



Group Project 1

Tax Harvesting App

Jerami & Andres

GOAL



- We came up with the idea of building a tax harvesting app because capital gain taxes are a pain in the a**, and everyone dislikes them. So we thought it would be a good idea to create an app that could handle all your taxes in one place, but not only that, also give you the best possible solutions so that you can take home the most amount of liquidity while paying the least amount in taxes.

Summary

- This app will display a portfolio, the app will tell you the number of shares to sell from a certain ticker and we will determine the best way for you to take out liquidity from your portfolio while paying the least percentage in taxes.

Questions:

- How can we distinguish between short term capital gains tax and long term capital gains tax while also giving the user the best option to take liquidity?
- Can we help people discover how much they would pay in taxes with a simple app that is user friendly, is it possible to make taxes simple and easy?



Data Collection

Collecting data

- Alpaca API
- Import CSV files

Importing dependencies

```
# Initial imports
import os
import requests
import pandas as pd
import numpy as np
from pathlib import Path
from dotenv import load_dotenv
import alpaca_trade_api as tradeapi
import streamlit as st
import hvplot.pandas
```

Python

```
# Load .env environment variables
load_dotenv()
```

Python

True

+ Code

+ Markdown

```
csv_name = input("enter csv name to load portfolio data")
filepath = Path(csv_name+'.csv')
```

Python

```
# read in portfolio.csv format date,ticker,shares,share_price, clean data, set date as index,
portfolio_df = pd.read_csv(Path(filepath), index_col=0, parse_dates=True, infer_datetime_format=True)
```

Python

```
# add profit/loss column, (takes the number of shares * purchase share_price) - current prices
portfolio_df['order_total'] = portfolio_df['no_of_shares']*portfolio_df['share_price']
```

Python

Working with Alpaca APIs

```
# get current prices. use api to get current price

API_KEY = os.getenv('ALPACA_API_KEY')

API_SECRET_KEY = os.getenv('ALPACA_SECRET_KEY')

type(API_KEY)

# Create the Alpaca API object
alpaca = tradeapi.REST(
    API_KEY,
    API_SECRET_KEY,
    api_version='v2'
)

# Format current date as ISO format NEED TO AUTO GRAB TODAY'S DATE
api_date = input('please enter yesterdays market close date yyyy-mm-dd')
today = pd.Timestamp(api_date, tz='America/New_York').isoformat()

# Set the tickers
tickers = portfolio_df['symbol'].unique()

# Set timeframe to "1Day" for Alpaca API
timeframe = "1Day"

stock_price_df = alpaca.get_bars(
    tickers,
    timeframe,
    start=today,
    end=today
).df

stock_price_df.head(10)

portfolio_df.reset_index(inplace=True)
```

Python

```
type(API_KEY)
```

Python

str

Loading our Dataframes with CSV files we created

```
# concat current prices to portfolio

from datetime import *

current_price_df = portfolio_df.merge(stock_price_df, how='left', on='symbol')
current_price_df['order_date'] = pd.to_datetime(current_price_df['order_date']).dt.date
current_price_df['current_value'] = current_price_df['no_of_shares']*current_price_df['close']
date_check = date.today() - current_price_df['order_date']
current_price_df['bought_days_from_today'] = date_check
current_price_df['bought_days_from_today'] = current_price_df['bought_days_from_today'].dt.days
current_price_df['tax_type'] = ''
current_price_df['PNL'] = current_price_df['current_value'] - current_price_df['order_total']
```

Python

```
current_price_df.loc[current_price_df['bought_days_from_today'] >=365, 'tax'] = 'LT'
current_price_df.loc[current_price_df['bought_days_from_today'] < 365, 'tax'] = 'ST'

current_price_df
```

Python

	order_date	symbol	no_of_shares	share_price	order_total	open	high	low	close	volume	trade_count	vwap	current_value	bought_days_from_today	tax_type	PNL
0	2020-03-16	AMZN	59	84.46	4983.14	93.05	94.0600	90.820	91.01	71535538	635423	91.975919	5369.59	995		386.45
1	2020-03-16	AAPL	83	60.50	5021.50	147.77	150.9199	145.770	146.63	68826442	586791	147.638418	12170.29	995		7148.79
2	2020-03-16	MSFT	37	135.42	5010.54	252.01	253.8190	248.060	250.20	23435294	276722	250.679535	9257.40	995		4246.86
3	2020-03-16	TSLA	169	29.67	5014.23	189.44	191.2700	180.550	182.45	93125754	1142292	184.658016	30834.05	995		25819.82
4	2021-03-05	AMZN	33	150.02	4950.66	93.05	94.0600	90.820	91.01	71535538	635423	91.975919	3003.33	641		-1947.33
5	2022-03-15	AAPL	320	155.09	49628.80	147.77	150.9199	145.770	146.63	68826442	586791	147.638418	46921.60	266		-2707.20
6	2022-01-24	MSFT	170	296.37	50382.90	252.01	253.8190	248.060	250.20	23435294	276722	250.679535	42534.00	316		-7848.90
7	2022-03-14	TSLA	200	255.46	51092.00	189.44	191.2700	180.550	182.45	93125754	1142292	184.658016	36490.00	267		-14602.00
8	2021-01-14	PTON	100	165.25	16525.00	13.06	13.3450	12.745	12.92	14984438	86742	13.031332	1292.00	691		-15233.00

Displaying balance of each tax type

```
# display balance of each tax type and ask user to input dollar total of withdrawal requested, put amount in column "amount_to_withdraw"
LT_portfolio_df = current_price_df.loc[current_price_df['tax'] == 'LT']
LT_portfolio_balance = LT_portfolio_df['current_value'].sum()
LT_portfolio_initial_investment = LT_portfolio_df['order_total'].sum()
LT_taxable_amount = LT_portfolio_balance - LT_portfolio_initial_investment
LT_negative_pnl = LT_portfolio_df[LT_portfolio_df['PNL'] < 0].sum()
LT_positive_pnl = LT_portfolio_df[LT_portfolio_df['PNL'] > 0].sum()
LT_negative_pnl = LT_negative_pnl['PNL']
LT_positive_pnl = LT_positive_pnl['PNL']
LT_max_tax_free_withdrawal_amount = abs(LT_negative_pnl) * 2
display(LT_portfolio_df)

print(f"the value of your long term gains taxable portfolio is $ {LT_portfolio_balance}")
print(f"your initial investment is $ {LT_portfolio_initial_investment}")
print(f"the taxable difference is $ {LT_portfolio_balance} - {LT_portfolio_initial_investment} = {LT_taxable_amount} ")
print(f"the positive pnl sum is $ {LT_positive_pnl}")
print(f"the negative pnl sum is $ {LT_negative_pnl}")
print(f"the max you can withdraw from your long term gains stocks and not pay tax is :$ {LT_max_tax_free_withdrawal_amount}")

ST_portfolio_df = current_price_df.loc[current_price_df['tax'] == 'ST']
ST_portfolio_balance = ST_portfolio_df['current_value'].sum()
ST_portfolio_initial_investment = ST_portfolio_df['order_total'].sum()
ST_taxable_amount = ST_portfolio_balance - ST_portfolio_initial_investment
total_portfolio_value = LT_portfolio_balance + ST_portfolio_balance
ST_negative_pnl = ST_portfolio_df[ST_portfolio_df['PNL'] < 0].sum()
ST_positive_pnl = ST_portfolio_df[ST_portfolio_df['PNL'] > 0].sum()
ST_negative_pnl = ST_negative_pnl['PNL']
ST_positive_pnl = ST_positive_pnl['PNL']
ST_max_tax_free_withdrawal_amount = abs(ST_negative_pnl) * 2

display(ST_portfolio_df)

print(f"the value of your short term gains taxable portfolio is $ {ST_portfolio_balance}")
print(f"your initial investment is $ {ST_portfolio_initial_investment}")
print(f"the taxable difference is $ {ST_portfolio_balance} - {ST_portfolio_initial_investment} = {ST_taxable_amount} ")
print(f"total portfolio value is $ {total_portfolio_value}")
print(f"the positive pnl sum is $ {ST_positive_pnl}")
print(f"the negative pnl sum is $ {ST_negative_pnl}")
print(f"the max you can withdraw from your short term gains stocks and not pay tax is :$ {ST_max_tax_free_withdrawal_amount}")
```

```
/var/folders/dx/8mBb_r_57793yl3byym44sh0000gn/T/ipykernel_29663/4055416708.py:6: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
```

```
LT_negative_pnl = LT_portfolio_df[LT_portfolio_df['PNL'] < 0].sum()
```

```
/var/folders/dx/8mBb_r_57793yl3byym44sh0000gn/T/ipykernel_29663/4055416708.py:7: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
```

```
LT_positive_pnl = LT_portfolio_df[LT_portfolio_df['PNL'] > 0].sum()
```

