Predicting EPS

Team: Finding Nemo

Harsh Mehta, Karthick Ramasubramanian, Rohitashwa Chakraborty

Executive Summary

Using Company fundamental data (Balance Sheet, Income Statement) and market variables, we are trying to predict the Net Income in the following year. This would be later converted EPS using current shares outstanding.

We are using regression models like PooledOLS, Panel Regression to make a prediction.

Data

- Pulled company level observations from WRDS Compustat, Fundamentals, Annual table.
- Pulled market data level data from CRSP and Compustat

Objective

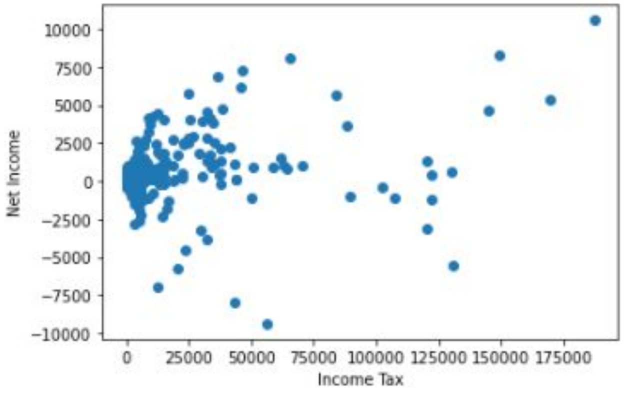
- Predicting EPS: Modelling done with output variable as Net Income.
- Identify the primary factors contributing to Earnings or Net Income.

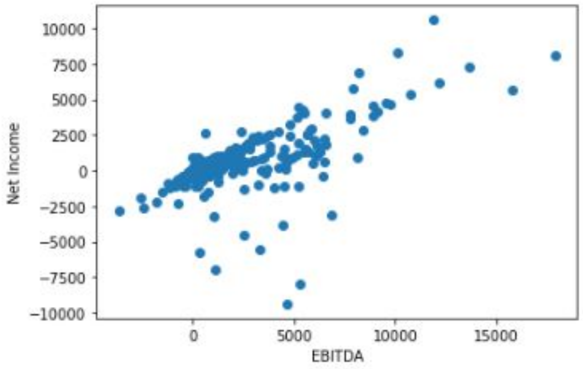
Variables

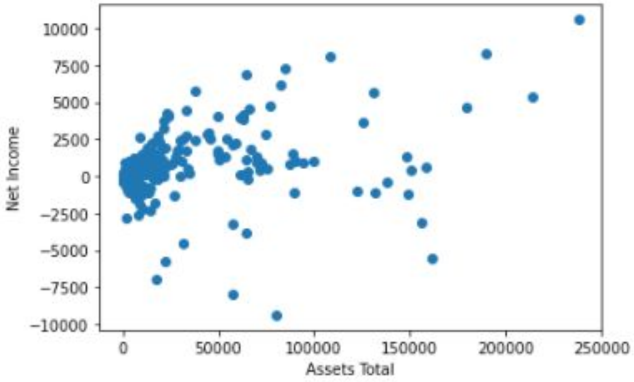
The variables are clubbed in 2 categories

- Company specific variables: From Balance Sheet and Income Statement. Here data from past years is used including - Sales, EBIT, Interest, Previous years NI, Industry code etc..
- Global/Market Variables: This includes variables that affect market and sector as whole. Variables included are - S&P and Sector Index, Treasury rates (long and short term), Oil prices and Inflation.

Plots







feature	VIF
at	7.762877e+08
lt	4.051255e+08
icapt	1.212110e+08
dltt	7.599849e+06
pi	6.533345e+02
seq	5.310615e+02
prev_ni	4.188967e+02
ebitda	1.324926e+02
act	1.239270e+02
revt	1.033458e+02
txt	9.591769e+01
ebit	8.599682e+01
gp	3.722652e+01
ch	1.928548e+01
ap	1.882822e+01

