Predicting Inflation in the United States using the S&P 500 Price

CS210 Project Step III

Nuh Al-Sharafi, 32702

22.5.2024

Introduction:

In the report that preceded this (Al-Sharafi, 2024), a discussion regarding the feasibility of predicting the inflation rate in the United States using current market movements took place. In that discussion, it was argued that the S&P 500 is a very good estimator for market movements, then, it was used to predict inflation. It was concluded that current market movements most accurately model inflation nine months ahead. Armed with this information, a linear regression model was constructed to predict normalized inflation using normalized S&P 500 prices.

```
Inflation_rate = (0.89 * S&P_500 price) - 0.01
```

This model performed well giving an R^2 of *0.86* and RMSE *0.163.* That raised the question of whether a machine learning model could predict inflation with the given data more accurately than linear regression, after all, linear regression is nothing more than a line.

The aim of this paper is to answer that question. Firstly, the models that will be used have to be chosen. In machine learning, there's supervised and unsupervised models. For the purposes of this paper, supervised models will be used because there's a label to predict. Within supervised models, there are models that do categorization and models that do regression, and since the target here is regression, categorization models will also be crossed out. That leaves KNN (K Nearest Neighbors), Decision Tree and Random Forest. Since Random Forest is just a group of decision trees, decision trees can be crossed out. Additionally, Random Forest reduces bias and overfitting compared to Decision Tree because of the multiplicity of trees. So, analysis will be conducted on KNN and Random Forest.

KNN:

Setup:

Since KNN needs labeled training data, the inflation rate was provided, and the feature that was chosen to predict inflation is S&P 500, as per previous results. Then, based on previous analysis, to predict inflation most accurately, the database was shifted by 9-months. In addition to the target label and features, KNN also has a hyperparameter *K* which determines the number of nearest neighbors, and a parameter *metric* which determines the distance algorithm.

Optimization:

In optimizing the variables, the *metric* was left as default, which uses the Euclidean distance (*KNeighborsRegressor*, n.d.). Euclidean distance is perfect for this case since it operates best on numeric values.

As for K, the model was trained 15 different times with 15 different K values ranging from 1 to 15. In every run, the RMSE and R2 scores were recorded based on a 90-10 train-test split. Then the top performing K value was selected.

Verification:

To verify the robustness and generalizability of the model, 5-fold cross validation was performed and the variance of the RMSE and R2 scores was taken into consideration.

Results:

