In [7]:
```python
import pandas as pd
import numpy as np
```

In [8]:
```python
# Load the data
file = 'C:/Users/Lajpat Rai/Desktop/Ryseron Data Science Diploma/CIND820 Project/data.csv'
df = pd.read_csv(file)
df.head() # let' see first 5 rows
```

Out[8]:

	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
0	4.5878	1018.7	83.675	3.5758	23.979	1086.2	549.83	134.67	11.898	0.32663	81.952
1	4.2932	1018.3	84.235	3.5709	23.951	1086.1	550.05	134.67	11.892	0.44784	82.377
2	3.9045	1018.4	84.858	3.5828	23.990	1086.5	550.19	135.10	12.042	0.45144	83.776
3	3.7436	1018.3	85.434	3.5808	23.911	1086.5	550.17	135.03	11.990	0.23107	82.505
4	3.7516	1017.8	85.182	3.5781	23.917	1085.9	550.00	134.67	11.910	0.26747	82.028

1. Data Cleaning

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36733 entries, 0 to 36732
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   AT      36733 non-null  float64
 1   AP      36733 non-null  float64
 2   AH      36733 non-null  float64
 3   AFDP    36733 non-null  float64
 4   GTEP    36733 non-null  float64
 5   TIT     36733 non-null  float64
 6   TAT     36733 non-null  float64
 7   TEY     36733 non-null  float64
 8   CDP     36733 non-null  float64
 9   CO      36733 non-null  float64
 10  NOX     36733 non-null  float64
dtypes: float64(11)
memory usage: 3.1 MB
```

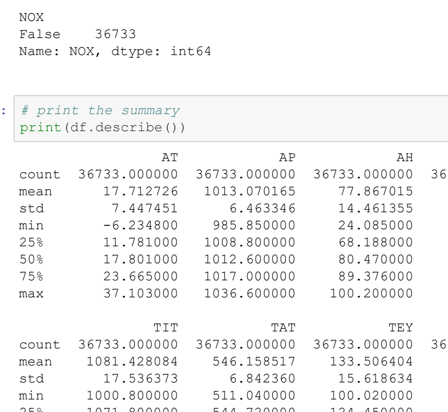
In [5]:
```python
# identify missing data
missing_data = df.isnull()
# Using a for loop in Python, the method ".value_counts()" counts the number of "True" values.
for column in missing_data.columns.values.tolist():
    print(column)
    print (missing_data[column].value_counts())
    print("")
```

```
AT
False    36733
Name: AT, dtype: int64

AP
False    36733
Name: AP, dtype: int64

AH
False    36733
Name: AH, dtype: int64

AFDP
False    36733
Name: AFDP, dtype: int64

GTEP
False    36733
Name: GTEP, dtype: int64

TIT
False    36733
Name: TIT, dtype: int64

TAT
False    36733
Name: TAT, dtype: int64

TEY
False    36733
Name: TEY, dtype: int64

CDP
False    36733
Name: CDP, dtype: int64

CO
False    36733
Name: CO, dtype: int64

NOX
False    36733
Name: NOX, dtype: int64
```

In [13]:
```python
# print the summary
print(df.describe())
```

```
                 AT            AP            AH          AFDP          GTEP  \
count  36733.000000  36733.000000  36733.000000  36733.000000  36733.000000   
mean      17.712726   1013.070165     77.867015      3.925518     25.563801   
std        7.447451      6.463346     14.461355      0.773936      4.195957   
min       -6.234800    985.850000     24.085000      2.087400     17.698000   
25%       11.781000   1008.800000     68.188000      3.355600     23.129000   
50%       17.801000   1012.600000     80.470000      3.937700     25.104000   
75%       23.665000   1017.000000     89.376000      4.376900     29.061000   
max       37.103000   1036.600000    100.200000      7.610600     40.716000   

                TIT           TAT           TEY           CDP            CO  \
count  36733.000000  36733.000000  36733.000000  36733.000000  36733.000000   
mean    1081.428084    546.158517    133.506404     12.060525      2.372468   
std       17.536373      6.842360     15.618634      1.088795      2.262672   
min     1000.800000    511.040000    100.020000      9.851800      0.000388   
25%     1071.800000    544.720000    124.450000     11.435000      1.182400   
50%     1085.900000    549.880000    133.730000     11.965000      1.713500   
75%     1097.000000    550.040000    144.080000     12.855000      2.842900   
max     1100.900000    550.610000    179.500000     15.159000     44.103000   