Model Preprocessing

```
data.drop(['consol','popsrc','indfmt'],axis=1,inplace=True) #same for all rows
data.drop(['dvpd','opiti','tii','uopi'], axis=1, inplace=True) #NaN values only
data.drop(['gld','gleps','glp'], axis=1, inplace=True) #more than 90% values are NaN

data.sort_values(by=['gvkey','fyear','fyr'],inplace=True) #sort by gvkey and fyear

#drop columns with constant value
for i in data.columns:
    if(len(data[i].unique()) == 1):
        data.drop(i, axis=1,inplace=True)

#drop columns with more than 10% missing values?
perc = 10.0
min_count =  int(((100-perc)/100)*data.shape[0] + 1)
data = data.dropna( axis=1, thresh=min_count)
data
```

```
#removing the look ahead bias by shifting the data
ni = df.groupby('gvkey')['ni'].shift(-1)
df['ni'] = ni
df.dropna(inplace=True)
```

Modelling

- As expected, we observe a high degree of multicollinearity. So, using Linear Regression would give incorrect output in terms of feature importance.
- To avoid this to some degree, we have used Stepwise Selection and PCA at some loss of interpretability.

Model Preprocessing

```
df = pd.DataFrame()
df['gvkey'] = data['gvkey']
df[data.columns[6:]] = data[data.columns[6:]]

#fill missing values using forward fill and backward fill and take average
#what this means is that the asset value in that year lied between the the asset value the year before and the one
#the year after
temp = df.groupby('gvkey').fillna(method='ffill')
temp = temp.fillna(0)
temp2 = df.groupby('gvkey').fillna(method='bfill')
temp2 = temp2.fillna(0)

cols = data.columns[6:]
for i in cols:
    df[i] = (temp[i] + temp2[i])//2
```

Model Preprocessing

```
def stepwise_selection(df, X, y, reg_model, initial_list=[], threshold_in=0.01, threshold_out = 0.05, verbose=True):
    included = list(initial_list)
    while True:
        changed=False
        # forward step
        excluded = list(set(X.columns)-set(included))
        new_pval = pd.Series(index=excluded)
        for new_column in excluded:
            exog = sm.add_constant(pd.DataFrame(X[included+[new_column]]))
            model = reg_model(df.ni, exog).fit()
            new_pval[new_column] = model.pvalues[new_column]
        best_pval = new_pval.min()
        if best_pval < threshold_in:
            best_feature = new_pval.idxmin()
            included.append(best_feature)
            changed=True
            if verbose:
                print('Add  {:30} with p-value {:.6}'.format(best_feature, best_pval))

        # backward step
        model = reg_model(df.ni, sm.add_constant(pd.DataFrame(X[included]))).fit()
        # use all coefs except intercept
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max() # null if pvalues is empty
        if worst_pval > threshold_out:
            changed=True
            worst_feature = pvalues.idxmax()
            included.remove(worst_feature)
            if verbose:
                print('Drop {:30} with p-value {:.6}'.format(worst_feature, worst_pval))
        if not changed:
            break
    return included
```

Model Building

```
def prediction(df1, components):
    df1 = df1.drop('gvkey',axis=1)
    alpha_vals = np.arange(0.01,1)
    X_train, X_test, y_train, y_test = train_test_split(df1.loc[:, df1.columns != 'ni'], df1['ni'],
                                                        test_size=0.33, random_state=42)
    pipe = Pipeline(steps=[('scaler', StandardScaler()), ('pca', PCA(n_components= components)),
                           ('ridge', Ridge(fit_intercept=True))])
    gsc = GridSearchCV(pipe, param_grid={ 'ridge__alpha': alpha_vals},cv=10, scoring='r2')
    gsc.fit(X_train, y_train)
    y_pred = gsc.predict(X_test)
    RMSE = (mean_squared_error(y_test,y_pred)**(1/2))
    print(r2_score(y_test, y_pred))
    print(RMSE)
```

```
#analysis of PCA
def PCA_analysis(temp, components):
    pca = PCA()
    dataset = pd.DataFrame()

    #checking correlation of target variable with different principal components.
    transformed = pca.fit_transform(temp.drop('ni',axis=1))
    for i in range(0, len(transformed[0])):
        dataset[i] = transformed[:,i]
        print(i, temp['ni'].corr(dataset[i]))

    pca = PCA(n_components=components)
    transformed = pca.fit_transform(temp.drop('ni',axis=1))
    print(pca.explained_variance_ratio_)
    print(sum(pca.explained_variance_ratio_))
```