Firstly, the most optimal K-value came out to be 2 with

RMSE: `0.647109`

R2: `0.830846`

The performance metrics dropped after K=2 but recovered around K=9

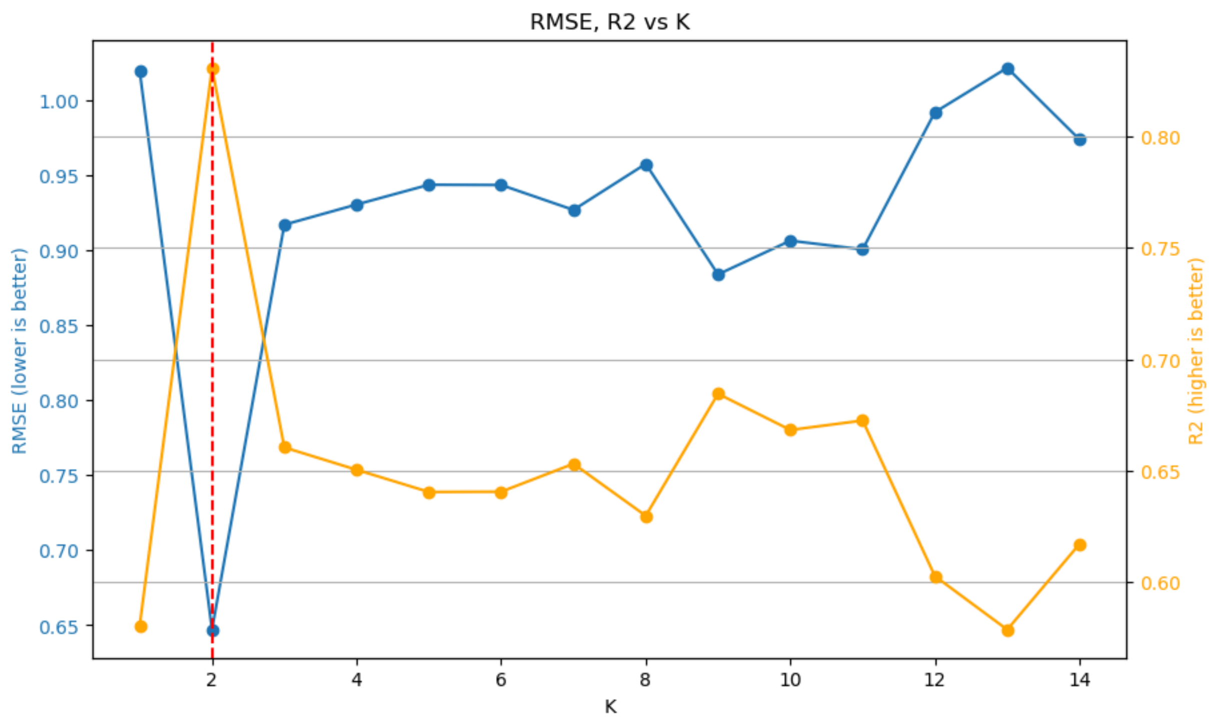

Figure 1, RMSE, R2 vs K

As for cross-validation,

```
      Cross Validation Scores (RMSE):
[0.71309186 0.47258156 0.79773986 0.52307849 0.90997558]
      Cross Validation Scores (R2):
[0.86497969 0.93684982 0.77199443 0.87227697 0.70507211]

      Average RMSE: 0.83211553052273
      Average R2:   0.7697666581563214
      --------------
      Variance RMSE: 0.006853057584077315
      Variance R2:   0.005792427431412683 5
```

The RMSE is in an acceptable range with acceptable variance. As for the R2 score, the mean is low, but variance is ok.

All in all, KNN is slightly more accurate than linear regression but fails to compete in terms of variance. Nonetheless, KNN can still be employed as a function to predict inflation. The loss to linear regression might be due to KNNs inherent weakness when it comes to extrapolation.

Random Forest:

Setup:

For Random Forest, multiple features are required, preferably independent. That is why the scope of analysis was expanded to include S&P 500, Gold, Crude Oil, and Copper Prices along with Crude Oil and Berkshire trading volume rate. The additional features expand the horizon beyond S&P 500 without being components of it which makes them more independent than other choices. Since the volume rate features have very low correlation with inflation, the model will be constructed with and without them to test whether they have significant weight. The results will determine whether they get dropped or not.

On the other hand, the model has 2 main hyper-parameters to tune, namely *n_estimators* which specifies the number of decision trees and *max_depth* which specifies the depth of each tree. Other parameters will be kept as default because it's sufficient (*RandomForestRegressor*, n.d.).

Optimization:

The plausible choices for both parameters were compiled where None means there's no control on the *max_depth* of the decision tree.

```
max_depths = [None, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
n_estimators = [50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150]
```

Then, 121 different random forests were generated with all the parameter combinations, the RMSE and R2 scores were recorded and then used to find the most optimal combination using a 90-10 train-test split.

Verification:

Same validation procedure as KNN was performed.

