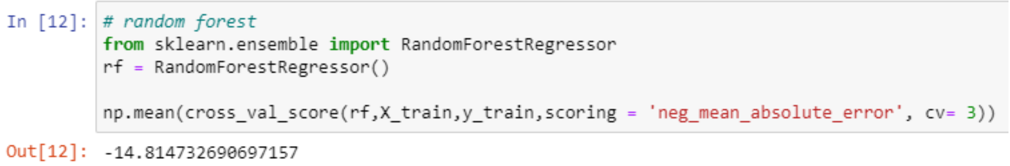
Model performance:

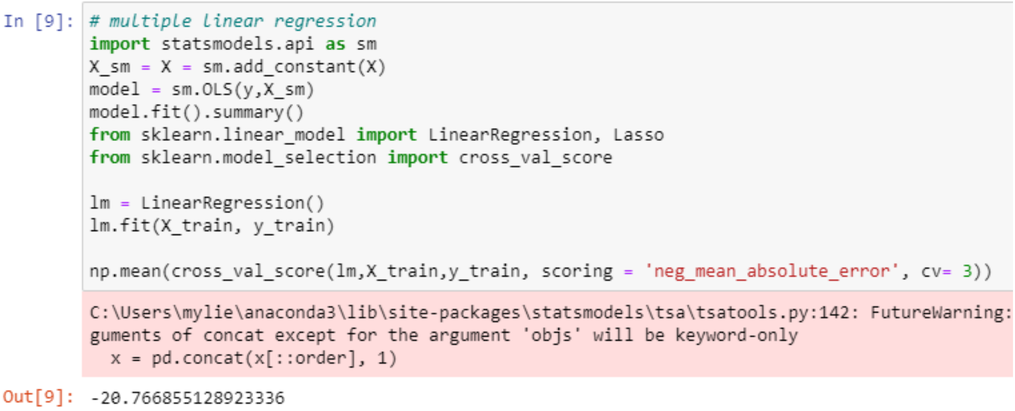
The Random Forest model far outperformed the other approaches on the test and validation sets.

- Random Forest: MAE = 14.22



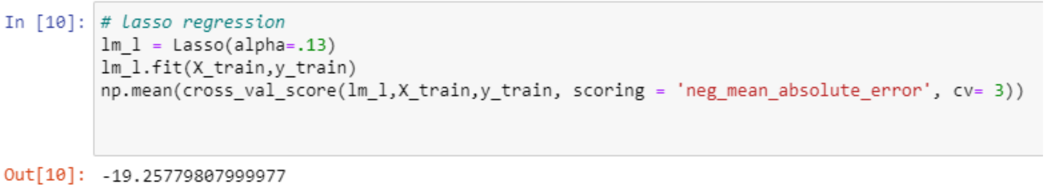
```
In [12]: # random forest
         from sklearn.ensemble import RandomForestRegressor
         rf = RandomForestRegressor()

         np.mean(cross_val_score(rf,X_train,y_train,scoring = 'neg_mean_absolute_error', cv= 3))

Out[12]: -14.814732690697157
```

- Linear Regression: MAE = 20.86



```
In [9]: # multiple linear regression
        import statsmodels.api as sm
        X_sm = X = sm.add_constant(X)
        model = sm.OLS(y,X_sm)
        model.fit().summary()
        from sklearn.linear_model import LinearRegression, Lasso
        from sklearn.model_selection import cross_val_score

        lm = LinearRegression()
        lm.fit(X_train, y_train)

        np.mean(cross_val_score(lm,X_train,y_train, scoring = 'neg_mean_absolute_error', cv= 3))

        C:\Users\mylie\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWarning:
        guments of concat except for the argument 'objs' will be keyword-only
          x = pd.concat(x[::order], 1)

Out[9]: -20.766855128923336
```

- Ridge Regression: MAE = 19.25



```
In [10]: # lasso regression
         lm_l = Lasso(alpha=.13)
         lm_l.fit(X_train,y_train)
         np.mean(cross_val_score(lm_l,X_train,y_train, scoring = 'neg_mean_absolute_error', cv= 3))

Out[10]: -19.25779807999977
```