CASSANDRA

An advanced barometer of risk
in equity markets

Scope

01

Problem statement

Assessing risks in equity markets
Defining our target variable
Definitions of success

02

Data collection / EDA

Selection of features
Panda-profiling

03

Model creation

Leveraging on *pycaret*
Pre-processing, model tuning

04

Conclusions

Model interpretation
Employing out barometer

01

Problem statement

Defining our problem statement

Timing is absolutely key in financial markets. The ability to predict market peaks can help protect investors from significant drawdowns and elevated volatility.

We may be able to leverage on the advances made in machine learning to create a more advanced barometer of risk for equity markets **(Predicting pullbacks)** to help investors navigate through the noise.

While this tool should be used in conjunction with other qualitative evidence, we believe it can be a useful input to the investment process

Defining our target + success metrics

TARGET (Pullback)	Defined as whether the S&P 500 index will be lower in a month's time (i.e. rolling 4 weeks).

The model we are looking to build should be able to do the following:

- Predict the likelihood that the S&P 500 index will fall over the coming month
- Explain which variables matter and how they interact

We want a model that is competent at predicting when there will not be pullbacks, and when there will be pullbacks.

We will focus on maximising **accuracy**. A confusion matrix is also a good way to see whether our model is biased towards either of our target classes

02

Data collection EDA, Feature engineering

Picking our features

Sourced from **Bloomberg**

CASSANDRA
Selected variables

- **Macro** barometers
  - Citi economic surprise indices
  - USD indices
  - US economic policy uncertainty index
  - Chicago Fed Financial Conditions
  - Inflation expectations; Breakevens / Swaps
- **Fundamental** indicators
  - Earnings revision ratios
  - 12-month forward P/E ratios
  - 12-month forward P/B ratios
- **Sentiment** indicators
  - VIX, MOVE, SKEW, CVIX
  - Put/call ratios (Aggregated)
  - AAII sentiment (Bulls vs. Bears)
  - CFTC net non-commercial longs (S&P 500)
- **Commodities**
  - Brent / WTI front-end contracts; Energy
  - Gold spot; Precious metals
  - Industrial metals
- **Credit**
  - US High-yield + US IG + EM Sovereigns (Hard) + Asia Credit option-adjusted spreads
- **Rates**
  - 2-year, 10-year US treasury yields
  - US 10-year TIPS
  - Fed funds rate, Eurodollar futures
- **Technical** indicators
  - 9/14-day RSIs; Breadth of 14-day RSIs
  - % of S&P 500 stocks at 52-week highs

Utilising panda-profiling

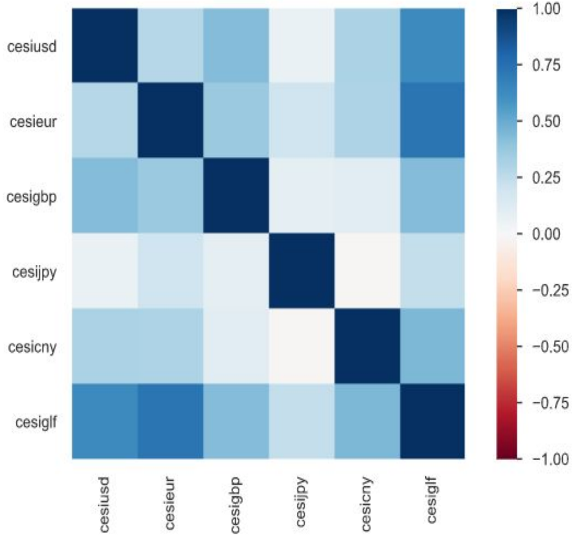
Our focus is on removing multicollinearity here

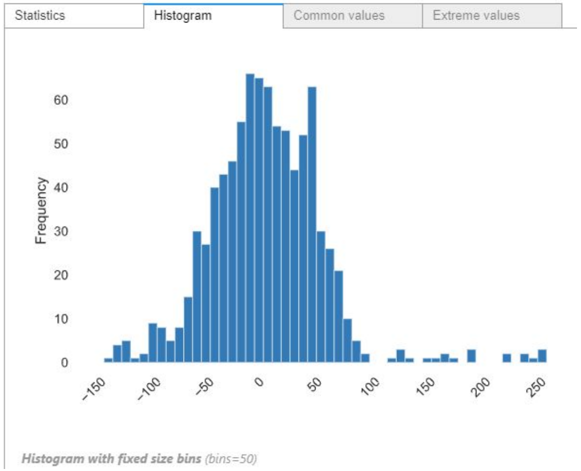
Thoughts: US economic surprises are highly correlated with Euro area, UK, China and global indices

Action: Keep ***cesiusd*** and drop the rest

Pros: Warnings provided

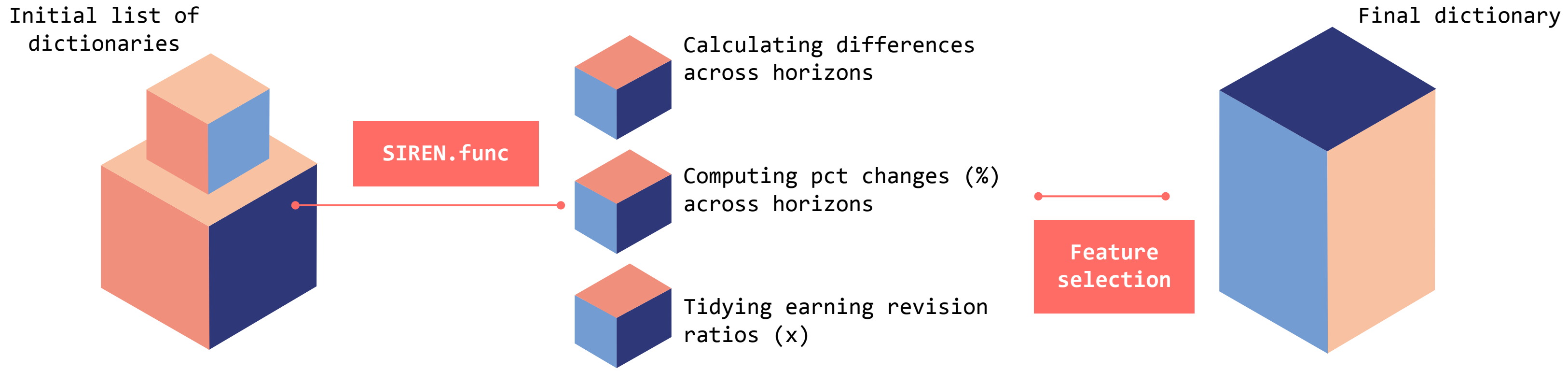
Cons: Limited use with larger datasets; Quadratic increase in time as size *n* increases





Creating our own custom module

- `eda_clean`: Provides a quick snapshot of our project
- `derive_yield_curves`: Calculating 30y10ys, 30y5ys, 30y2ys, 30y3ms, 10y5ys, 10y2ys, 10y3ms, 5y2ys, 5y3ms, 2y3ms for US and Euro-area regions
- `fix_credit`: Standardising credit spreads as pp; Calculating spread between US high-yield and investment-grade bonds
- `fix_cftc`: Deriving CFTC net non-commercial positions as a % of total open interest
- `eri_diff`: Derive earnings revision indices and rolling changes across different horizons (4, 13-week)
- `roll_diff`: Calculate rolling differences for different time horizons (1, 4, 13, 26-week)
- `lag_roll_pct_chg`: Lagging rolling percentage changes for various equity indices (4-week)
- `roll_pct_chg`: Calculate rolling percentage changes for different time horizons (1, 4, 13, 26-week)
- `adjust_dates_only`: Standardise dates for merging dataframes later



02

Creating our model

Leveraging on pycaret

Set-up

Explore feature interaction, remove multicollinearity (tweak threshold), normalise (z-score), fix imbalance dataset (SMOTE), time-series fold (10x)

Comparison

Our focus is on tree-based models (dt, et, rf, xgboost, lightgbm, etc)

Create model

Single, stacking, ensemble methods; (Simplicity + Explainability are preferred)

Evaluation

ROC-AUC curves, confusion matrices, SHAPplots

Set-up parameters + model comparison | tuning

```
1  # Setting up our pycaret environment
2  pyc = setup(train, target='spx_class', test_data=test, feature_interaction=False,
   session_id=42, remove_multicollinearity=True, multicollinearity_threshold=0.7, \
3      normalize=False, fix_imbalance=True, ignore_features=['date', 'spx', 'fdtr'],
       data_split_shuffle=False, fold_strategy='timeseries', fold=20, remove_outliers=False)
```

✓ 2.6s Python

	Model	Accuracy	AUC	Recall	Prec.	F1
xgboost	Extreme Gradient Boosting	0.6815	0.6414	0.3409	0.4167	0.3750
lda	Linear Discriminant Analysis	0.6879	0.6311	0.5227	0.4510	0.4842
catboost	CatBoost Classifier	0.6943	0.6229	0.1818	0.4000	0.2500
lr	Logistic Regression	0.6752	0.6142	0.3182	0.4000	0.3544
lightgbm	Light Gradient Boosting Machine	0.7006	0.6110	0.3182	0.4516	0.3733
dt	Decision Tree Classifier	0.5605	0.5975	0.6818	0.3529	0.4651
ridge	Ridge Classifier	0.6624	0.5920	0.4318	0.4043	0.4176
gbc	Gradient Boosting Classifier	0.6688	0.5887	0.3182	0.3889	0.3500
ada	Ada Boost Classifier	0.6242	0.5817	0.2955	0.3171	0.3059
qda	Quadratic Discriminant Analysis	0.7261	0.5718	0.2273	0.5263	0.3175
knn	K Neighbors Classifier	0.5732	0.5298	0.5227	0.3333	0.4071
svm	SVM - Linear Kernel	0.3121	0.5221	1.0000	0.2895	0.4490
nb	Naive Bayes	0.6752	0.5173	0.1591	0.3333	0.2154
et	Extra Trees Classifier	0.6752	0.5067	0.0909	0.2667	0.1356
rf	Random Forest Classifier	0.6815	0.5044	0.1591	0.3500	0.2188
dummy	Dummy Classifier	0.2803	0.5000	1.0000	0.2803	0.4378

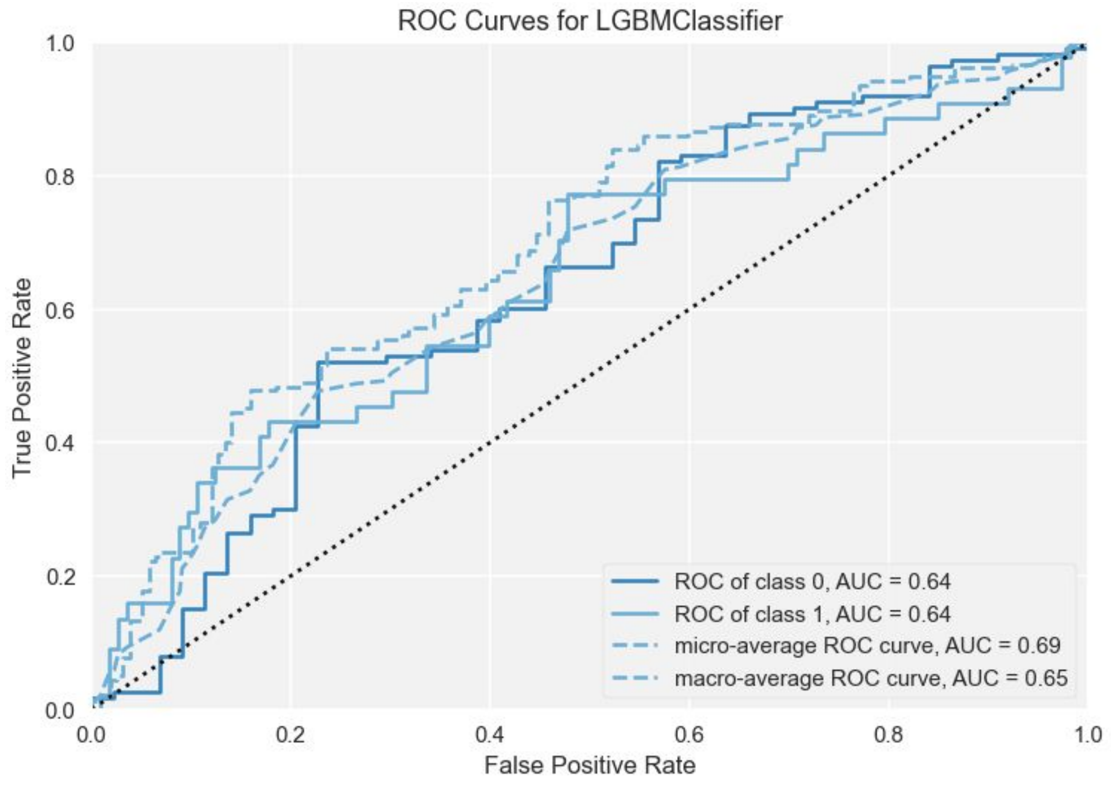
Notes:

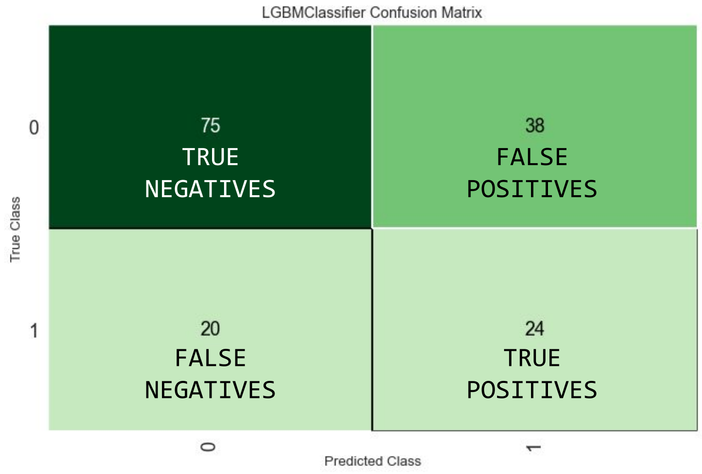
* Did not pursue feature interaction
* Focus on tree-based models
* Creation of model done without cross-validation (given dimensions of df)
* Tuning done (fold_strategy = time-series, folds = 20, n_iter = 100); Optimising **recall**

Other considerations:

* Stacking / Ensemble models did not improve the score significantly so I scrapped that
* Remember my preference is for simplicity!

Model evaluation





Metrics post-tuning

Accuracy: 63%

Precision: 39%

Recall: 55%

Specificity: 66%

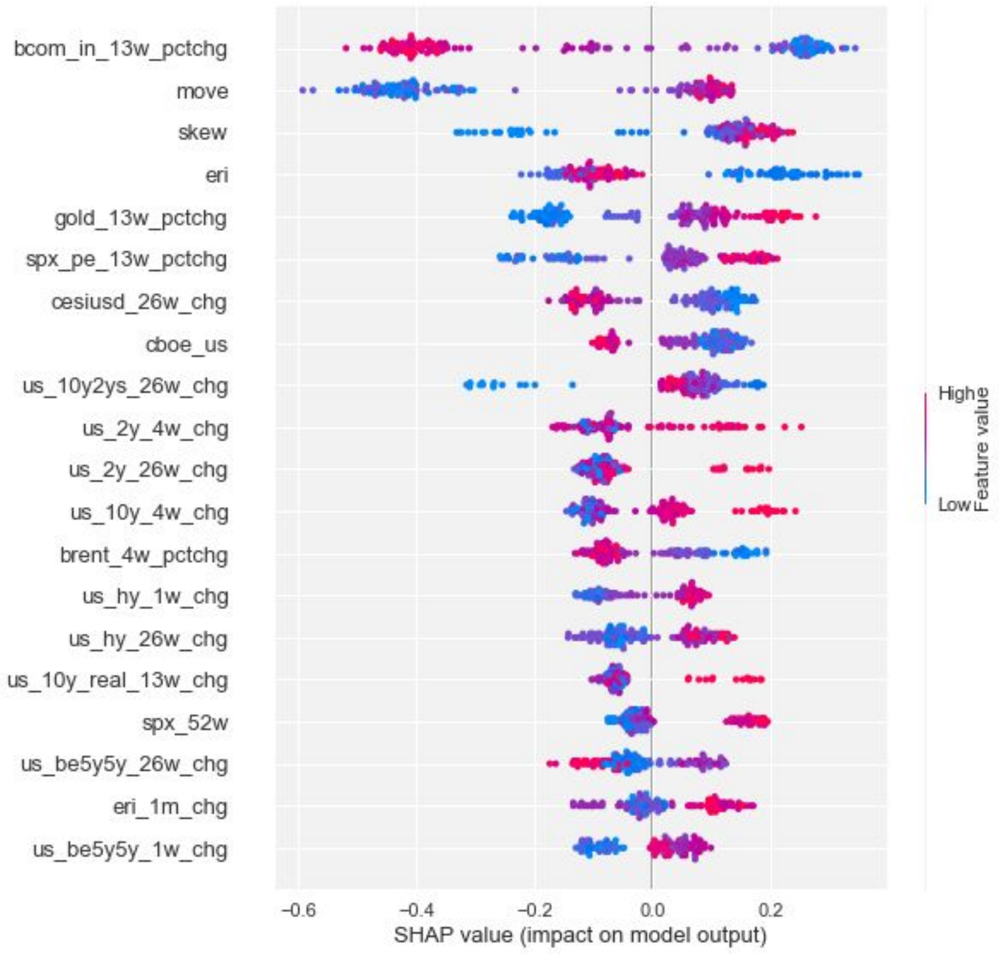
F1: 46%

02

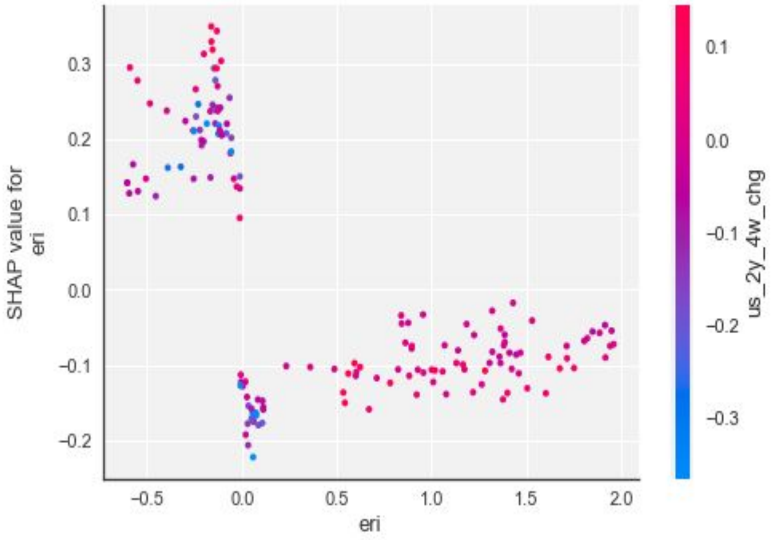
Conclusions + Takeaways

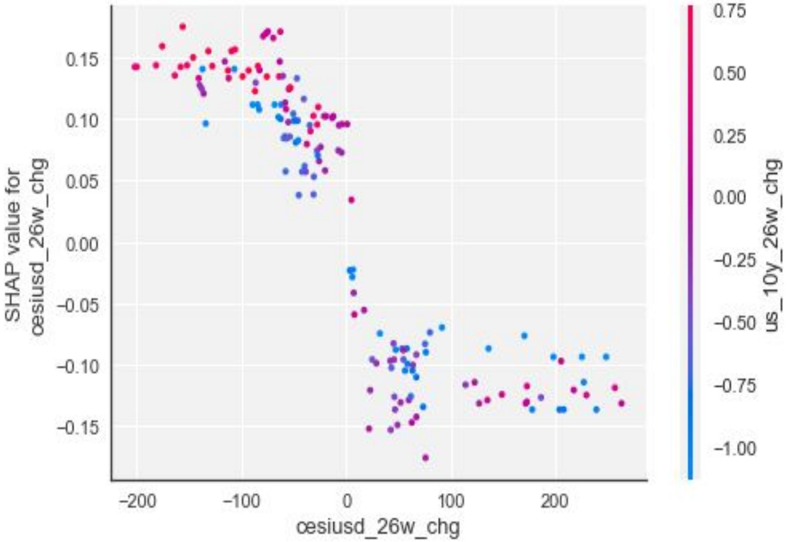
SHAP plots

Notes:
* SHAP values highlight the value to which a feature has contributed to the model's prediction; It makes correlations transparent
* Interpretability of a model is of utmost importance in finance; Keep it simple!
* Our focus is on the top features: Industrial metals + gold 13-week % change, MOVE / SKEW indices, earnings revision indices, multiple re-ratings



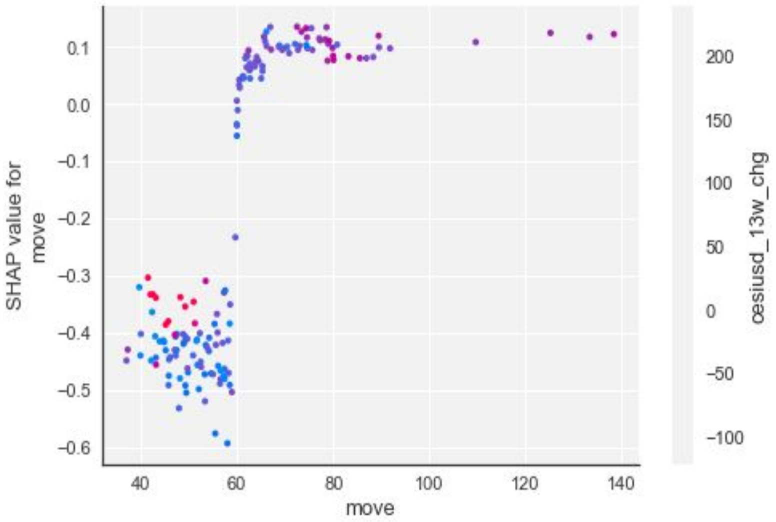
Quant insights

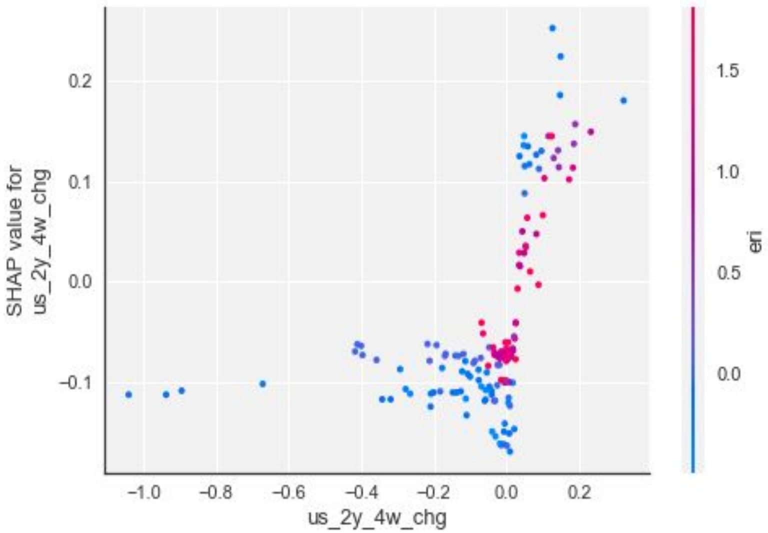




Fundamentals still matter: Earnings revision (eri) and economic surprise indices (cesiusd); US economic data has been surprising to the upside… but that has raised fears of a more hawkish Fed… An earnings revision ratio below 0 - typically increases the likelihood of a decline in equities

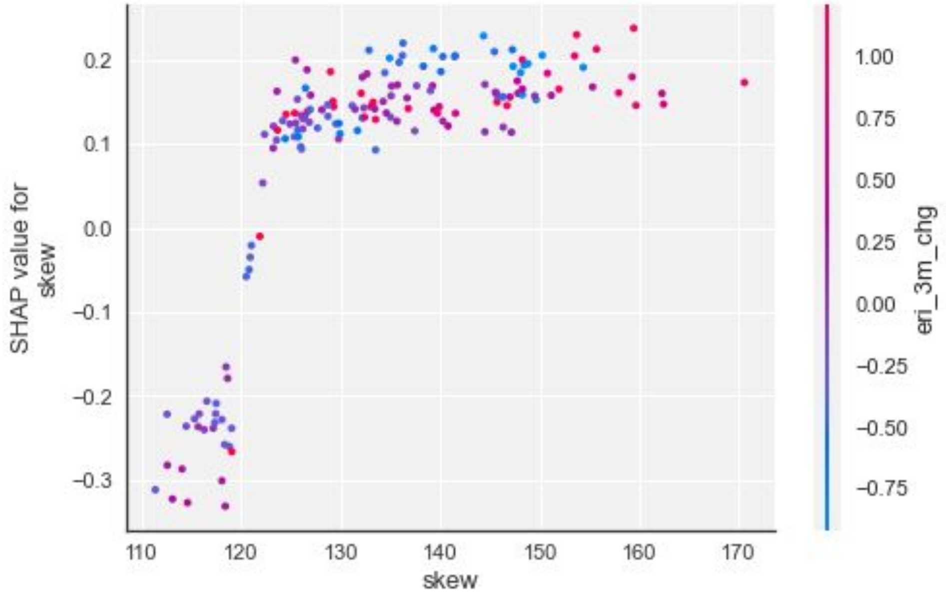
Quant insights II





Watch the Fed: Given the flood of liquidity on the back of QE over the past decade, whatever the Fed decides to do over the coming months is key; Higher rates = Lower multiples = Tighter liquidity

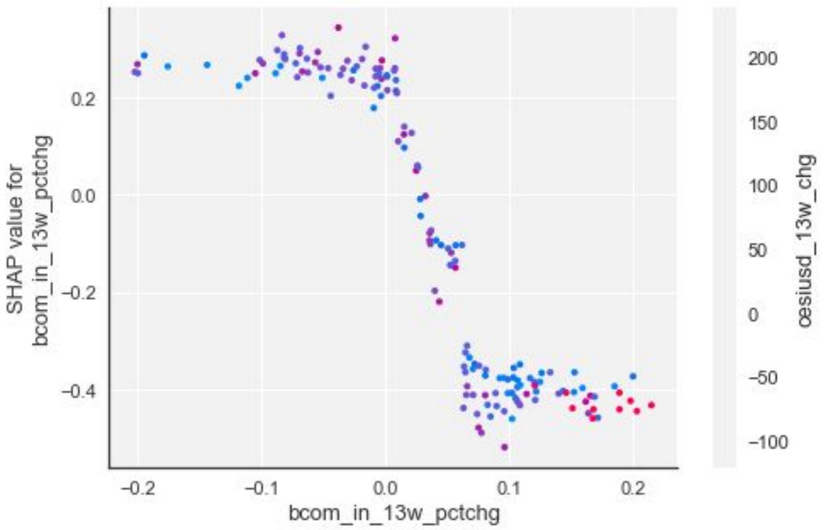
Quant insights III

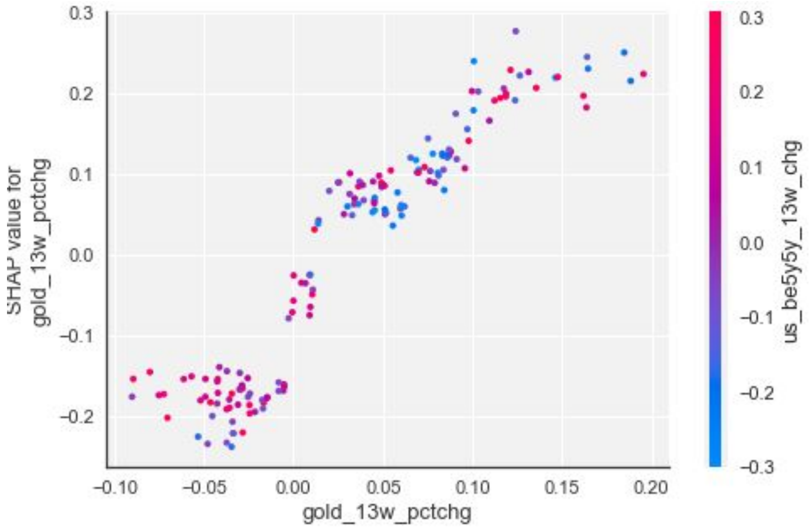


Pay attention to the SKEW: The SKEW index is often overlooked by most people (myself included). It is a measure of perceived tail risk in the S&P 500 based on deep OTM options. Unlike other sentiment indicators, the direction of travel does not appear to be contrarian.

