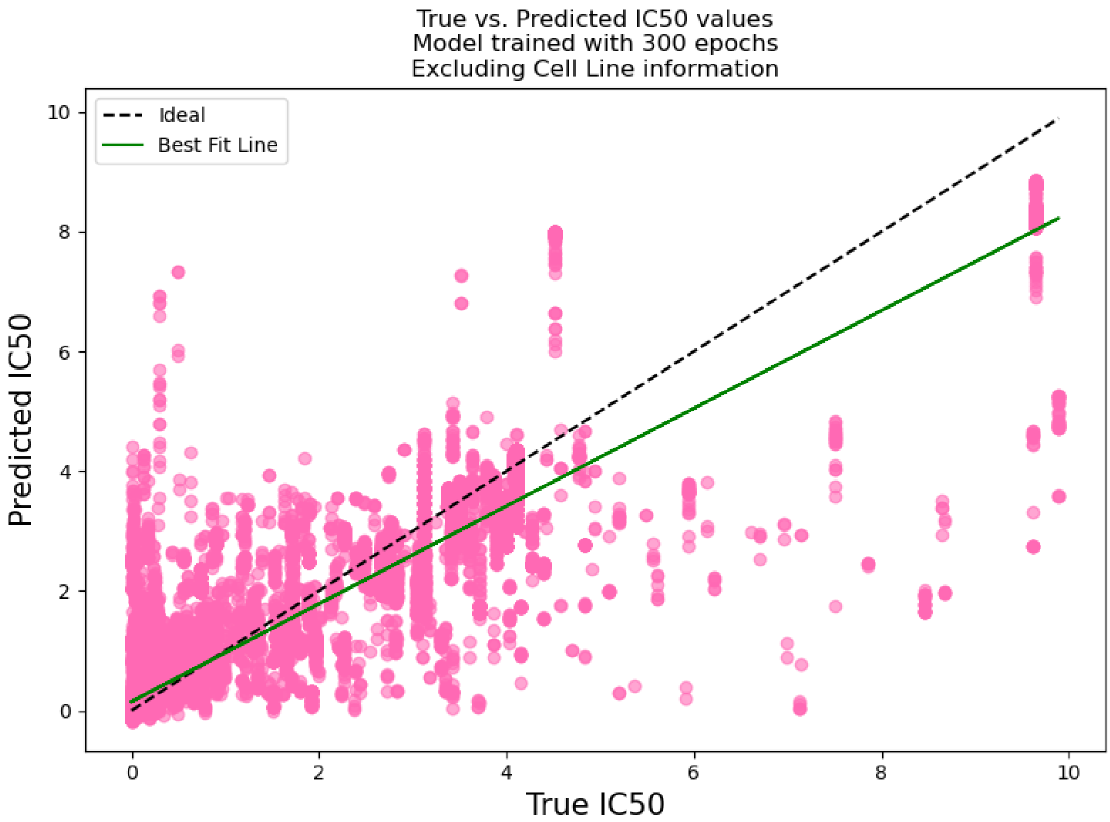
```
drug_features_test = X_test[['Molecular Mass', 'LogP', 'NumHDonors',␣
↪'NumHAcceptors', 'TPSA']]
patient_features_test = X_test.drop(columns=drug_features_test.columns)

metrics = evaluate_model(model, drug_features_test, patient_features_test,␣
↪y_test)

print(f"MSE: {metrics[0]:.4f}")
print(f"MAE: {metrics[1]:.4f}")
print(f"R²: {metrics[2]:.4f}")
```

```
MSE: 0.3875
MAE: 0.2711
R²: 0.8248
```

[132]:

