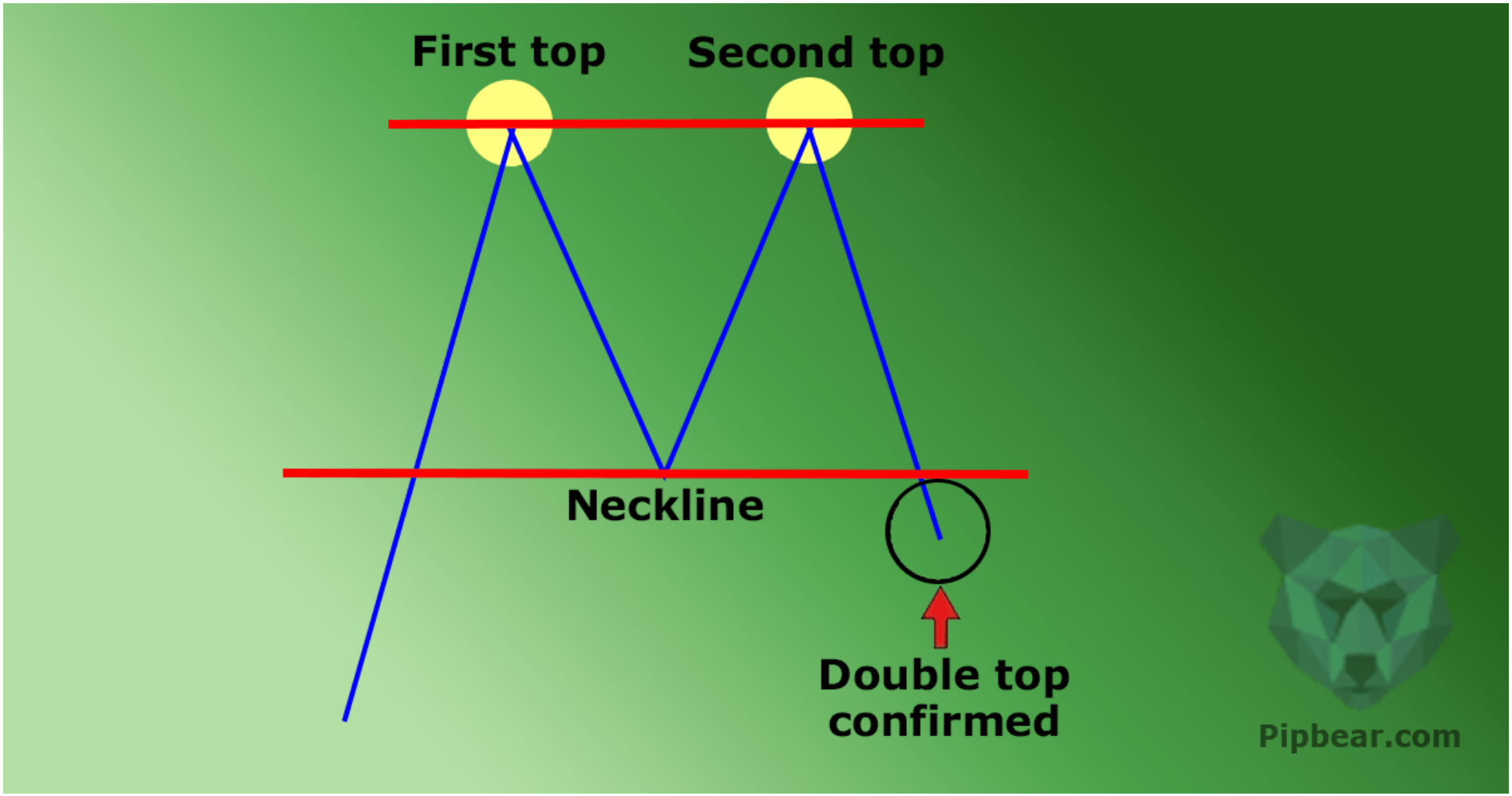
Stock Price Prediction Base on Double Top Pattern

Goal

To build a prediction model that can predict the movement of stocks prices given that a double top stock chart pattern just occur.



