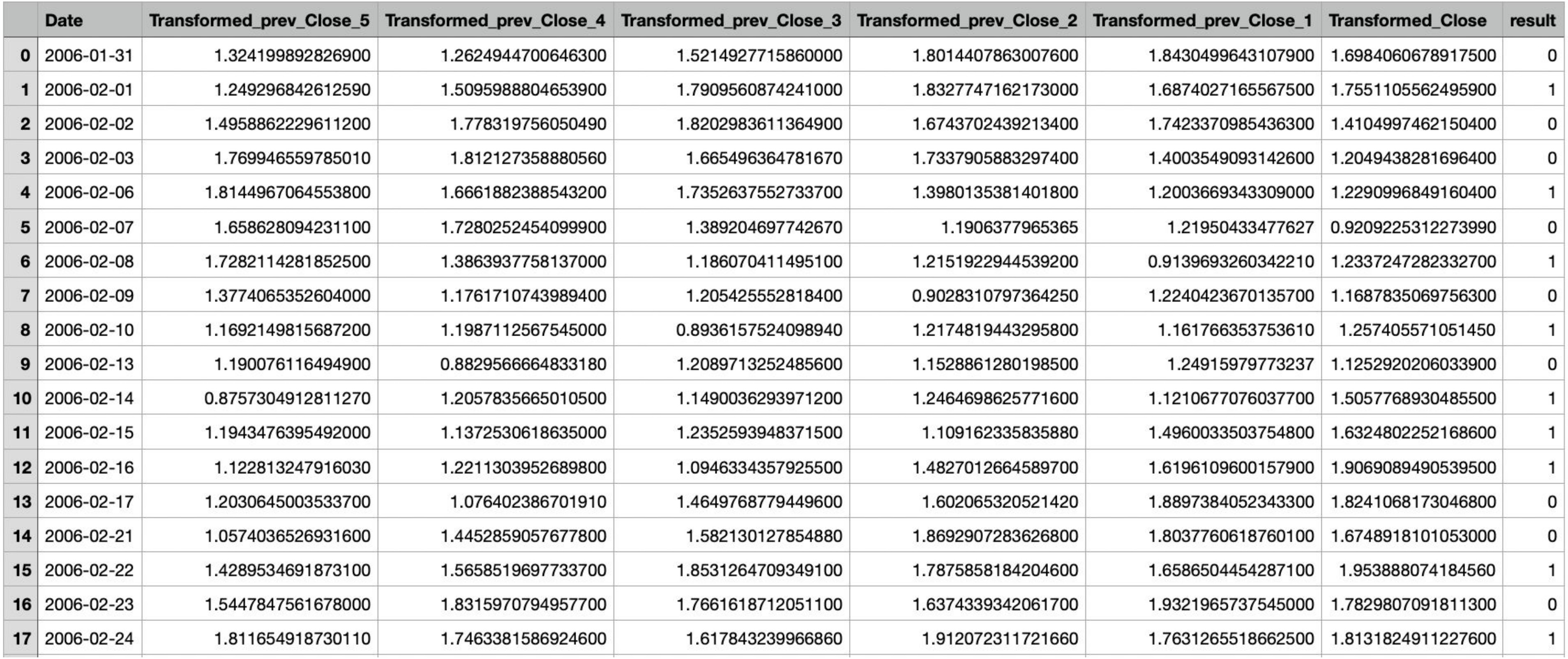
Predicting the Stock Market

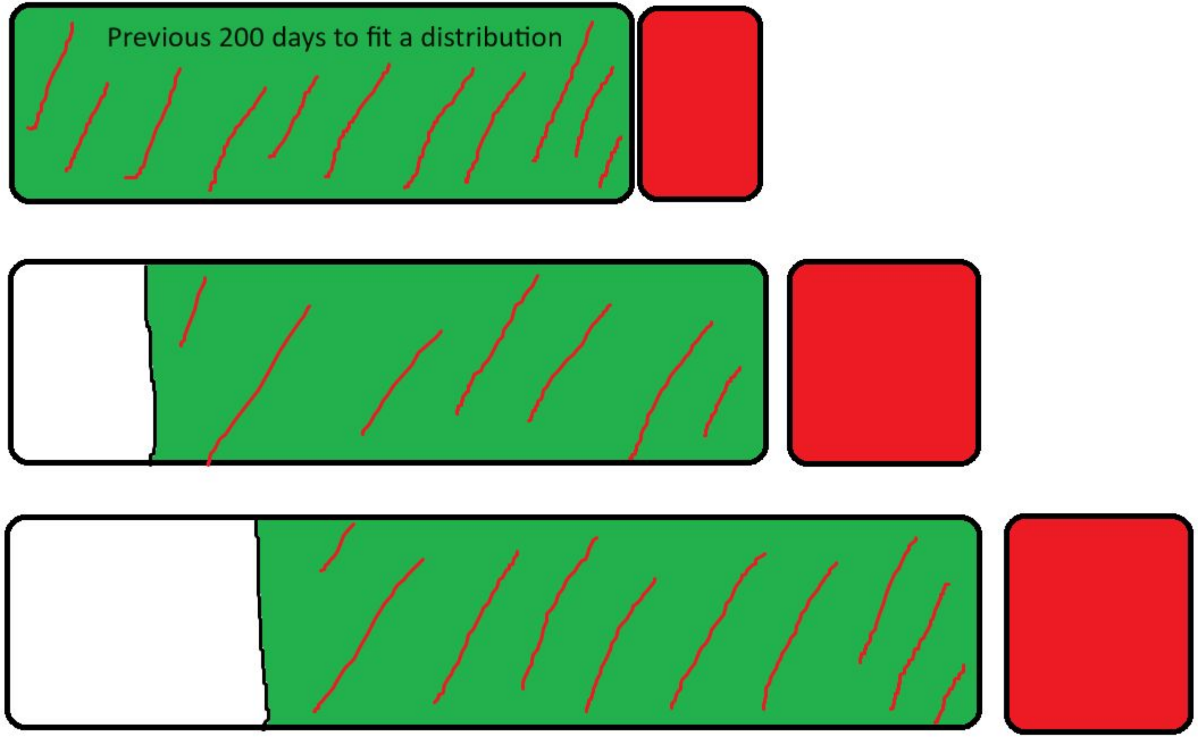
Yiling Dai, Junjie Li, Pawin Linmaneechote, Michael Raffanti, Qingyang Wei

Dataset



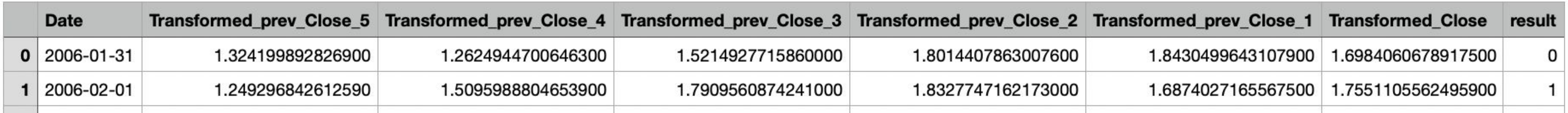
	Date	Transformed_prev_Close_5	Transformed_prev_Close_4	Transformed_prev_Close_3	Transformed_prev_Close_2	Transformed_prev_Close_1	Transformed_Close	result
0	2006-01-31	1.324199892826900	1.2624944700646300	1.5214927715860000	1.8014407863007600	1.8430499643107900	1.6984060678917500	0
1	2006-02-01	1.249296842612590	1.5095988804653900	1.7909560874241000	1.8327747162173000	1.6874027165567500	1.7551105562495900	1
2	2006-02-02	1.4958862229611200	1.778319756050490	1.8202983611364900	1.6743702439213400	1.7423370985436300	1.4104997462150400	0
3	2006-02-03	1.769946559785010	1.812127358880560	1.665496364781670	1.7337905883297400	1.4003549093142600	1.2049438281696400	0
4	2006-02-06	1.8144967064553800	1.6661882388543200	1.7352637552733700	1.3980135381401800	1.2003669343309000	1.2290996849160400	1
5	2006-02-07	1.658628094231100	1.7280252454099900	1.389204697742670	1.1906377965365	1.21950433477627	0.9209225312273990	0
6	2006-02-08	1.7282114281852500	1.3863937758137000	1.186070411495100	1.2151922944539200	0.9139693260342210	1.2337247282332700	1
7	2006-02-09	1.3774065352604000	1.1761710743989400	1.205425552818400	0.9028310797364250	1.2240423670135700	1.1687835069756300	0
8	2006-02-10	1.1692149815687200	1.1987112567545000	0.8936157524098940	1.2174819443295800	1.161766353753610	1.257405571051450	1
9	2006-02-13	1.190076116494900	0.8829566664833180	1.2089713252485600	1.1528861280198500	1.24915979773237	1.1252920206033900	0
10	2006-02-14	0.8757304912811270	1.2057835665010500	1.1490036293971200	1.2464698625771600	1.1210677076037700	1.5057768930485500	1
11	2006-02-15	1.1943476395492000	1.1372530618635000	1.2352593948371500	1.109162335835880	1.4960033503754800	1.6324802252168600	1
12	2006-02-16	1.122813247916030	1.2211303952689800	1.0946334357925500	1.4827012664589700	1.6196109600157900	1.9069089490539500	1
13	2006-02-17	1.2030645003533700	1.076402386701910	1.4649768779449600	1.602065320521420	1.8897384052343300	1.8241068173046800	0
14	2006-02-21	1.0574036526931600	1.4452859057677800	1.582130127854880	1.8692907283626800	1.8037760618760100	1.6748918101053000	0
15	2006-02-22	1.4289534691873100	1.5658519697733700	1.8531264709349100	1.7875858184204600	1.6586504454287100	1.953888074184560	1
16	2006-02-23	1.5447847561678000	1.8315970794957700	1.7661618712051100	1.6374339342061700	1.9321965737545000	1.7829807091811300	0
17	2006-02-24	1.811654918730110	1.7463381586924600	1.617843239966860	1.912072311721660	1.7631265518662500	1.8131824911227600	1

Feature Scaling



Methods

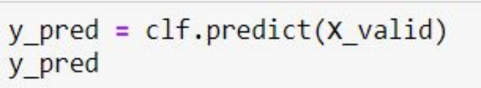
Model	X	y
SVC	previous 5 days prices	Result
Logistic regression	previous 5 days prices	Result
SVR	previous 5 days prices	Transformed_Close
kNN regression	previous 5 days prices	Transformed_Close
Linear regression	previous 5 days prices	Transformed_Close
Decision Tree regression	previous 5 days prices	Transformed_Close



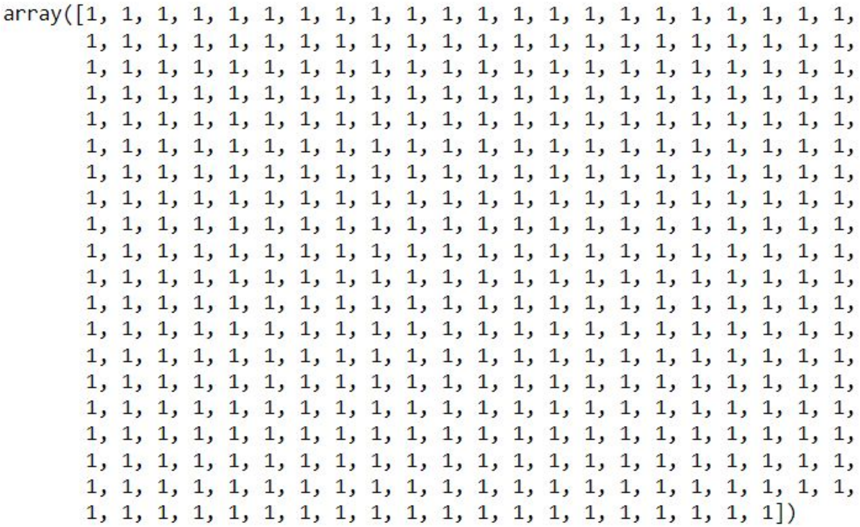
	Date	Transformed_prev_Close_5	Transformed_prev_Close_4	Transformed_prev_Close_3	Transformed_prev_Close_2	Transformed_prev_Close_1	Transformed_Close	result
0	2006-01-31	1.324199892826900	1.262494470064630	1.521492771586000	1.801440786300760	1.843049964310790	1.698406067891750	0
1	2006-02-01	1.249296842612590	1.509598880465390	1.790956087424100	1.832774716217300	1.687402716556750	1.755110556249590	1

Classification for Binary

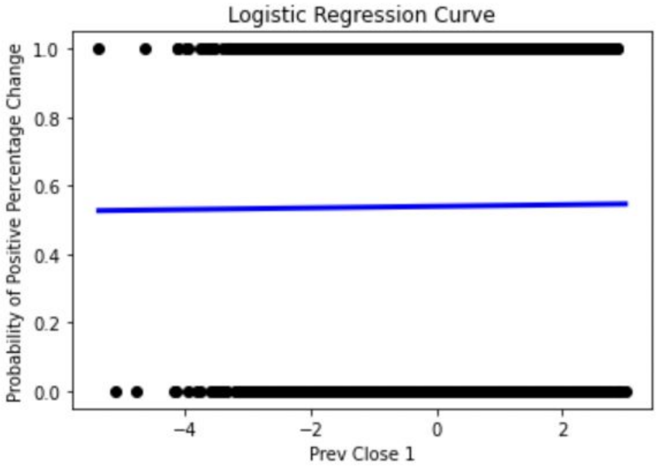
SVC and Logistic model are inappropriate:

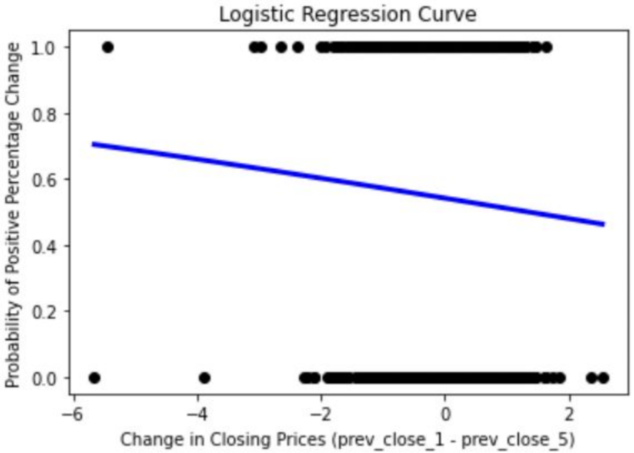


```
y_pred = clf.predict(X_valid)
y_pred
```



```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

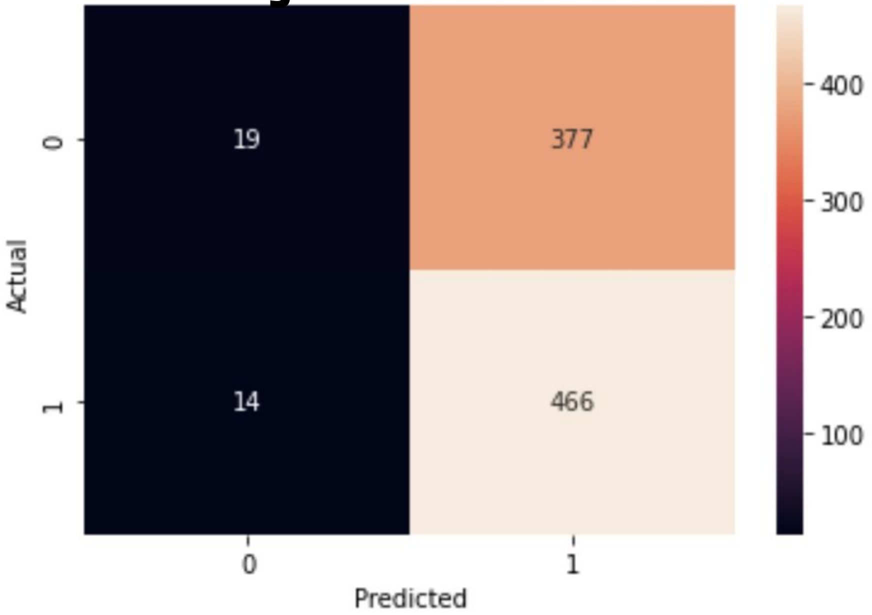




Confusion Matrix - Logistic Regression

0 - Represents a negative % changes across the previous closing days

True Negatives **False Positives**

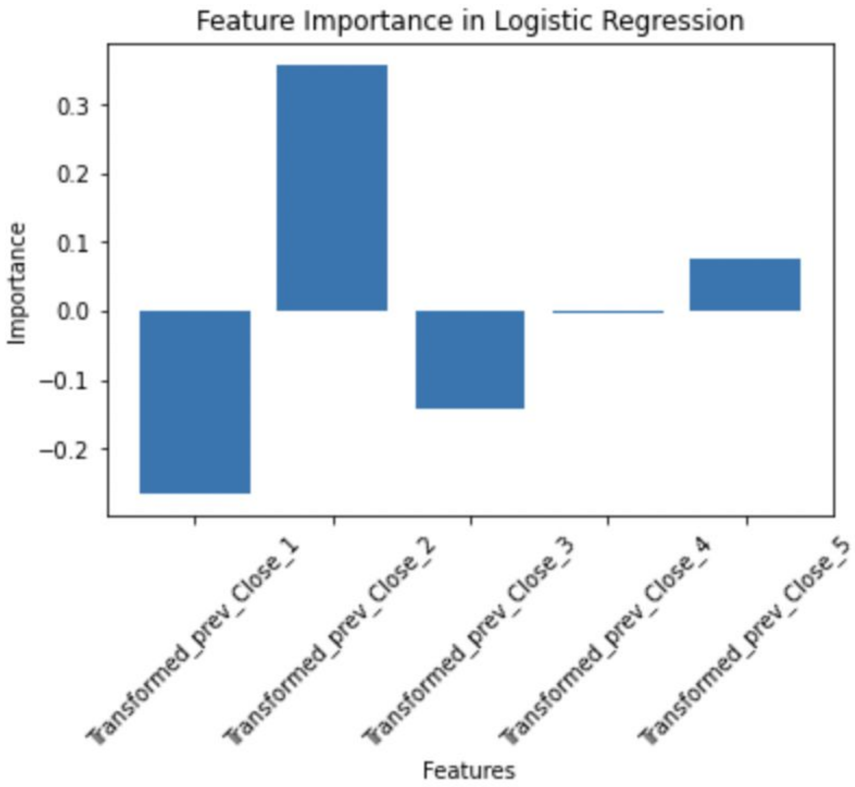


False Negatives **True Positives**

1 - Represents a positive % changes across the previous closing days

The logistic regression model may have a bias towards predicting one class more frequently, which can be due to the model's inherent biases, the way it's been trained, or the features it's using

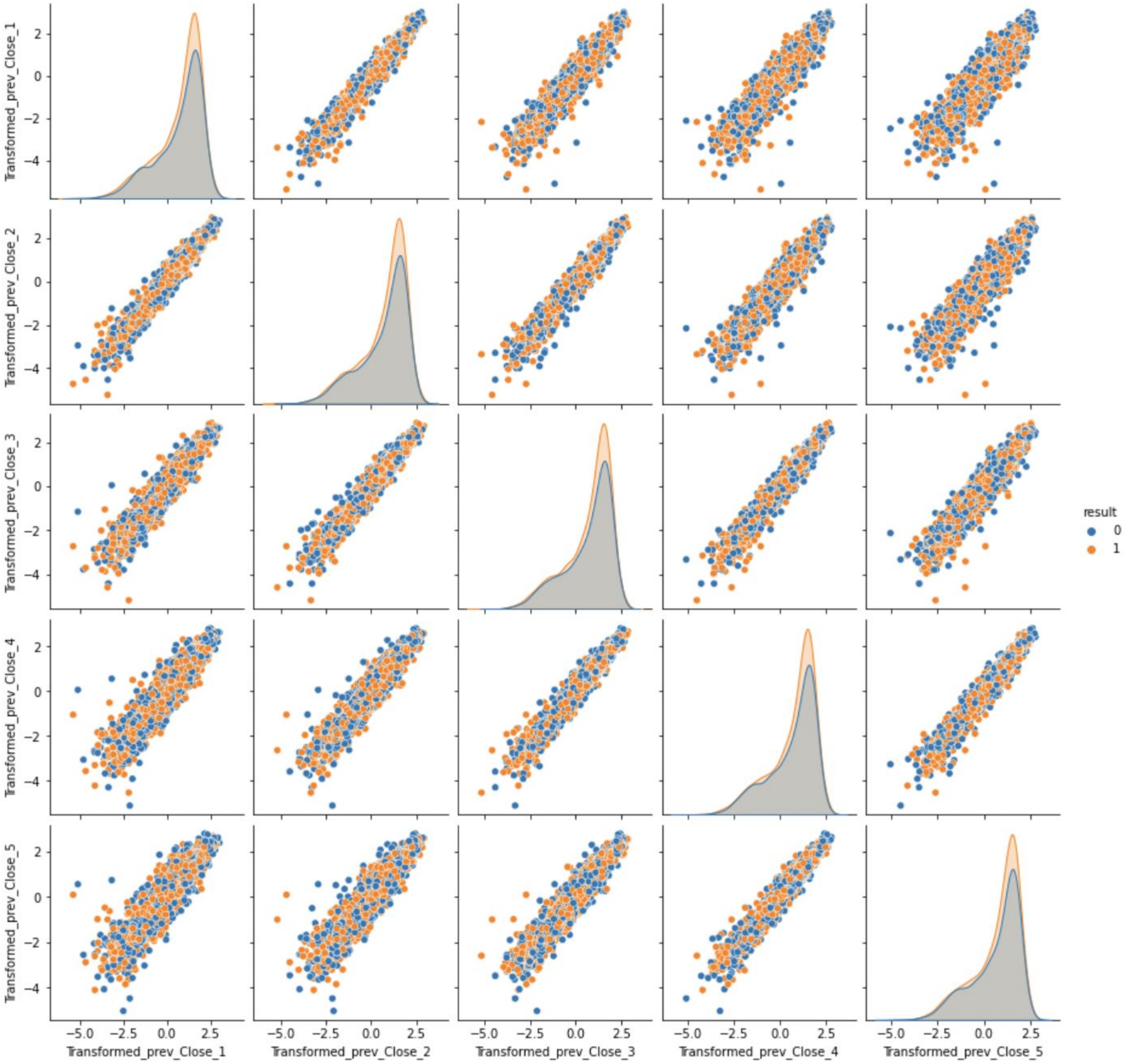
Feature Importance - Logistic Regression



- Bars above the zero line suggest that an increase in the corresponding feature's value is associated with an increase in the probability of the target variable being 1.
- Bars below the zero line (negative values) suggest that an increase in the corresponding feature's value is associated with a decrease in the probability of the target variable being 1.

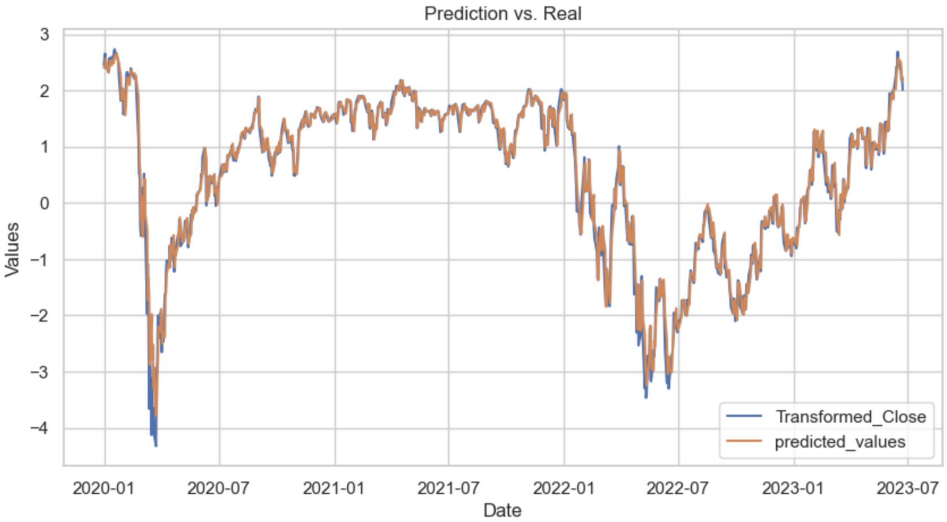
Pairplot - Logistic Regression

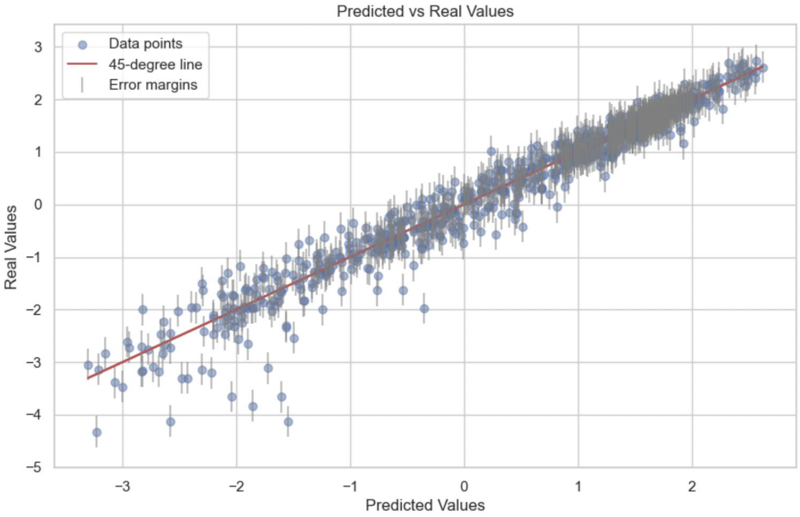
- The features have a positive linear relationship with each other
- These features do not strongly distinguish between days with a positive and negative percentage change in stock price
- The distribution plots suggest that the transformed previous close values themselves do not differ drastically



Predicting df[‘Transformed_Close’]

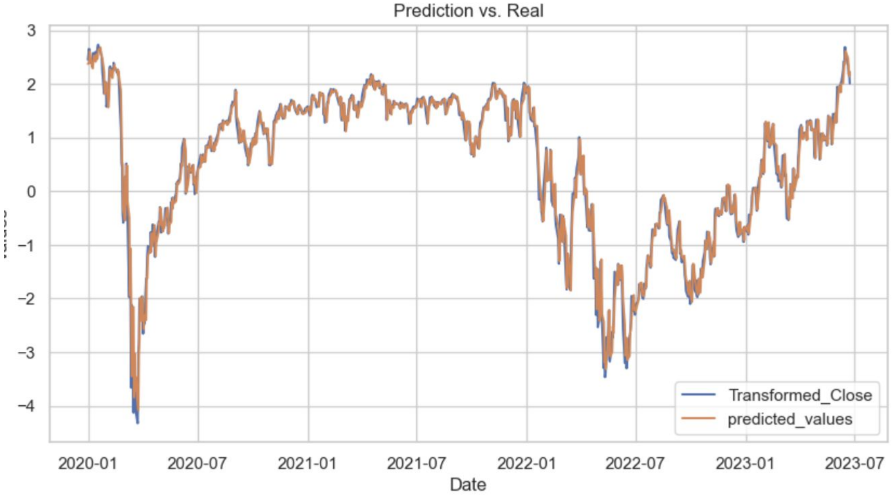
Based on the prices from previous 5 days, using SVR

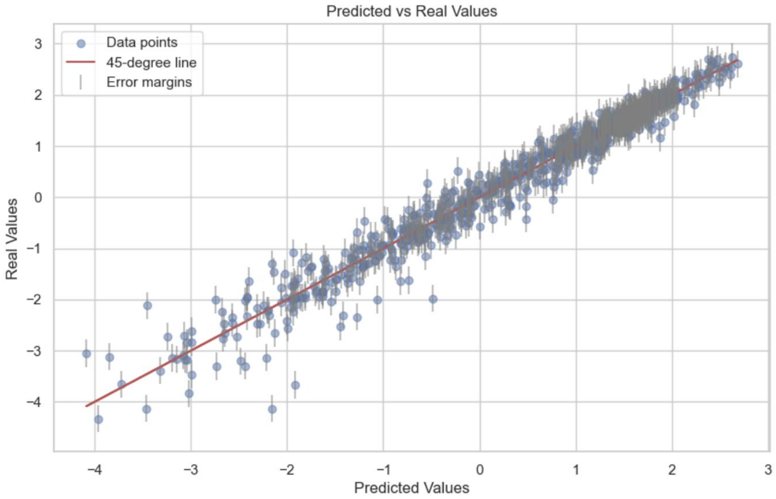




Predicting df[‘Transformed_Close’]

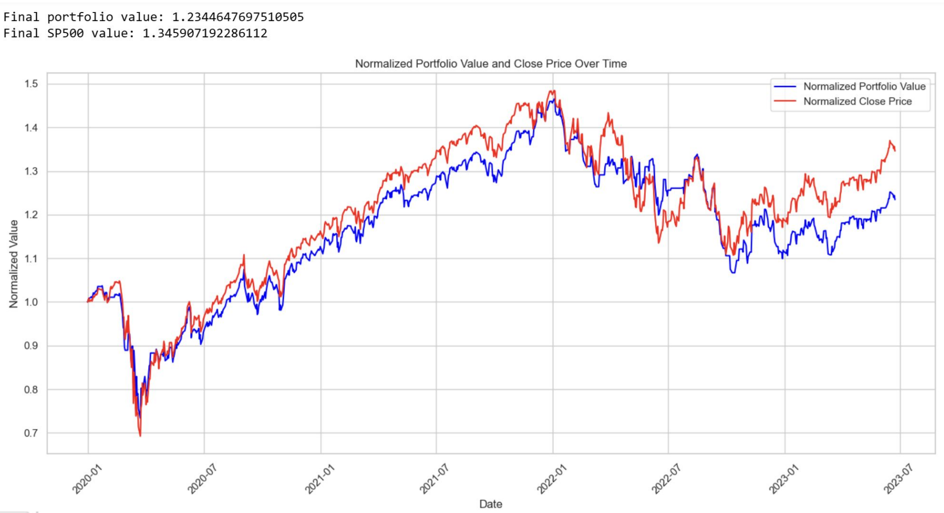
Based on the prices from previous 5 days, using Linear



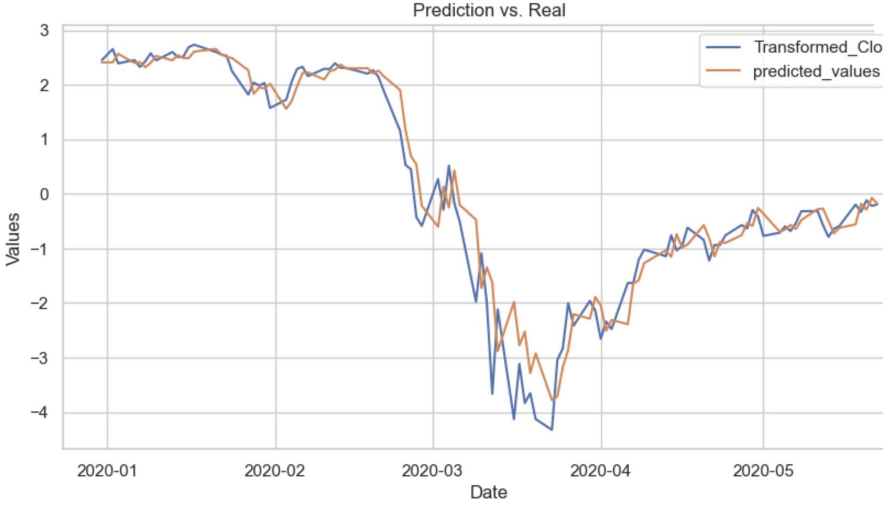


Getting Rich?

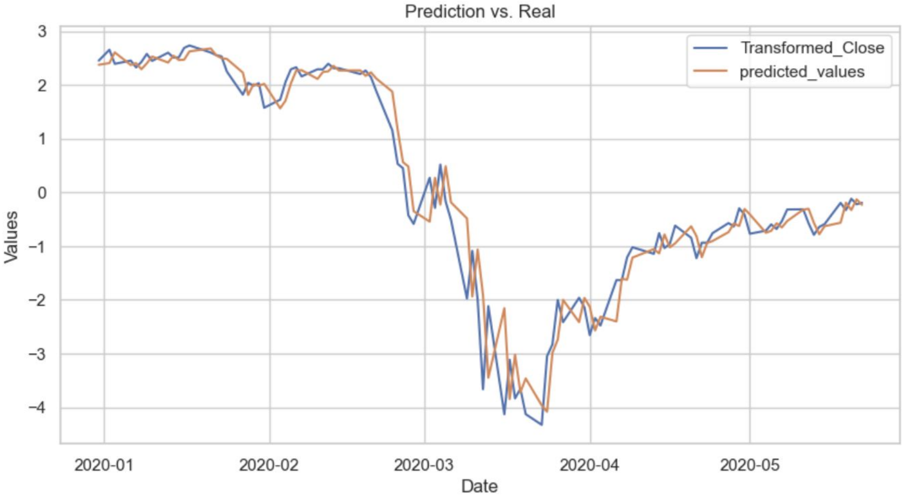
A closer look at SVR…



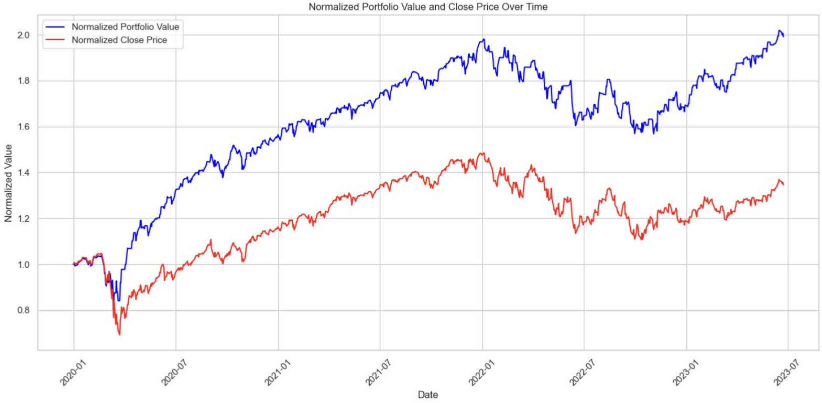
```
Final portfolio value: 1.2344647697510505
Final SP500 value: 1.345907192286112
```



OG Regression



```
Final portfolio value: 1.9907027253730356
Final SP500 value: 1.345907192286112
```



Predicting df[‘Transformed_Close’]

Based on the price from previous 5 days, we can also use Decision Tree regression and kNN regression to predict closing price for today.

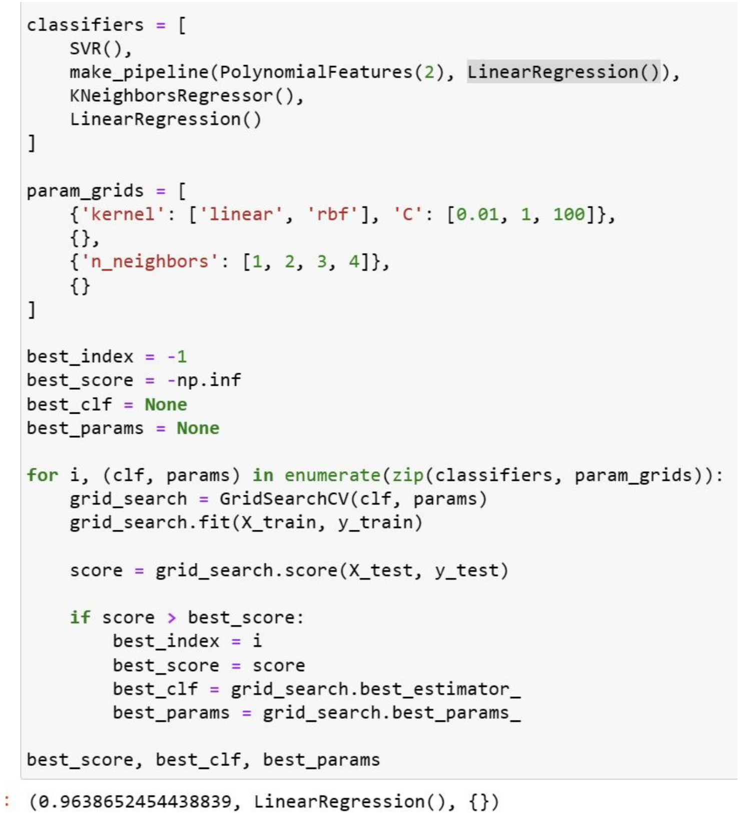
Decision Tree Regression: 0.9286

KNN: 0.9486

Model Comparison

Accuracy:

- Logistic Regression: 0.55
- Decision Tree Regression: 0.9286
- KNN: 0.9486
- SVR: 0.9637
- Linear Regression: 0.9639



```
classifiers = [
    SVR(),
    make_pipeline(PolynomialFeatures(2), LinearRegression()),
    KNeighborsRegressor(),
    LinearRegression()
]

param_grids = [
    {'kernel': ['linear', 'rbf'], 'C': [0.01, 1, 100]},
    {},
    {'n_neighbors': [1, 2, 3, 4]},
    {}
]

best_index = -1
best_score = -np.inf
best_clf = None
best_params = None

for i, (clf, params) in enumerate(zip(classifiers, param_grids)):
    grid_search = GridSearchCV(clf, params)
    grid_search.fit(X_train, y_train)

    score = grid_search.score(X_test, y_test)

    if score > best_score:
        best_index = i
        best_score = score
        best_clf = grid_search.best_estimator_
        best_params = grid_search.best_params_

best_score, best_clf, best_params
```

```
: (0.9638652454438839, LinearRegression(), {})
```

Q&A?