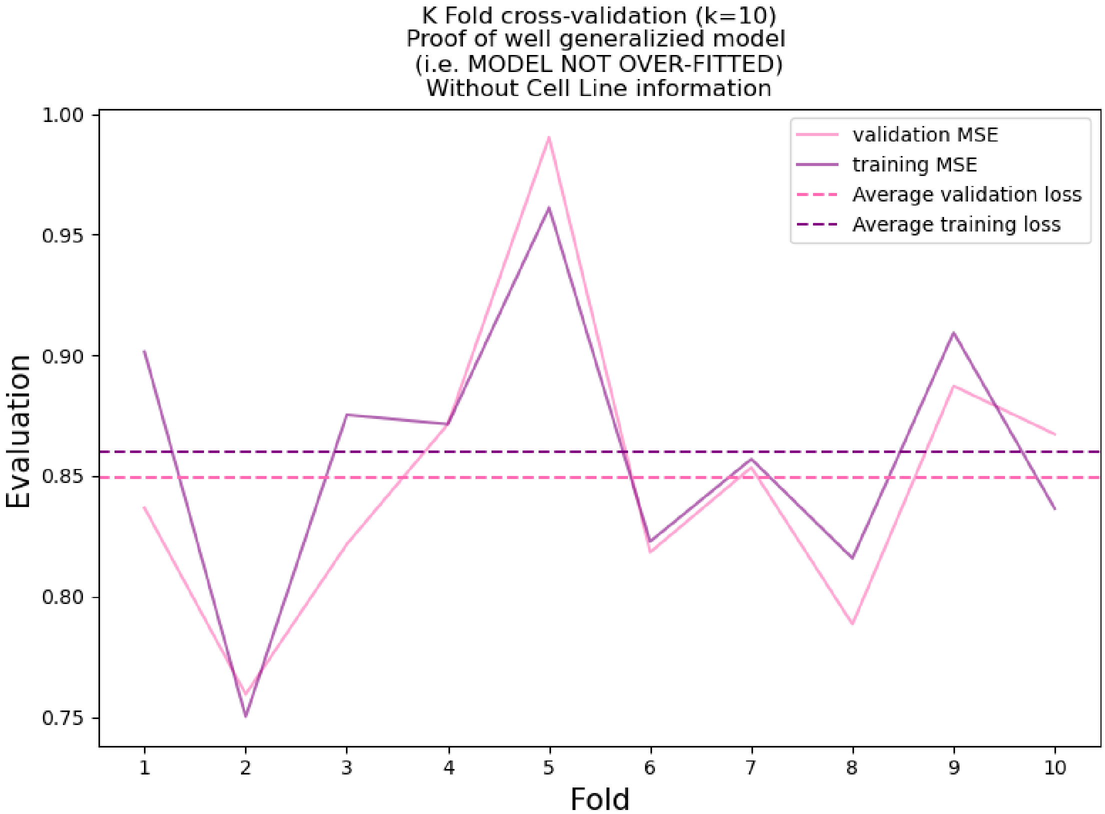
```
with torch.no_grad():
    preds = model(torch.tensor(drug_features_test.values).float(),
                  torch.tensor(patient_features_test.values).float()).cpu().
↪numpy()
plot_pred_vs_true_without_cells(y_test.to_numpy(), preds)
```

```
/var/folders/s8/ghqk1l4n7n9_w17t7hx21g2m0000gn/T/ipykernel_7821/778809264.py:5:
UserWarning: color is redundantly defined by the 'color' keyword argument and
the fmt string "r--" (-> color='r'). The keyword argument will take precedence.
  plt.plot([true.min(), true.max()], [true.min(), true.max()], 'r--',
label='Ideal', color = "Black")
```



[61]:
```
k_fold_cv = compute_CV_error(X_train,y_train)
```

[63]:
```
Kfold_CV_plot_without_cells(k_fold_cv)
```


K Fold cross-validation (k=10)
Proof of well generalizied model
(i.e. MODEL NOT OVER-FITTED)
Without Cell Line information
validation MSE
training MSE
Average validation loss
Average training loss
Evaluation
Fold
1.00
0.95
0.90
0.85
0.80
0.75
1
2
3
4
5
6
7
8
9
10

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: