In [6]:
```
#DCF Model to calculate the fair value of Tesla
!pip install yfinance
```

```
Collecting yfinance
  Downloading yfinance-0.1.72-py2.py3-none-any.whl (27 kB)
Requirement already satisfied: pandas>=0.24.0 in c:\users\aldan\anaconda3\lib\site-pa
ckages (from yfinance) (1.4.2)
Requirement already satisfied: requests>=2.26 in c:\users\aldan\anaconda3\lib\site-pa
ckages (from yfinance) (2.27.1)
Requirement already satisfied: numpy>=1.15 in c:\users\aldan\anaconda3\lib\site-packa
ges (from yfinance) (1.21.5)
Collecting multitasking>=0.0.7
  Downloading multitasking-0.0.10.tar.gz (8.2 kB)
Requirement already satisfied: lxml>=4.5.1 in c:\users\aldan\anaconda3\lib\site-packa
ges (from yfinance) (4.8.0)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\aldan\anaconda3\lib
\site-packages (from pandas>=0.24.0->yfinance) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\aldan\anaconda3\lib\site-pack
ages (from pandas>=0.24.0->yfinance) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\aldan\anaconda3\lib\site-packages
(from python-dateutil>=2.8.1->pandas>=0.24.0->yfinance) (1.16.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\aldan\anaconda3\lib\site-pack
ages (from requests>=2.26->yfinance) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\aldan\anaconda3\lib
\site-packages (from requests>=2.26->yfinance) (1.26.9)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\aldan\anaconda3\lib\sit
e-packages (from requests>=2.26->yfinance) (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\aldan\anaconda3
\lib\site-packages (from requests>=2.26->yfinance) (2.0.4)
Building wheels for collected packages: multitasking
  Building wheel for multitasking (setup.py): started
  Building wheel for multitasking (setup.py): finished with status 'done'
  Created wheel for multitasking: filename=multitasking-0.0.10-py3-none-any.whl size=
8500 sha256=ca3b19e538bf21bf7af314a6205fa7da4ca5eeca0f1aaef9543fe82cbe2af2fe
  Stored in directory: c:\users\aldan\appdata\local\pip\cache\wheels\f2\b5\2c\59ba95d
cf854e542944c75fe3da584e4e3833b319735a0546c
Successfully built multitasking
Installing collected packages: multitasking, yfinance
Successfully installed multitasking-0.0.10 yfinance-0.1.72
```

In [7]:
```
import yfinance as yf
```

In [9]:
```
#Tesla Fair Value Calculation
tsla = yf.Ticker("tsla")
outstandingshares = tsla.info['sharesOutstanding']
```

In [21]:
```
#1. Required Rate of Return Calculation
#Cost of Debt
interest_expense = 371000
LT_Debt = 8873000
cost_of_debt = interest_expense/LT_Debt

income_tax_expense = 699000
income_before_tax = 6343000
effective_taxrate = income_tax_expense/income_before_tax

actual_costofdebt = cost_of_debt*(1-effective_taxrate)
print(actual_costofdebt)
```

```
0.03720452136982496
```

In [25]:

```
#Cost of Equity
Riskfree_rate = 0.0323
Beta = 2.13
Market_return = 0.0851
cost_of_equity = Riskfree_rate+(Beta*(Market_return-Riskfree_rate))
print(cost_of_equity)
```

```
0.14476399999999998
```

In [26]:

```
#Weight of Debt and Equity
total_debt = 8873000
market_cap = 72417000
total = total_debt+market_cap
print(total)
```

```
81290000
```

In [28]:

```
#Weight of Debt and Weight of Equity
weight_of_debt = total_debt/total
print(weight_of_debt)
weight_of_equity = market_cap/total
print(weight_of_equity)
```

```
0.1091524172714971
0.8908475827285028
```

In [35]:

```
#Required Rate or Return
required_rate_of_return = (weight_of_debt*actual_costofdebt)+(weight_of_equity*cost_of
print(required_rate_of_return)
```

```
0.13302362290705444
```

In [37]:

```
#Assumptions
required_rate = required_rate_of_return
perpetual_rate = 0.02
cashflowgrowthrate = 0.7
```

In [38]:

```
years = [1, 2, 3, 4]
```

In [39]:

```
#Cashflow for the last four years
freecashflow = [21461268, 24578000, 31536000, 53823000]
```

In [40]:

```
futurefreecashflow = []
discountfactor = []
discountedfuturefreecashflow = []
```

In [41]:

```
terminalvalue = freecashflow[-1] * (1+perpetual_rate)/(required_rate-perpetual_rate)
print(terminalvalue)
```

```
485734385.32533026
```

In [47]:

```
#Discount Factor Calculation
for year in years:
    cashflow = freecashflow[-1] * (1+cashflowgrowthrate)**year
    futurefreecashflow.append(cashflow)
    discountfactor.append((1 + required_rate)**year)
```

```
#print(discountfactor)
#print(futurefreecashflow)
```

In [48]:

```
for i in range(0, len(years)):
    discountedfuturefreecashflow.append(futurefreecashflow[i]/discountfactor[i])
    print(discountedfuturefreecashflow)

#Values below are represented from the first future year to the final
```

```
[80756568.66291654]
[80756568.66291654, 121167965.0374069]
[80756568.66291654, 121167965.0374069, 181801629.19735467]
[80756568.66291654, 121167965.0374069, 181801629.19735467, 272776986.62850946]
```

In [52]:

```
discountedterminalvalue = terminalvalue/(1 + required_rate)**4
discountedfuturefreecashflow.append(discountedterminalvalue)

current_value = sum(discountedfuturefreecashflow)

fair_value = current_value*1000/outstandingshares
print("The fair value of Tesla is ${}".format(round(fair_value,2)))
```

```
The fair value of Tesla is $1487.18
```