	order_date	symbol	no_of_shares	share_price	order_total	open	high	low	close	volume	trade_count	vwap	current_value	bought_days_from_today	tax_type	PNL
0	2020-03-16	AMZN	59	84.46	4983.14	93.05	94.0600	90.820	91.01	71535538	635423	91.975919	5369.59		995	386.45
1	2020-03-16	AAPL	83	60.50	5021.50	147.77	150.9199	145.770	146.63	68826442	586791	147.638418	12170.29		995	7148.79
2	2020-03-16	MSFT	37	135.42	5010.54	252.01	253.8190	248.060	250.20	23435294	276722	250.679535	9257.40		995	4246.86
3	2020-03-16	TSLA	169	29.67	5014.23	189.44	191.2700	180.550	182.45	93125754	1142292	184.658016	30834.05		995	25819.82
4	2021-03-05	AMZN	33	150.02	4950.66	93.05	94.0600	90.820	91.01	71535538	635423	91.975919	3003.33		641	-1947.33
8	2021-01-14	PTON	100	165.25	16525.00	13.06	13.3450	12.745	12.92	14984438	86742	13.031332	1292.00		691	-15233.00

```
the value of your long term gains taxable portfolio is $ 61926.66
```

```
your initial investment is $ 41585.07
```

```
the taxable difference is $ 61926.66 - 41585.07 = 20421.590000000004
```

```
the positive pnl sum is $ 37681.92
```

```
the negative pnl sum is $ -17180.33
```

```
the max you can withdraw from your long term gains stocks and not pay tax is: $ 34294.32
```

```
/var/folders/dx/8mBb_r_57793yl3byym44sh0000gn/T/ipykernel_29663/4055416708.py:27: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
```

```
ST_negative_pnl = ST_portfolio_df[ST_portfolio_df['PNL'] < 0].sum()
```

	order_date	symbol	no_of_shares	share_price	order_total	open	high	low	close	volume	trade_count	vwap	current_value	bc
5	2022-03-15	AAPL	320	155.09	49628.8	147.77	150.9199	145.77	146.63	68826442	586791	147.638418	46921.6	
6	2022-01-24	MSFT	170	296.37	50382.9	252.01	253.8190	248.06	250.20	23435294	276722	250.679535	42534.0	
7	2022-03-14	TSLA	200	255.46	51092.0	189.44	191.2700	180.55	182.45	93125754	1142292	184.658016	36490.0	

```
the value of your short term gains taxable portfolio is $ 125945.6
```

```
your initial investment is $ 151103.7
```

```
the taxable difference is $ 125945.6 - 151103.7 = -25158.100000000006
```

```
total portfolio value is $ 187872.26
```

```
the positive pnl sum is $ 0.0
```

```
the negative pnl sum is $ -25158.100000000006
```

```
the max you can withdraw from your short term gains stocks and not pay tax is: $ 50316.200000000001
```

Display of all portfolio balances & determining short term and long term capital gains

Determining Federal tax rates

```
# ask user for yearly income to determine tax bracket. put amount in column "taxable_income"
print("let's determine your federal tax/short term gains bracket")
user_income = input("please enter your annual taxable income")
user_income = int(user_income)
print(f"you entered ${user_income}")
```

Python

```
let's determine your federal tax/short term gains bracket
you entered $100000
```

```
# convert "taxable_income" to percentage of taxes owed from tax bracket into column "tax_pct"
if user_income <= 10275:
    tax_bracket = 0.10
elif user_income <= 41775:
    tax_bracket = 0.12
elif user_income <= 89075:
    tax_bracket = 0.22
elif user_income <= 170050:
    tax_bracket = 0.24
elif user_income <= 215950:
    tax_bracket = 0.32
elif user_income <= 539000:
    tax_bracket = 0.35
elif user_income >539000:
    tax_bracket = 0.37

print(f"your federal tax/short term gains rate is : {tax_bracket}")
```

Python

```
your federal tax/short term gains rate is : 0.24
```

Describing Long term and Short term taxes to the user

```
# describe long term and short term gain tax rates
display_tax_types = input('would you like to see information on tax types? y/n')
```

```
if display_tax_types == 'y':
    print(
    ...
```

```
"How much do I have to pay?
```

```
The tax rate you pay in 2022 depends on whether your gain is short-term or long-term.
```

```
Short-term profits are usually taxed at your maximum tax rate, just like your salary, up to 37% and could even be subject to the additional 3.8% Medicare surtax, depending on your
Long-term gains are treated much better. Long-term gains are taxed at 0%, 15% or 20% depending on your taxable income and filing status.
```

```
Long-term gains on collectibles—such as stamps, antiques and coins—are taxed at 28%, or at your ordinary-income tax rate if lower.
```

```
Gains on real estate that are attributable to depreciation—since depreciation deductions reduce your cost basis, they also increase your profit dollar for dollar—are taxed at 25%,
Long-term gains from stock sales by children under age 19—under age 24 if they are full-time students—may not qualify for the 0% rate because of the Kiddie Tax rules. (When these
...
```

```
)
```

Python

```
"How much do I have to pay?
```

```
The tax rate you pay in 2022 depends on whether your gain is short-term or long-term.
```

```
Short-term profits are usually taxed at your maximum tax rate, just like your salary, up to 37% and could even be subject to the additional 3.8% Medicare surtax, depending on your
income level.
```

```
Long-term gains are treated much better. Long-term gains are taxed at 0%, 15% or 20% depending on your taxable income and filing status.
```

```
Long-term gains on collectibles—such as stamps, antiques and coins—are taxed at 28%, or at your ordinary-income tax rate if lower.
```

```
Gains on real estate that are attributable to depreciation—since depreciation deductions reduce your cost basis, they also increase your profit dollar for dollar—are taxed at 25%, or
at your ordinary-income tax rate if lower.
```

```
Long-term gains from stock sales by children under age 19—under age 24 if they are full-time students—may not qualify for the 0% rate because of the Kiddie Tax rules. (When these
rules apply, the child's gains may be taxed at the parents' higher rates.
```

Off-setting long term and short term taxes

```
# how to offset long term and short term gains from sales
display_tax_offset_rules = input('would you like to see information on of tax loss offset rules? y/n')
```

```
if display_tax_offset_rules == 'y':
    print(
    ...
```

Almost everything you own and use for personal or investment purposes is a capital asset. Examples include a home, personal-use items like household furnishings, and stocks or bonds.

Short-Term or Long-Term

To correctly arrive at your net capital gain or loss, capital gains and losses are classified as long-term or short-term. Generally, if you hold the asset for more than one year before you sell it, the gain or loss is long-term.

If you have a net capital gain, a lower tax rate may apply to the gain than the tax rate that applies to your ordinary income. The term "net capital gain" means the amount by which your capital gains exceed your capital losses.

Capital Gain Tax Rates

The tax rate on most net capital gain is no higher than 15% for most individuals. Some or all net capital gain may be taxed at 0% if your taxable income is less than or equal to \$44,625 for single; more than \$44,625 but less than or equal to \$125,000 for married filing jointly.

A capital gain rate of 15% applies if your taxable income is more than \$40,400 but less than or equal to \$445,850 for single; more than \$80,800 but less than or equal to \$501,600 for married filing jointly.