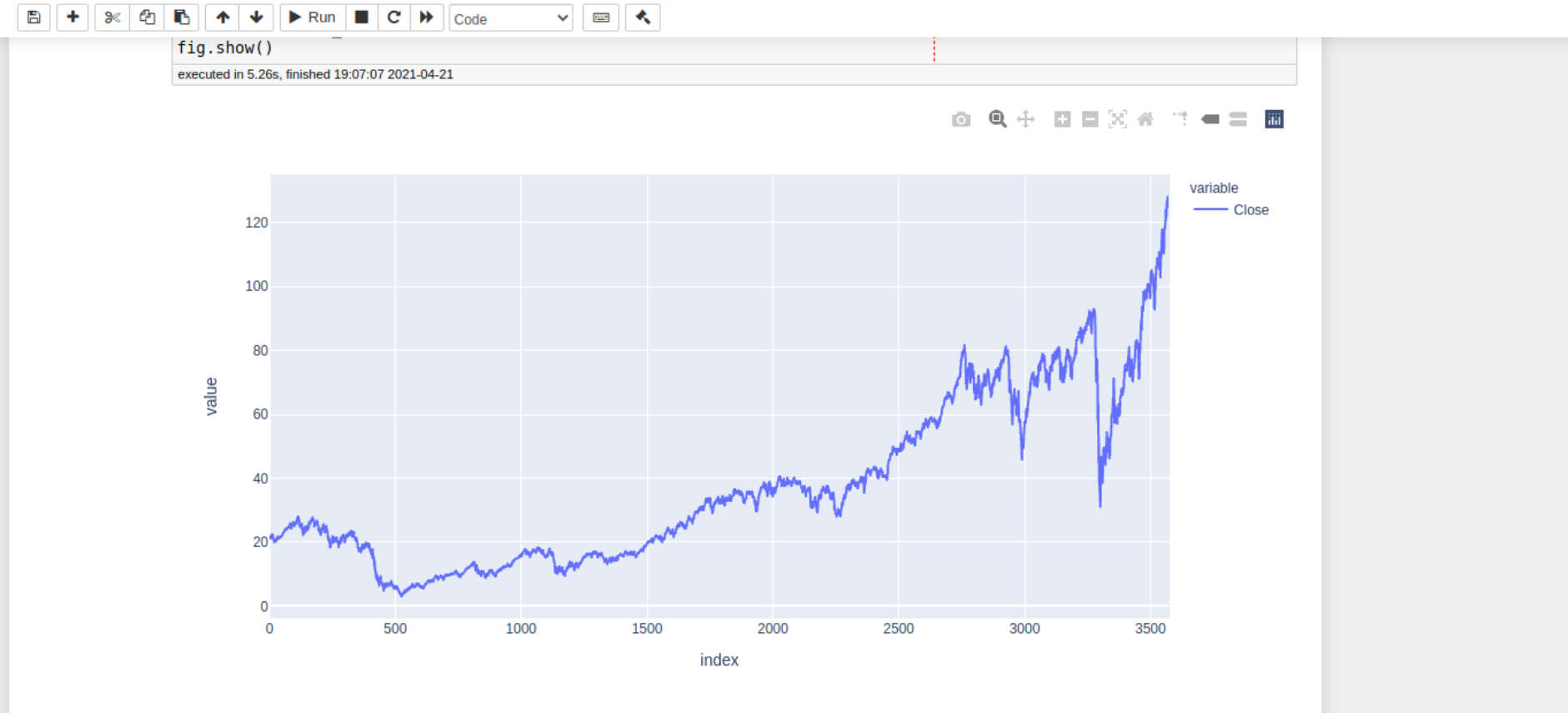
Create dataset

1.) Web scraping from *yahoofinance.com*

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2007-02-02	21.500963	21.500963	21.500963	21.500963	0	0.0	0.0
1	2007-02-05	21.500963	21.500963	21.500963	21.500963	0	0.0	0.0
2	2007-02-06	21.580568	21.580568	21.580568	21.580568	900	0.0	0.0
3	2007-02-07	21.698502	21.698502	21.698502	21.698502	1200	0.0	0.0
4	2007-02-08	21.562882	21.562882	21.539297	21.539297	45000	0.0	0.0
...	...	...	...	...	...	...	...	...
3573	2021-04-14	127.440002	127.440002	125.639999	125.639999	1700	0.0	0.0
3574	2021-04-15	127.470001	127.489998	126.919998	127.489998	1300	0.0	0.0
3575	2021-04-16	128.000000	128.429993	127.599998	128.130005	2000	0.0	0.0
3576	2021-04-19	126.570000	126.720001	126.099998	126.720001	1600	0.0	0.0
3577	2021-04-20	126.059998	126.959999	123.839996	124.690002	1000	0.0	0.0

3578 rows × 8 columns

2.) Determine minima and maxima of every stock prices for various industries.



3.) Detect double top stock chart pattern..