                NOX  
count  36733.000000  
mean      65.293067  
std       11.678357  
min       25.905000  
25%       57.162000  
50%       63.849000  
75%       71.548000  
max      119.910000  
```

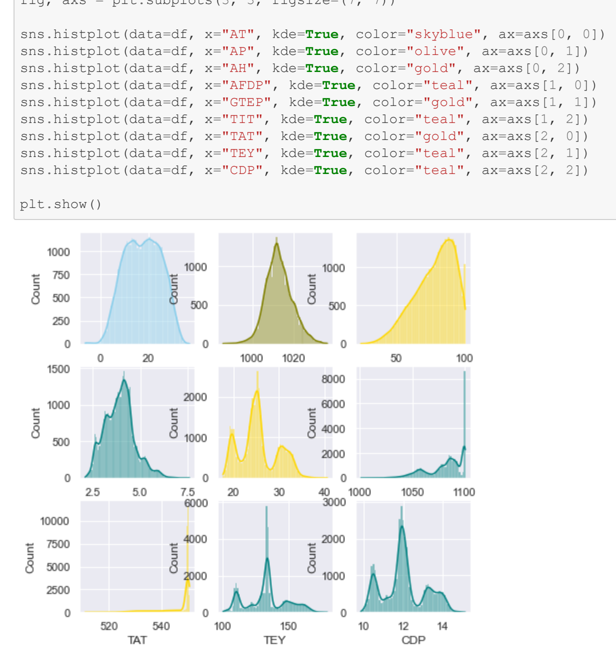
In [17]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
# set a grey background (use sns.set_theme() if seaborn version 0.11.0 or above)
sns.set(style="darkgrid")

fig, axs = plt.subplots(3, 3, figsize=(7, 7))

sns.histplot(data=df, x="AT", kde=True, color="skyblue", ax=axs[0, 0])
sns.histplot(data=df, x="AP", kde=True, color="olive", ax=axs[0, 1])
sns.histplot(data=df, x="AH", kde=True, color="gold", ax=axs[0, 2])
sns.histplot(data=df, x="AFDP", kde=True, color="teal", ax=axs[1, 0])
sns.histplot(data=df, x="GTEP", kde=True, color="gold", ax=axs[1, 1])
sns.histplot(data=df, x="TIT", kde=True, color="teal", ax=axs[1, 2])
sns.histplot(data=df, x="TAT", kde=True, color="gold", ax=axs[2, 0])
sns.histplot(data=df, x="TEY", kde=True, color="teal", ax=axs[2, 1])
sns.histplot(data=df, x="CDP", kde=True, color="teal", ax=axs[2, 2])

plt.show()
```



2. Exploratory Data Analysis

In [18]:
```python
# create histogram for each variabel to check normality.
import matplotlib.pyplot as plt
import matplotlib as plt
df.hist(bins=10, figsize=(10,10))
df.hist();
```

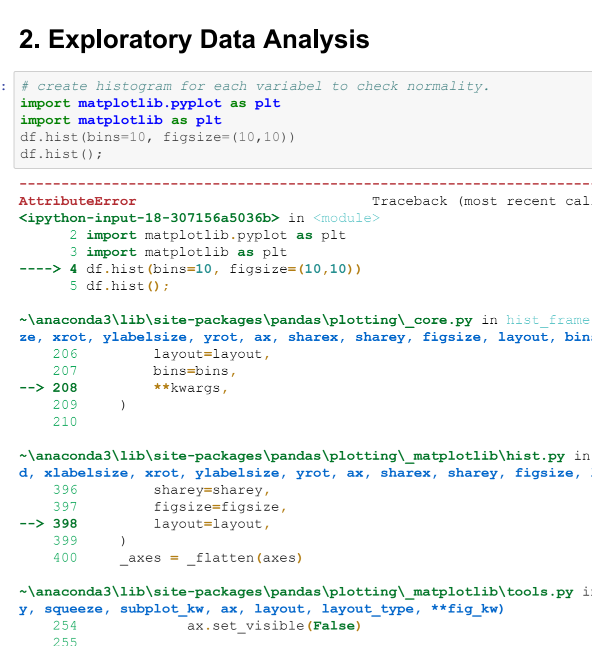
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-18-307156a5036b> in <module>
      2 import matplotlib.pyplot as plt
      3 import matplotlib as plt
----> 4 df.hist(bins=10, figsize=(10,10))
      5 df.hist();

~\anaconda3\lib\site-packages\pandas\plotting\_core.py in hist_frame(data, column, by, grid, xlabelsize, xrot, ylabelsize, yrot, ax, sharex, sharey, figsize, layout, bins, backend, **kwargs)
    206         layout=layout,
    207         bins=bins,
--> 208         **kwargs,
    209     )
    210 

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\hist.py in hist_frame(data, column, by, grid, xlabelsize, xrot, ylabelsize, yrot, ax, sharex, sharey, figsize, layout, bins, **kwds)
    396         sharey=sharey,
    397         figsize=figsize,
--> 398         layout=layout,
    399     )
    400     _axes = _flatten(axes)

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\tools.py in _subplots(naxes, sharex, sharey, squeeze, subplot_kw, ax, layout, layout_type, **fig_kw)
    254             ax.set_visible(False)
    255 
--> 256     _handle_shared_axes(axarr, nplots, naxes, nrows, ncols, sharex, sharey)
    257 
    258     if squeeze:

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\tools.py in _handle_shared_axes(axarr, nplots, naxes, nrows, ncols, sharex, sharey)
    296                 layout = np.zeros((nrows + 1, ncols + 1), dtype=np.bool)
    297                 for ax in axarr:
--> 298                     layout[ax.rowNum, ax.colNum] = ax.get_visible()
    299 
    300                 for ax in axarr:

AttributeError: 'AxesSubplot' object has no attribute 'rowNum'
```



Dataset is clean and has no missing values. Few parameters are normal except TIT, TEY, TAT, GTEP, CO & AH. Furthermore, Standard devitaion is small for all parameters hence most of data is close to mean. Data will be normalized before splitting it into training & testing data.

In [20]:
```python
# Check the outliers

df.boxplot(figsize = (10, 10), widths = 1)
```

Out[20]: <AxesSubplot:>

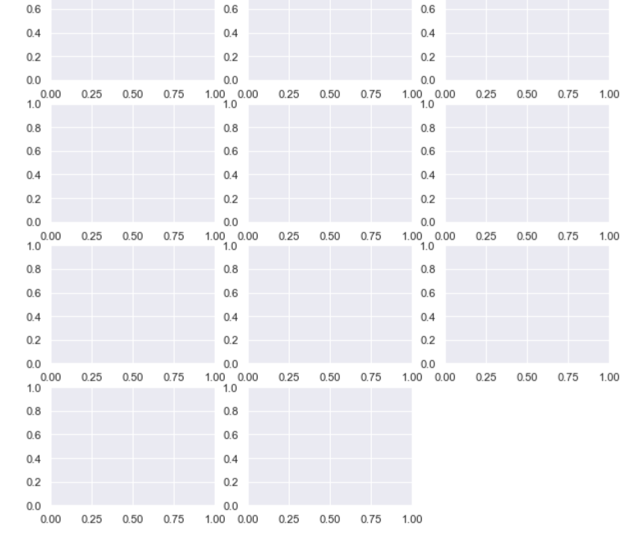


The boxplot shows that more input variables are outliers, so I'll use mean absolute error (MAE) to evaluate the model in modeling. The MAE is not sensitive to the outliers

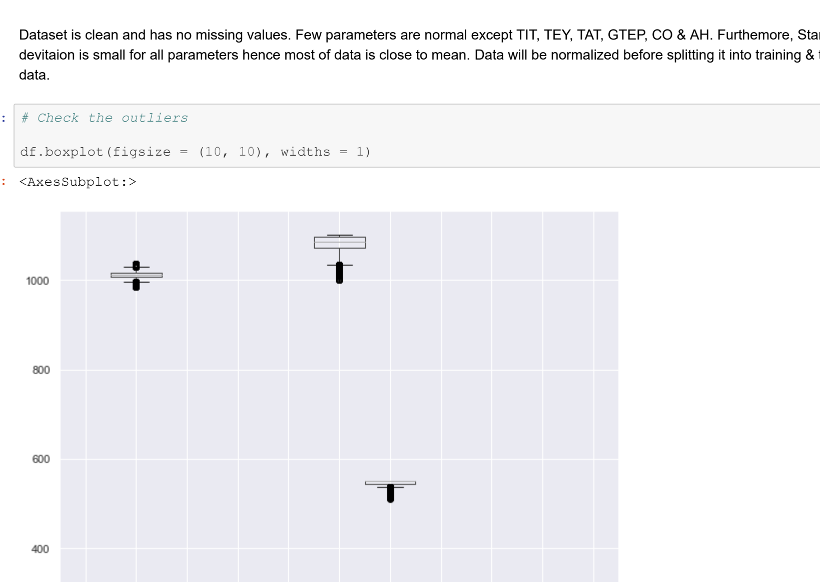
3. Feature selection Filter Based Tecnhiques

3.1 Feature Selection with the help of correlation

In [23]:
```python
# let's visualize the correlation thru heatmap
import matplotlib.pyplot as plt
import seaborn as sns
corr_matrix = df.corr()
plt.figure(figsize=(11,9))
dropSelf = np.zeros_like(corr_matrix)
dropSelf[np.triu_indices_from(dropSelf)] = True

sns.heatmap(corr_matrix, cmap=sns.diverging_palette(220, 10, as_cmap=True), annot=True, fmt=".2f", mask=dropSelf)

sns.set(font_scale=1.5)
```



The heatmap above shows the correlation between features and output variables(CO and NOx). It is easy to see that some features are negatively correlated each other. For example, the correlation between TIT and CO is -0.71, and it means that when the Turbine Inlet Temperature (TIT) decreases, the gas-turbine engine will produce more CO because a low TIT reduces the efficiency of the gas-turbine engine(look at the figure below).

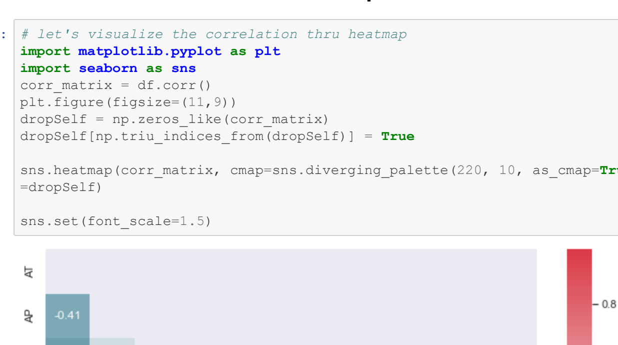


In [24]:
```python
X = df.drop(columns = ['CO', 'NOX'])
Y_CO = df[['CO']]
Y_NOX = df[['NOX']]
print(X)
print(Y_CO)
```

```
           AT      AP      AH    AFDP    GTEP     TIT     TAT     TEY     CDP
0      4.5878  1018.7  83.675  3.5758  23.979  1086.2  549.83  134.67  11.898
1      4.2932  1018.3  84.235  3.5709  23.951  1086.1  550.05  134.67  11.892
2      3.9045  1018.4  84.858  3.5828  23.990  1086.5  550.19  135.10  12.042
3      3.7436  1018.3  85.434  3.5808  23.911  1086.5  550.17  135.03  11.990
4      3.7516  1017.8  85.182  3.5781  23.917  1085.9  550.00  134.67  11.910
...       ...     ...     ...     ...     ...     ...     ...     ...     ...
36728  3.6268  1028.5  93.200  3.1661  19.087  1037.0  541.59  109.08  10.411
36729  4.1674  1028.6  94.036  3.1923  19.016  1037.6  542.28  108.79  10.344
36730  5.4820  1028.5  95.219  3.3128  18.857  1038.0  543.48  107.81  10.462
36731  5.8837  1028.7  94.200  3.9831  23.563  1076.9  550.11  131.41  11.771
36732  6.0392  1028.8  94.547  3.8752  22.524  1067.9  548.23  125.41  11.462

[36733 rows x 9 columns]
             CO
0       0.32663
1       0.44784
2       0.45144
3       0.23107
4       0.26747
...         ...
36728  10.99300
36729  11.14400
36730  11.41400
36731   3.31340
36732  11.98100

[36733 rows x 1 columns]
```

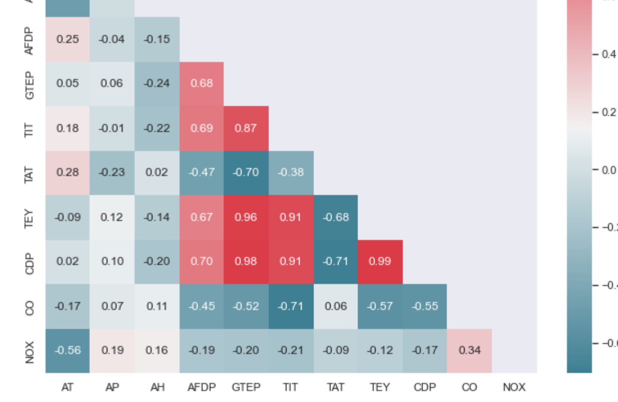
In [26]:
```python
#Find the information gain or mutual information of the independent variable with respect to a target variable
import warnings
warnings.simplefilter("ignore")
from sklearn.feature_selection import mutual_info_regression, mutual_info_classif
mi_CO = mutual_info_regression(X, Y_CO)
mi_CO = pd.Series(mi_CO)
mi_CO.index = X.columns
mi_CO.sort_values(ascending=False)
mi_CO.sort_values(ascending=False).plot.bar(figsize=(10, 4))
```

Out[26]: <AxesSubplot:>



In [27]:
```python
mi_NOX = mutual_info_regression(X, Y_NOX)
mi_NOX = pd.Series(mi_NOX)
mi_NOX.index = X.columns
mi_NOX.sort_values(ascending=False)
mi_NOX.sort_values(ascending=False).plot.bar(figsize=(10, 4))
```

Out[27]: <AxesSubplot:>



4 Wrapper technique for feature selection

In [28]:
```
!pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\lajpat rai\anaconda3\lib\site-packages (0.20.0)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\lajpat rai\anaconda3\lib\site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\lajpat rai\anaconda3\lib\site-packages (from mlxtend) (3.5.2)
Requirement already satisfied: numpy>=1.16.2 in c:\users\lajpat rai\anaconda3\lib\site-packages (from mlxtend) (1.18.1)
Requirement already satisfied: scipy>=1.2.1 in c:\users\lajpat rai\anaconda3\lib\site-packages (from mlxtend) (1.4.1)
Requirement already satisfied: joblib>=0.13.2 in c:\users\lajpat rai\anaconda3\lib\site-packages (from mlxtend) (1.0.1)
Requirement already satisfied: setuptools in c:\users\lajpat rai\anaconda3\lib\site-packages (from mlxtend) (45.2.0.post20200210)
Requirement already satisfied: pandas>=0.24.2 in c:\users\lajpat rai\anaconda3\lib\site-packages (from mlxtend) (1.0.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\lajpat rai\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend) (3.1.0)
Requirement already satisfied: packaging>=20.0 in c:\users\lajpat rai\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (20.1)
Requirement already satisfied: cycler>=0.10 in c:\users\lajpat rai\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\lajpat rai\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (7.0.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\lajpat rai\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\lajpat rai\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.4.6)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\lajpat rai\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (4.28.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\lajpat rai\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.1.0)
Requirement already satisfied: pytz>=2017.2 in c:\users\lajpat rai\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2019.3)
Requirement already satisfied: six in c:\users\lajpat rai\anaconda3\lib\site-packages (from packaging>=20.0->matplotlib>=3.0.0->mlxtend) (1.14.0)
```

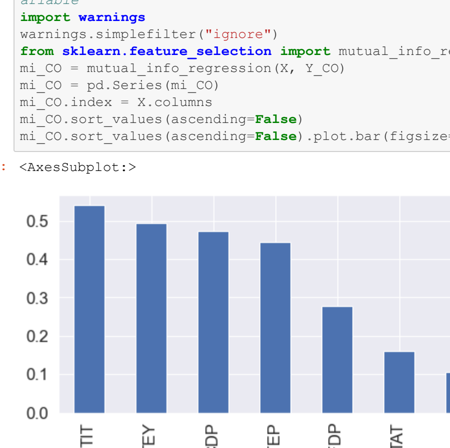
4.1 Forward Selection — SFS() from mlxtend

In [29]:
```python
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
from sklearn.linear_model import LinearRegression

#Define Sequential Forward Selection (sfs)
sfs = SFS(LinearRegression(),
          k_features=5,
          forward=True,
          floating=False,
          scoring = 'r2',
          cv = 0)
#Use SFS to select the top 5 features
sfs.fit(X, Y_CO)

#Create a dataframe for the SFS results
df_SFS_results = pd.DataFrame(sfs.subsets_).transpose()
df_SFS_results
```

Out[29]:

	feature_idx	cv_scores	avg_score	feature_names
1	(5,)	[0.49882444817014526]	0.498824	(TIT,)
2	(5, 6)	[0.5507217644680591]	0.550722	(TIT, TAT)
3	(2, 5, 6)	[0.5552174969842718]	0.555217	(AH, TIT, TAT)
4	(2, 5, 6, 7)	[0.5564461888633703]	0.556446	(AH, TIT, TAT, TEY)
5	(2, 5, 6, 7, 8)	[0.5581241617573389]	0.558124	(AH, TIT, TAT, TEY, CDP)

In [30]:
```python
from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs
sfs = SFS(LinearRegression(),
         k_features=9,
         forward=True,
         floating=False,
         scoring='r2',
         cv=5)

sfs = sfs.fit(X, Y_CO)
fig = plot_sfs(sfs.get_metric_dict(), kind='std_err')

plt.title('Sequential Forward Selection (w. StdErr)')
plt.grid()
plt.show()
```

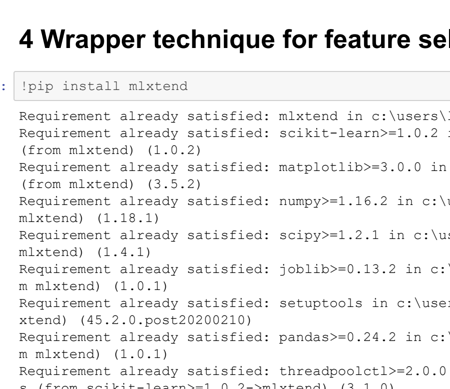


In [31]:
```python
sfs.fit(X, Y_NOX)

#Create a dataframe for the SFS results
df_SFS_results = pd.DataFrame(sfs.subsets_).transpose()
df_SFS_results
```

Out[31]:

	feature_idx	cv_scores	avg_score	feature_names
1	(0,)	[0.3772186881848645, 0.02335901211707392, 0.09...	0.101282	(AT,)
2	(0, 4)	[0.4375572055870701, 0.08253102232092047, 0.11...	0.141457	(AT, GTEP)
3	(0, 4, 5)	[0.46127520723803206, 0.11786682141083127, 0.1...	0.165202	(AT, GTEP, TIT)
4	(0, 4, 5, 6)	[0.4574368886284075, 0.17872145749274448, 0.20...	0.201565	(AT, GTEP, TIT, TAT)
5	(0, 4, 5, 6, 7)	[0.4706119731078341, 0.30627659376343375, 0.30...	0.3373	(AT, GTEP, TIT, TAT, TEY)
6	(0, 1, 4, 5, 6, 7)	[0.45982441375422856, 0.3154808646727236, 0.31...	0.339749	(AT, AP, GTEP, TIT, TAT, TEY)
7	(0, 1, 4, 5, 6, 7, 8)	[0.4585681576319547, 0.3098772733868126, 0.311...	0.334031	(AT, AP, GTEP, TIT, TAT, TEY, CDP)
8	(0, 1, 2, 4, 5, 6, 7, 8)	[0.5247090226323172, 0.42793025624651726, 0.34...	0.330764	(AT, AP, AH, GTEP, TIT, TAT, TEY, CDP)
9	(0, 1, 2, 3, 4, 5, 6, 7, 8)	[0.5278778454801685, 0.4291409781303308, 0.218...	0.305416	(AT, AP, AH, AFDP, GTEP, TIT, TAT, TEY, CDP)

In [32]:
```python
sfs = SFS(LinearRegression(),
         k_features=9,
         forward=True,
         floating=False,
         scoring='r2',
         cv=5)

sfs = sfs.fit(X, Y_NOX)
fig = plot_sfs(sfs.get_metric_dict(), kind='std_err')

plt.title('Sequential Forward Selection (w. StdErr)')
plt.grid()
plt.show()
```



4.2 Backward elimination — RFE() from Sklearn

In [33]:
```python
from sklearn.feature_selection import RFE
#Build a logistic regression model
#model = LinearRegression()
#Define RFE
rfe = RFE(estimator=LinearRegression(), n_features_to_select=5)
#Use RFE to select the top 5 features
fit = rfe.fit(X, Y_CO)

features = X.columns.to_list()
df_RFE_results = []
for i in range(X.shape[1]):
    df_RFE_results.append(
        {
            'Feature_names': features[i],
            'Selected':  rfe.support_[i],
            'RFE_ranking':  rfe.ranking_[i],
        }
    )

df_RFE_results = pd.DataFrame(df_RFE_results)
df_RFE_results.index.name='Columns'
df_RFE_results
```

Out[33]:

Columns	Feature_names	Selected	RFE_ranking
0	AT	False	3
1	AP	False	5
2	AH	False	4
3	AFDP	True	1
4	GTEP	False	2
5	TIT	True	1
6	TAT	True	1
7	TEY	True	1
8	CDP	True	1

In [34]:
```python
#Use RFE to select the top 5 features
fit = rfe.fit(X, Y_NOX)

features = X.columns.to_list()
df_RFE_results = []
for i in range(X.shape[1]):
    df_RFE_results.append(
        {
            'Feature_names': features[i],
            'Selected':  rfe.support_[i],
            'RFE_ranking':  rfe.ranking_[i],
        }
    )

df_RFE_results = pd.DataFrame(df_RFE_results)
df_RFE_results.index.name='Columns'
df_RFE_results
```

Out[34]:

Columns	Feature_names	Selected	RFE_ranking
0	AT	True	1
1	AP	False	2
2	AH	False	4
3	AFDP	False	3
4	GTEP	False	5
5	TIT	True	1
6	TAT	True	1
7	TEY	True	1
8	CDP	True	1

backward elimintaion using SFS

In [35]:
```python
#Define Sequential Forward Selection (sfs)
sfs = SFS(LinearRegression(),
          k_features=5,
          forward=False,
          floating=False,
          scoring = 'r2',
          cv = 0)
#Use SFS to select the top 5 features
sfs.fit(X, Y_CO)

#Create a dataframe for the SFS results
df_SFS_results = pd.DataFrame(sfs.subsets_).transpose()
df_SFS_results
```

Out[35]:

	feature_idx	cv_scores	avg_score	feature_names
9	(0, 1, 2, 3, 4, 5, 6, 7, 8)	[0.5634322493233096]	0.563432	(AT, AP, AH, AFDP, GTEP, TIT, TAT, TEY, CDP)
8	(0, 2, 3, 4, 5, 6, 7, 8)	[0.5633593345177799]	0.563359	(AT, AH, AFDP, GTEP, TIT, TAT, TEY, CDP)
7	(0, 2, 4, 5, 6, 7, 8)	[0.5624599981695952]	0.56246	(AT, AH, GTEP, TIT, TAT, TEY, CDP)
6	(0, 2, 5, 6, 7, 8)	[0.5612057205079215]	0.561206	(AT, AH, TIT, TAT, TEY, CDP)
5	(0, 5, 6, 7, 8)	[0.5587001023065896]	0.5587	(AT, TIT, TAT, TEY, CDP)

In [36]:
```python
sfs = SFS(LinearRegression(),
          k_features=5,
          forward=False,
          floating=False,
          scoring = 'r2',
          cv = 0)
#Use SFS to select the top 5 features
sfs.fit(X, Y_NOX)

#Create a dataframe for the SFS results
df_SFS_results = pd.DataFrame(sfs.subsets_).transpose()
df_SFS_results
```

Out[36]:

	feature_idx	cv_scores	avg_score	feature_names
9	(0, 1, 2, 3, 4, 5, 6, 7, 8)	[0.5178226679212272]	0.517823	(AT, AP, AH, AFDP, GTEP, TIT, TAT, TEY, CDP)
8	(0, 1, 2, 3, 5, 6, 7, 8)	[0.5177685010169375]	0.517769	(AT, AP, AH, AFDP, TIT, TAT, TEY, CDP)
7	(0, 1, 2, 3, 5, 6, 7)	[0.5176071570924646]	0.517607	(AT, AP, AH, AFDP, TIT, TAT, TEY)
6	(0, 1, 2, 5, 6, 7)	[0.5167645854414764]	0.516765	(AT, AP, AH, TIT, TAT, TEY)
5	(0, 2, 5, 6, 7)	[0.5031467462217525]	0.503147	(AT, AH, TIT, TAT, TEY)

4.3 Step-wise Selection — SFFS() from mlxtend

In [37]:
```python
#Define Sequential Forward Selection (sfs)
sfs = SFS(LinearRegression(),
          k_features=5,
          forward=True,
          floating=True,
          scoring = 'r2',
          cv = 0)
#Use SFS to select the top 5 features
sfs.fit(X, Y_CO)

#Create a dataframe for the SFS results
df_SFS_results = pd.DataFrame(sfs.subsets_).transpose()
df_SFS_results
```

Out[37]:

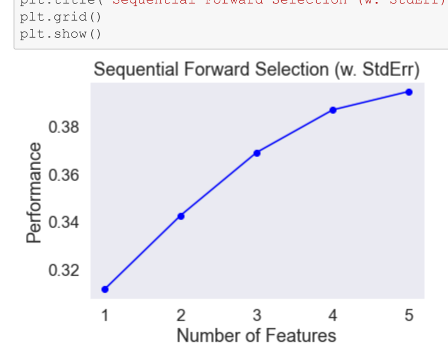
	feature_idx	cv_scores	avg_score	feature_names
1	(5,)	[0.49882444817014526]	0.498824	(TIT,)
2	(5, 6)	[0.5507217644680591]	0.550722	(TIT, TAT)
3	(2, 5, 6)	[0.5552174969842718]	0.555217	(AH, TIT, TAT)
4	(2, 5, 6, 7)	[0.5564461888633703]	0.556446	(AH, TIT, TAT, TEY)
5	(2, 5, 6, 7, 8)	[0.5581241617573389]	0.558124	(AH, TIT, TAT, TEY, CDP)

In [38]:
```python
#Define Sequential Forward Selection (sfs)
sfs = SFS(LinearRegression(),
          k_features=5,
          forward=True,
          floating=True,
          scoring = 'r2',
          cv = 0)
#Use SFS to select the top 5 features
sfs.fit(X, Y_NOX)

#Create a dataframe for the SFS results
df_SFS_results = pd.DataFrame(sfs.subsets_).transpose()
df_SFS_results
```

Out[38]:

	feature_idx	cv_scores	avg_score	feature_names
1	(0,)	[0.31155773122511365]	0.311558	(AT,)
2	(0, 4)	[0.34261205608496104]	0.342612	(AT, GTEP)
3	(0, 2, 4)	[0.369117213791586]	0.369117	(AT, AH, GTEP)
4	(0, 2, 3, 4)	[0.3870041880302296]	0.387004	(AT, AH, AFDP, GTEP)
5	(0, 2, 3, 4, 5)	[0.3947387026237553]	0.394739	(AT, AH, AFDP, GTEP, TIT)

In [39]:
```python
fig1 = plot_sfs(sfs.get_metric_dict(), kind='std_dev')
plt.title('Sequential Forward Selection (w. StdErr)')
plt.grid()
plt.show()
```



5. Embedded Feature Selection Methods

5.1 Random Forest

In [40]:
```python
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn import tree
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import SelectFromModel
```

In [41]:
```python
# Normalize feature vector
X1 = StandardScaler().fit_transform(X)
# Split the dataset
X_train, X_test, y_train_CO, y_test_CO = train_test_split(X1, Y_CO, test_size = 0.30)

# selecting the features
sel = SelectFromModel(RandomForestRegressor(n_estimators = 100,max_features=5))
sel.fit(X_train, y_train_CO)
```

Out[41]:
```
SelectFromModel(estimator=RandomForestRegressor(max_features=5))
```

In [42]:
```python
selected_feat= X.columns[(sel.get_support())]
sel.get_support().sum()

print(selected_feat)
```

```
Index(['TIT', 'TAT', 'TEY'], dtype='object')
```

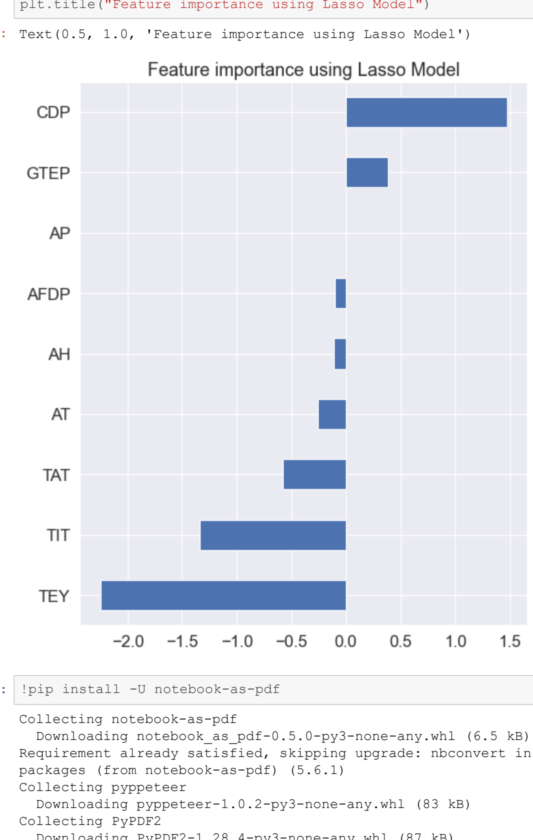
5.2 Lasso Regression

In [43]:
```python
from sklearn.linear_model import LassoCV
lcv = LassoCV()
# Split the dataset
X_train, X_test, y_train_CO, y_test_CO = train_test_split(X1, Y_CO, test_size = 0.30, random_state = 0)

lcv.fit(X_train, y_train_CO)

print("Best alpha using built-in LassoCV: %f" % lcv.alpha_)
print("Best score using built-in LassoCV: %f" %lcv.score(X_train,y_train_CO))
coef = pd.Series(lcv.coef_, index = X.columns)

print("Lasso picked " + str(sum(coef != 0)) + " variables and eliminated the other " +  str(sum(coef ==
 0)) + " variables")
```

```
Best alpha using built-in LassoCV: 0.001593
Best score using built-in LassoCV: 0.559507
Lasso picked 9 variables and eliminated the other 0 variables
```

In [44]:
```python
imp_coef = coef.sort_values()
import matplotlib
matplotlib.rcParams['figure.figsize'] = (8.0, 10.0)
imp_coef.plot(kind = "barh")
plt.title("Feature importance using Lasso Model")
```

Out[44]:
```
Text(0.5, 1.0, 'Feature importance using Lasso Model')
```



In [46]:
```
!pip install -U notebook-as-pdf
```

```
Collecting notebook-as-pdf
  Downloading notebook_as_pdf-0.5.0-py3-none-any.whl (6.5 kB)
Requirement already satisfied, skipping upgrade: nbconvert in c:\users\lajpat rai\anaconda3\lib\site-
packages (from notebook-as-pdf) (5.6.1)
Collecting pyppeteer
  Downloading pyppeteer-1.0.2-py3-none-any.whl (83 kB)
Collecting PyPDF2
  Downloading PyPDF2-1.28.4-py3-none-any.whl (87 kB)
Requirement already satisfied, skipping upgrade: mistune<2,>=0.8.1 in c:\users\lajpat rai\anaconda3\l
ib\site-packages (from nbconvert->notebook-as-pdf) (0.8.4)
Requirement already satisfied, skipping upgrade: pandocfilters>=1.4.1 in c:\users\lajpat rai\anaconda
3\lib\site-packages (from nbconvert->notebook-as-pdf) (1.4.2)
Requirement already satisfied, skipping upgrade: testpath in c:\users\lajpat rai\anaconda3\lib\site-p
ackages (from nbconvert->notebook-as-pdf) (0.4.4)
Requirement already satisfied, skipping upgrade: jinja2>=2.4 in c:\users\lajpat rai\anaconda3\lib\sit
e-packages (from nbconvert->notebook-as-pdf) (2.11.1)
Requirement already satisfied, skipping upgrade: jupyter-core in c:\users\lajpat rai\anaconda3\lib\si
te-packages (from nbconvert->notebook-as-pdf) (4.6.1)
Requirement already satisfied, skipping upgrade: defusedxml in c:\users\lajpat rai\anaconda3\lib\site
-packages (from nbconvert->notebook-as-pdf) (0.6.0)
Requirement already satisfied, skipping upgrade: traitlets>=4.2 in c:\users\lajpat rai\anaconda3\lib
\site-packages (from nbconvert->notebook-as-pdf) (4.3.3)
Requirement already satisfied, skipping upgrade: pygments in c:\users\lajpat rai\anaconda3\lib\site-p
ackages (from nbconvert->notebook-as-pdf) (2.5.2)
Requirement already satisfied, skipping upgrade: nbformat>=4.4 in c:\users\lajpat rai\anaconda3\lib\s
ite-packages (from nbconvert->notebook-as-pdf) (5.0.4)
Requirement already satisfied, skipping upgrade: entrypoints>=0.2.2 in c:\users\lajpat rai\anaconda3
\lib\site-packages (from nbconvert->notebook-as-pdf) (0.3)
Requirement already satisfied, skipping upgrade: bleach in c:\users\lajpat rai\anaconda3\lib\site-pac
kages (from nbconvert->notebook-as-pdf) (3.1.0)
Collecting appdirs<2.0.0,>=1.4.3
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Collecting pyee<9.0.0,>=8.1.0
  Downloading pyee-8.2.2-py2.py3-none-any.whl (12 kB)
Collecting websockets<11.0,>=10.0
  Downloading websockets-10.3-cp37-cp37m-win_amd64.whl (98 kB)
Requirement already satisfied, skipping upgrade: tqdm<5.0.0,>=4.42.1 in c:\users\lajpat rai\anaconda3
\lib\site-packages (from pyppeteer->notebook-as-pdf) (4.42.1)
Requirement already satisfied, skipping upgrade: urllib3<2.0.0,>=1.25.8 in c:\users\lajpat rai\anacon
da3\lib\site-packages (from pyppeteer->notebook-as-pdf) (1.25.8)
Requirement already satisfied, skipping upgrade: importlib-metadata>=1.4 in c:\users\lajpat rai\anaco
nda3\lib\site-packages (from pyppeteer->notebook-as-pdf) (1.5.0)
Collecting certifi>=2021
  Downloading certifi-2022.5.18.1-py3-none-any.whl (155 kB)
Requirement already satisfied, skipping upgrade: MarkupSafe>=0.23 in c:\users\lajpat rai\anaconda3\li
b\site-packages (from jinja2>=2.4->nbconvert->notebook-as-pdf) (1.1.1)
Requirement already satisfied, skipping upgrade: pywin32>=1.0; sys_platform == "win32" in c:\users\la
jpat rai\anaconda3\lib\site-packages (from jupyter-core->nbconvert->notebook-as-pdf) (227)
Requirement already satisfied, skipping upgrade: ipython-genutils in c:\users\lajpat rai\anaconda3\li
b\site-packages (from traitlets>=4.2->nbconvert->notebook-as-pdf) (0.2.0)
Requirement already satisfied, skipping upgrade: decorator in c:\users\lajpat rai\anaconda3\lib\site-
packages (from traitlets>=4.2->nbconvert->notebook-as-pdf) (4.4.1)
Requirement already satisfied, skipping upgrade: six in c:\users\lajpat rai\anaconda3\lib\site-packag
es (from traitlets>=4.2->nbconvert->notebook-as-pdf) (1.14.0)
Requirement already satisfied, skipping upgrade: jsonschema!=2.5.0,>=2.4 in c:\users\lajpat rai\anaco
nda3\lib\site-packages (from nbformat>=4.4->nbconvert->notebook-as-pdf) (3.2.0)
Requirement already satisfied, skipping upgrade: webencodings in c:\users\lajpat rai\anaconda3\lib\si
te-packages (from bleach->nbconvert->notebook-as-pdf) (0.5.1)
Requirement already satisfied, skipping upgrade: zipp>=0.5 in c:\users\lajpat rai\anaconda3\lib\site-
packages (from importlib-metadata>=1.4->pyppeteer->notebook-as-pdf) (2.2.0)
Requirement already satisfied, skipping upgrade: attrs>=17.4.0 in c:\users\lajpat rai\anaconda3\lib\s
ite-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (19.3.0)
Requirement already satisfied, skipping upgrade: setuptools in c:\users\lajpat rai\anaconda3\lib\site
-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (45.2.0.post20200
210)
Requirement already satisfied, skipping upgrade: pyrsistent>=0.14.0 in c:\users\lajpat rai\anaconda3
\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (0.15.7)
Installing collected packages: appdirs, pyee, websockets, certifi, pyppeteer, PyPDF2, notebook-as-pdf
  Attempting uninstall: certifi
    Found existing installation: certifi 2019.11.28
    Uninstalling certifi-2019.11.28:
      Successfully uninstalled certifi-2019.11.28
Successfully installed PyPDF2-1.28.4 appdirs-1.4.4 certifi-2022.5.18.1 notebook-as-pdf-0.5.0 pyee-8.
2.2 pyppeteer-1.0.2 websockets-10.3
```