However, a net capital gain tax rate of 20% applies to the extent that your taxable income exceeds the thresholds set for the 15% capital gain rate.

There are a few other exceptions where capital gains may be taxed at rates greater than 20%:

The taxable part of a gain from selling section 1202 qualified small business stock is taxed at a maximum 28% rate.

Net capital gains from selling collectibles (such as coins or art) are taxed at a maximum 28% rate.

The portion of any unrecaptured section 1250 gain from selling section 1250 real property is taxed at a maximum 25% rate.

Note: Net short-term capital gains are subject to taxation as ordinary income at graduated tax rates.

Limit on the Deduction and Carryover of Losses

```
If your capital losses exceed your capital gains, the amount of the excess loss that you can claim to lower your income is the lesser of $3,000 ($1,500 if married filing separately)
...
)
```

Determining long term losers/winners and short term winners/losers

```
# display all losing stocks to sell and winning stocks to net zero the gains for maximin withdrawal
#long term losers
LT_losers_df = LT_portfolio_df.loc[LT_portfolio_df['PNL'] < 0]
LT_losers_withdrawal_total = LT_losers_df['current_value'].sum()
LT_losers_losses_reported = LT_losers_df['PNL'].sum()
print(f"selling off these losing stocks will give you {LT_losers_withdrawal_total} cash and a credit loss of {LT_losers_losses_reported}")
display(LT_losers_df)

#long term winners
print(f"to maximize withdrawal and offset losses. it is recommended to")
LT_winners_df = LT_portfolio_df.loc[LT_portfolio_df['PNL'] > 0]
LT_sorted_winners_df = LT_winners_df.sort_values(['PNL'], ascending= False).reset_index(drop=True)

if LT_losers_losses_reported == 0:
    LT_scenario = 0
elif LT_sorted_winners_df['PNL'].iloc[0] > LT_losers_losses_reported:
    LT_scenario = 1
elif LT_sorted_winners_df['PNL'].iloc[0] + LT_sorted_winners_df['PNL'].iloc[1] > LT_losers_losses_reported:
    LT_scenario = 2
elif LT_sorted_winners_df['PNL'].iloc[0] + LT_sorted_winners_df['PNL'].iloc[1] + LT_sorted_winners_df['PNL'].iloc[2] > LT_losers_losses_reported:
    LT_scenario = 3
else:
    print("losses are much higher than winning stocks right now. sell whatever shares you'd like to net zero losses and profits")
    print(LT_sorted_winners_df)

if LT_scenario == 0:
    print("all of your stocks are up! there are no losses to harvest! We cannot make any sell recommendations at this time!")
elif LT_scenario == 1:
    LT_winner_stock_to_sell = LT_sorted_winners_df['symbol'].iloc[0]
    print(f"sell enough shares of {LT_winner_stock_to_sell} to offset {LT_losers_losses_reported} in losses")
    LT_offset_shares_to_sell = LT_losers_losses_reported / LT_sorted_winners_df['close'].iloc[0]
    LT_offset_shares_to_sell = round(abs(LT_offset_shares_to_sell),0)
    LT_offset_close_price = LT_sorted_winners_df['close'].iloc[0]
    LT_offset_sell_amount_total = LT_offset_shares_to_sell * LT_offset_close_price
    LT_offset_net_tax_liability = round(LT_offset_sell_amount_total - abs(LT_losers_losses_reported),2)
    LT_cash_in_bank = LT_losers_withdrawal_total + LT_offset_sell_amount_total
    print(f"selling {LT_offset_shares_to_sell} shares at {LT_offset_close_price} equals a total of {LT_offset_sell_amount_total}")
    print(f"for a total Long Term gains tax liability of ${LT_offset_net_tax_liability} taxed as long term gains")
    print(f"liquidity gained from long term gains tax harvesting transactions totals ${LT_cash_in_bank} ")
    LT_sorted_winners_df['no_of_shares'].iloc[0] = LT_sorted_winners_df['no_of_shares'].iloc[0] - LT_offset_shares_to_sell
    print("current Long Term gain stock holdings")
    after_harvest_LT_holdings = LT_sorted_winners_df
    display(after_harvest_LT_holdings)
```

PNL after selling certain stocks

```
#short term losers
ST_losers_df = ST_portfolio_df.loc[ST_portfolio_df['PNL'] < 0]
ST_losers_withdrawal_total = ST_losers_df['current_value'].sum()
ST_losers_losses_reported = ST_losers_df['PNL'].sum()
print(f"selling off these losing stocks will give you {ST_losers_withdrawal_total} cash and a credit loss of {ST_losers_losses_reported}")
display(ST_losers_df)

#short term winners
ST_winners_df = ST_portfolio_df.loc[ST_portfolio_df['PNL'] > 0]
ST_sorted_winners_df = ST_winners_df.sort_values(['PNL'], ascending= False).reset_index(drop=True)
```

Python

selling off these losing stocks will give you 4295.33 cash and a credit loss of -17180.33

	order_date	symbol	no_of_shares	share_price	order_total	open	high	low	close	volume	trade_count	vwap	current_value	bought_days_from_today	tax_type	PNL	tax
4	2021-03-05	AMZN	33	150.02	4950.66	93.05	94.060	90.820	91.01	71535538	635423	91.975919	3003.33	641		-1947.33	LT
8	2021-01-14	PTON	100	165.25	16525.00	13.06	13.345	12.745	12.92	14984438	86742	13.031332	1292.00	691		-15233.00	LT

to maximize withdrawal and offset losses, it is recommended to

sell enough shares of TSLA to offset -17180.33 in losses

selling 94.0 shares at 182.45 equals a total of 17150.3

for a total Long Term gains tax liability of \$-30.03 taxed as long term gains

liquidity gained from long term gains tax harvesting transactions totals \$21445.629999999997

current Long Term gain stock holdings

/var/folders/dx/8m8b_r_57793yl3byvym44sh0000gn/T/ipykernel_29663/2091429246.py:40: 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

```
LT_sorted_winners_df['no_of_shares'].iloc[0] = LT_sorted_winners_df['no_of_shares'].iloc[0] - LT_offset_shares_to_sell
```

	order_date	symbol	no_of_shares	share_price	order_total	open	high	low	close	volume	trade_count	vwap	current_value	bought_days_from_today	tax_type	PNL
0	2020-03-16	TSLA	75	29.67	5014.23	189.44	191.2700	180.55	182.45	93125754	1142292	184.658016	30834.05	995		25819.82
1	2020-03-16	AAPL	83	60.50	5021.50	147.77	150.9199	145.77	146.63	68826442	586791	147.638418	12170.29	995		7148.79
2	2020-03-16	MSFT	37	135.42	5010.54	252.01	253.8190	248.06	250.20	23435294	276722	250.679535	9257.40	995		4246.86
3	2020-03-16	AMZN	50	94.46	4923.14	93.05	94.0600	90.820	91.01	71535538	635423	91.975919	5000.50	995		286.45

Tax Harvesting App

Hi, welcome to our project 🙌

Tax Harvesting for portfolios

This app will display your portfolio, you will choose the number of shares you wish to sell from a certain ticker and we will determine the best way for you to take liquidity from your portfolio while paying the least percentage in taxes.

Portfolio

	order_date	symbol	no_of_shares	share_price
0	16-mar-20	AMZN	59	84.4600
1	16-mar-20	AAPL	83	60.5000
2	16-mar-20	MSFT	37	135.4200
3	16-mar-20	TSLA	169	29.6700
4	05-mar-21	AMZN	33	150.0200
5	15-MAR-22	AAPL	320	155.0900
6	24-JAN-22	MSFT	170	296.3700
7	14-MAR-22	TSLA	200	255.4600
8	14-jan-21	PTON	100	165.2500

Withdrawal in USD

Below enter the date in which you bought the stock then enter the ticker you would like to withdrawal from, as well as the number of shares you would like to sell.

Select order_date (YYYY/MM/DD)

16-mar-20



select ticker:

AMZN



Select amount of shares you wish to sell:

0



Submit Withdrawal Request