	increment	ema	window	f1	f2	f3	f4	f5	fw_ret_1	fw_ret_2	fw_ret_3	label	industry	date
0	1	3	3	7.121051	7.353892	7.101648	7.353892	7.101648	0.017759	0.027322	0.028689	2	Foreign Regional Banks	(1675, 1689)
1	1	3	10	14.860000	16.240000	14.870000	16.260000	14.900000	0.019463	0.016779	0.048993	3	Gold	(4045, 4067)
2	1	3	3	2.744415	2.810281	2.744415	2.810281	2.744415	0.024000	0.000000	0.024000	1	Money Center Banks	(145, 151)
3	1	3	3	7.713200	7.817133	7.720625	7.787437	7.750320	0.005747	0.000000	0.002874	1	Money Center Banks	(3532, 3545)
4	1	3	3	7.141579	7.186120	7.141579	7.163850	7.126730	0.005209	0.002084	0.004167	3	Money Center Banks	(3628, 3644)
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1437	1	3	3	0.149173	0.149966	0.149173	0.149966	0.149173	0.005319	0.005319	0.005319	3	Rubber & Plastics	(4257, 4263)
1438	1	3	3	0.210344	0.215019	0.210344	0.215019	0.210344	0.022222	0.000000	0.022222	3	Rubber & Plastics	(4788, 4796)
1439	1	10	3	0.062426	0.065054	0.062426	0.065054	0.062426	0.000000	0.000000	0.021052	3	Rubber & Plastics	(3477, 3496)
1440	1	3	3	0.949174	0.963081	0.949174	0.966558	0.949174	-0.010990	-0.010990	0.000000	1	Foreign Regional Banks	(219, 228)
1441	1	3	3	292.901337	303.040253	294.027863	301.913696	294.027863	0.022988	0.057471	0.045977	3	Information Technology Services	(4027, 4039)

1442 rows × 14 columns

Data Description

The dataset use in this project was extracted from the scraped historical dataset of stock prices previously discussed. In particular closing price, industry, date/indexes are utilize to create the dataset for this project.

It now contains a target feature (label) that have 3 levels, decrease "0", neutral "1", increase "2" and 7 descriptive features, in which 5 of it (f1, f2, f3, f4, f5) consist of minima and maxima that forms a double top stock chart pattern, 1 is the date or indexes for start and end of the pattern lastly, industry of the stock where the pattern occur.

Data exploration

Created 3 additional descriptive features which are the absolute differences between f1 and f3, f3 and f5, and took the average of the value from f1 to f5. This will help increase the accuracy of the predictive model later on.

Model Selection

The dataset we have consists of quantitative features and a single categorical feature which is the target feature. The target feature contains multiple levels. Therefore, we will fit several models that are multiclass to our dataset, in particular multiclass KNN, multinomial logistic regression, multiclass SVM etc. to find the best predictive model of this project. Fortunately, we can fit these models to our dataset simultaneously using pycaret machine learning library.

Moreover, pycaret also automatically normalized then splits dataset if specified, do one hat encoding for nominal features, perform cross validation, tuned hyperparameter for every model etc.. In our case, we specify the normalization method as minmax scaler then split it to 80% training set and 20% testing set, then let it perform 5 fold cross validation with 10 repetition.

```
best_model = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.5616	0.6750	0.4883	0.5182	0.5217	0.2450	0.2592	0.7182
ada	Ada Boost Classifier	0.5579	0.6360	0.4842	0.5210	0.5178	0.2389	0.2536	0.1242
rf	Random Forest Classifier	0.5574	0.6689	0.4841	0.5151	0.5199	0.2395	0.2522	0.6804
lightgbm	Light Gradient Boosting Machine	0.5220	0.6446	0.4681	0.5016	0.5061	0.2045	0.2080	0.1574
et	Extra Trees Classifier	0.5152	0.6353	0.4518	0.4906	0.4949	0.1839	0.1887	0.7194
lr	Logistic Regression	0.5036	0.5758	0.3998	0.4512	0.4465	0.1133	0.1290	0.0466
ridge	Ridge Classifier	0.4930	0.0000	0.3990	0.4453	0.4455	0.1088	0.1200	0.0146
dt	Decision Tree Classifier	0.4889	0.5860	0.4511	0.4877	0.4860	0.1733	0.1744	0.0206
svm	SVM - Linear Kernel	0.4840	0.0000	0.3961	0.4364	0.4360	0.1039	0.1151	0.1336
lda	Linear Discriminant Analysis	0.4769	0.5756	0.4006	0.4436	0.4493	0.1088	0.1134	0.0336
knn	K Neighbors Classifier	0.4214	0.5481	0.3817	0.4241	0.4201	0.0664	0.0670	0.0750
qda	Quadratic Discriminant Analysis	0.2842	0.5155	0.3519	0.4228	0.2230	0.0206	0.0291	0.0236
nb	Naive Bayes	0.2797	0.5451	0.3600	0.4343	0.2139	0.0282	0.0383	0.0138

Tuned hyperparameters

```
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.1, loss='deviance', max_depth=3,
                           max_features=None, max_leaf_nodes=None,
                           min_impurity_decrease=0.0, min_impurity_split=None,
                           min_samples_leaf=1, min_samples_split=2,
                           min_weight_fraction_leaf=0.0, n_estimators=100,
                           n_iter_no_change=None, presort='deprecated',
                           random_state=13, subsample=1.0, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)
```

Model performance evaluation

```
test_scores = predict_model(best_model)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Gradient Boosting Classifier	0.5986	0.6689	0.4995	0.5018	0.5260	0.2837	0.3187