Whenever the cost of tail-risk protection rises, CASSANDRA believes there is an increased likelihood of a drawdown in equity markets

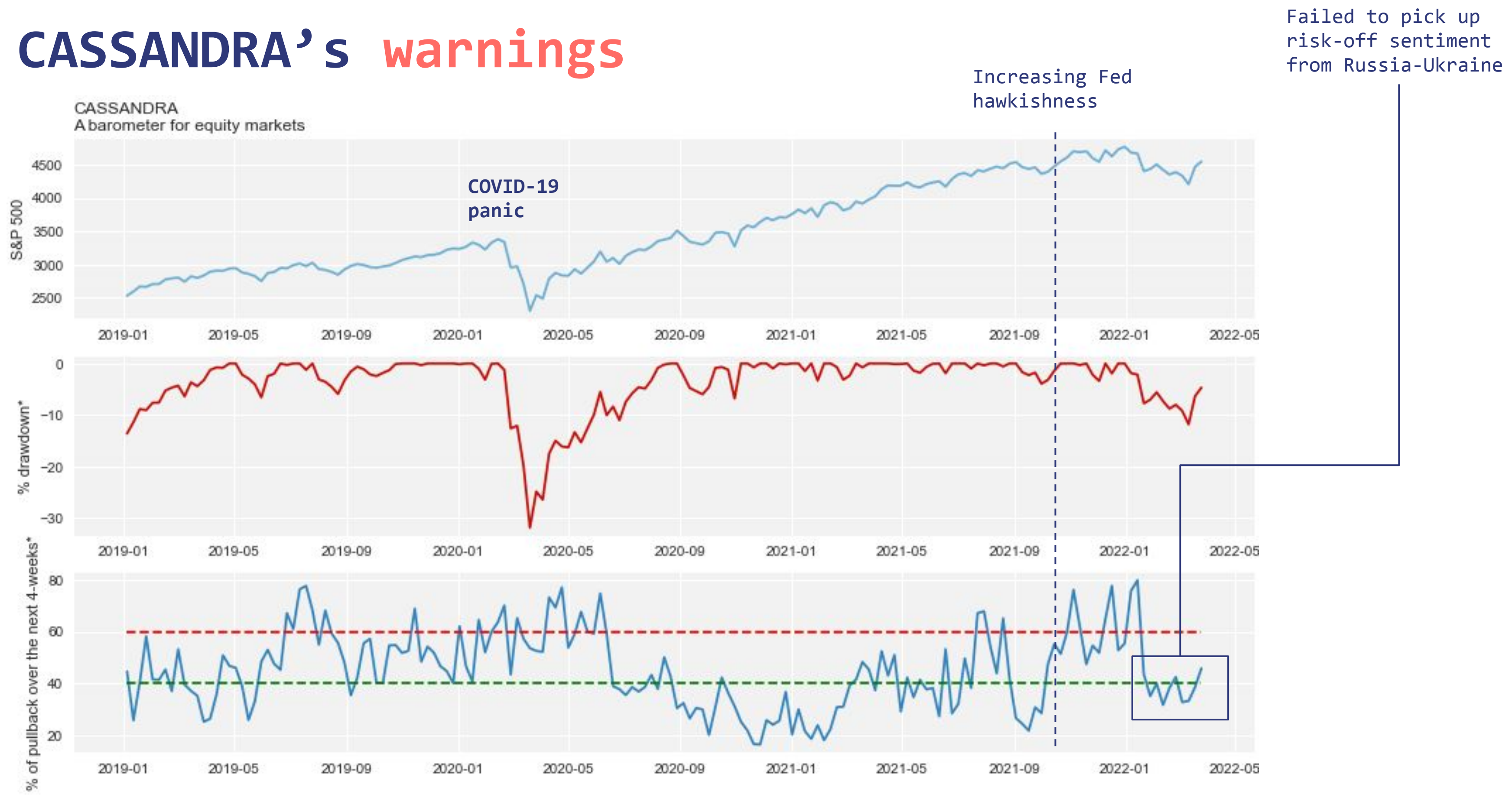
Quant insights IV





Commodities: They're known to be important drivers of equity markets contemptuously. A fall in the price of industrial metals increases the likelihood of a drawdown in equity markets whereas the safe-haven properties of gold are illustrated here

CASSANDRA's warnings



DJ Cheong

DATA SCIENCE ENTHUSIAST

THANK YOU

https://www.linkedin.com/in/djcheong/
https://github.com/deltajuliette