Model Building

```
#using the between estimates regression method.
be = df.groupby('gvkey').mean()
be.reset_index(inplace=True)
prediction(be,11)
```

```
0.9509331763239879
66.38957376898622
```

```
be['ni'].std()
```

```
                         BetweenOLS Estimation Summary
================================================================================
Dep. Variable:                     ni   R-squared:                        0.9945
Estimator:                 BetweenOLS   R-squared (Between):              0.9945
No. Observations:                 185   R-squared (Within):               0.0830
Date:                Mon, Feb 07 2022   R-squared (Overall):              0.4115
Time:                        11:48:32   Log-likelihood                   -868.91
Cov. Estimator:            Unadjusted
                                        F-statistic:                      5337.3
Entities:                         185   P-value                           0.0000
Avg Obs:                       12.611   Distribution:                   F(6,178)
Min Obs:                       1.0000
Max Obs:                       21.000   F-statistic (robust):             5337.3
                                        P-value                           0.0000
Time periods:                      22   Distribution:                   F(6,178)
Avg Obs:                       106.05
Min Obs:                       18.000
Max Obs:                       137.00

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const          1.5890     2.1889     0.7260     0.4688     -2.7305      5.9086
ebit          -0.1083     0.0514    -2.1079     0.0364     -0.2097     -0.0069
ebitda        -0.1596     0.0346    -4.6079     0.0000     -0.2280     -0.0913
pi             1.1974     0.0353     33.872     0.0000      1.1276      1.2671
revt           0.0146     0.0012     12.014     0.0000      0.0122      0.0170
txt           -0.7747     0.0813    -9.5298     0.0000     -0.9351     -0.6143
txp           -0.5251     0.0966    -5.4379     0.0000     -0.7157     -0.3346
==============================================================================
```

Model Building

PooledOLS Estimation Summary

Dep. Variable:	ni	R-squared:	0.6464
Estimator:	PooledOLS	R-squared (Between):	0.9649
No. Observations:	2333	R-squared (Within):	0.4616
Date:	Mon, Feb 07 2022	R-squared (Overall):	0.6464
Time:	11:45:06	Log-likelihood	-1.75e+04
Cov. Estimator:	Unadjusted		
		F-statistic:	302.63
Entities:	185	P-value	0.0000
Avg Obs:	12.611	Distribution:	F(14,2318)
Min Obs:	1.0000		
Max Obs:	21.000	F-statistic (robust):	302.63
		P-value	0.0000
Time periods:	22	Distribution:	F(14,2318)
Avg Obs:	106.05		
Min Obs:	18.000		
Max Obs:	137.00		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	12.434	10.258	1.2122	0.2256	-7.6814	32.550
ebit	1.2357	0.0865	14.289	0.0000	1.0662	1.4053
ch	0.2113	0.0242	8.7433	0.0000	0.1639	0.2587
ebitda	-0.7644	0.0805	-9.5019	0.0000	-0.9222	-0.6067
at	-0.2696	0.0399	-6.7495	0.0000	-0.3479	-0.1913
dvt	0.7601	0.1066	7.1285	0.0000	0.5510	0.9691
icapt	-0.1178	0.0173	-6.7967	0.0000	-0.1517	-0.0838
seq	0.4200	0.0405	10.357	0.0000	0.3405	0.4995
lt	0.2853	0.0396	7.2091	0.0000	0.2077	0.3629
ap	0.1391	0.0182	7.6487	0.0000	0.1034	0.1747
revt	-0.0345	0.0051	-6.7687	0.0000	-0.0444	-0.0245
cshpri	0.1231	0.0336	3.6638	0.0003	0.0572	0.1890
pi	0.1154	0.0329	3.5119	0.0005	0.0510	0.1799
txt	-0.2032	0.0529	-3.8457	0.0001	-0.3069	-0.0996
invt	0.0359	0.0120	2.9979	0.0027	0.0124	0.0594

RandomEffects Estimation Summary

Dep. Variable:	ni	R-squared:	0.6464
Estimator:	RandomEffects	R-squared (Between):	0.9649
No. Observations:	2333	R-squared (Within):	0.4616
Date:	Mon, Feb 07 2022	R-squared (Overall):	0.6464
Time:	11:47:45	Log-likelihood	-1.75e+04
Cov. Estimator:	Unadjusted		
		F-statistic:	302.63
Entities:	185	P-value	0.0000
Avg Obs:	12.611	Distribution:	F(14,2318)
Min Obs:	1.0000		
Max Obs:	21.000	F-statistic (robust):	302.63
		P-value	0.0000
Time periods:	22	Distribution:	F(14,2318)
Avg Obs:	106.05		
Min Obs:	18.000		
Max Obs:	137.00		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	12.434	10.258	1.2122	0.2256	-7.6814	32.550
ebit	1.2357	0.0865	14.289	0.0000	1.0662	1.4053
ch	0.2113	0.0242	8.7433	0.0000	0.1639	0.2587
ebitda	-0.7644	0.0805	-9.5019	0.0000	-0.9222	-0.6067
at	-0.2696	0.0399	-6.7495	0.0000	-0.3479	-0.1913
dvt	0.7601	0.1066	7.1285	0.0000	0.5510	0.9691
icapt	-0.1178	0.0173	-6.7967	0.0000	-0.1517	-0.0838
seq	0.4200	0.0405	10.357	0.0000	0.3405	0.4995
lt	0.2853	0.0396	7.2091	0.0000	0.2077	0.3629
ap	0.1391	0.0182	7.6487	0.0000	0.1034	0.1747
revt	-0.0345	0.0051	-6.7687	0.0000	-0.0444	-0.0245
cshpri	0.1231	0.0336	3.6638	0.0003	0.0572	0.1890
pi	0.1154	0.0329	3.5119	0.0005	0.0510	0.1799
txt	-0.2032	0.0529	-3.8457	0.0001	-0.3069	-0.0996
invt	0.0359	0.0120	2.9979	0.0027	0.0124	0.0594

Test set - Rsquared: 0.7, RMSE: 1000

Model Building

OLS Regression Results

Dep. Variable:	ni	R-squared:	0.654
Model:	OLS	Adj. R-squared:	0.651
Method:	Least Squares	F-statistic:	188.5
Date:	Mon, 07 Feb 2022	Prob (F-statistic):	0.00
Time:	01:33:11	Log-Likelihood:	-12188.
No. Observations:	1613	AIC:	2.441e+04
Df Residuals:	1596	BIC:	2.450e+04
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	33.0224	16.664	1.982	0.048	0.337	65.708
prev3_ni	0.3754	0.027	14.106	0.000	0.323	0.428
citotal	0.2763	0.057	4.852	0.000	0.165	0.388
ebit	1.2612	0.084	14.930	0.000	1.096	1.427
prev_ni	-0.1159	0.062	-1.859	0.063	-0.238	0.006
dltt	-0.0897	0.018	-5.113	0.000	-0.124	-0.055
ap	0.1150	0.020	5.799	0.000	0.076	0.154
ebitda	-0.9128	0.077	-11.783	0.000	-1.065	-0.761
ch	0.2440	0.025	9.921	0.000	0.196	0.292
acominc	0.3432	0.038	9.012	0.000	0.269	0.418
lt	0.3315	0.045	7.442	0.000	0.244	0.419
at	-0.2968	0.044	-6.785	0.000	-0.383	-0.211
seq	0.2613	0.048	5.426	0.000	0.167	0.356
txp	-0.8841	0.241	-3.663	0.000	-1.358	-0.411
cshpri	0.1673	0.039	4.270	0.000	0.090	0.244
invt	0.0272	0.012	2.278	0.023	0.004	0.051
t90ret	-2030.5552	770.504	-2.635	0.008	-3541.862	-519.249

Test Set R2 is:0.7157

Model Results

Model	R-square
Stepwise Regression	0.72
Panel Regression	0.69
Random Forest	0.72
Gradient Boosting	0.43

Conclusion and Next Steps

- Stepwise regression model with lag variables gave the best results.

- Using more complex models like Neural Nets.
- Using Video and Forum sentiments.
- Include Stock Buyings and IPOs.
- Board Members’ stock holdings.

Questions?