Results:

Firstly, for the model with the volume features, the most optimal pair of parameters was *max_depth=1, n_estimators=140*

```
RMSE = 0.809837
R2 = 0.735075
```

As for the other model, the one without the volume features, the most optimal pair or parameters was *max_depth=1, n_estimators=110*

```
RMSE = 0.835378
R2 = 0.718101
```

After finding the most optimal pairs, cross validation was performed to test which of the two models performed better

```
Extra Features:
Mean RMSE: 0.8281954177143355
Mean R2: 0.7587030032095308
----------
RMSE Variance: 0.0074712428984354765
```

```
R2 Variance: 0.005358637290876207

Without Extra Features:
Mean RMSE: 0.824044364852643
Mean R2: 0.7627763224071359
----------
RMSE Variance: 0.0071400920546144065
R2 Variance: 0.0045579425430488495
```

By comparing the values above or looking at figures 2 and 3, it can be seen that the model without volume features performed slightly better than the one with volume features, but the difference is so minute it can be attributed to luck.

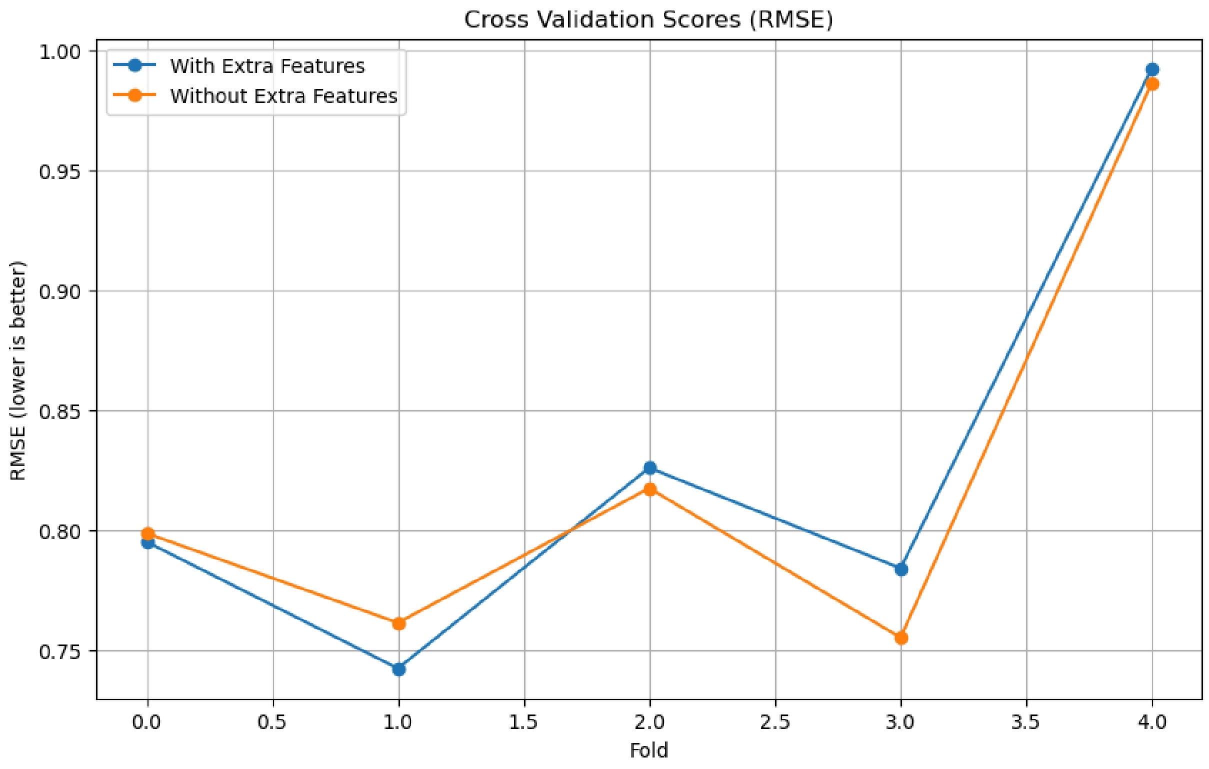

Figure 2, RMSE cross validation for model with vs. without volume features

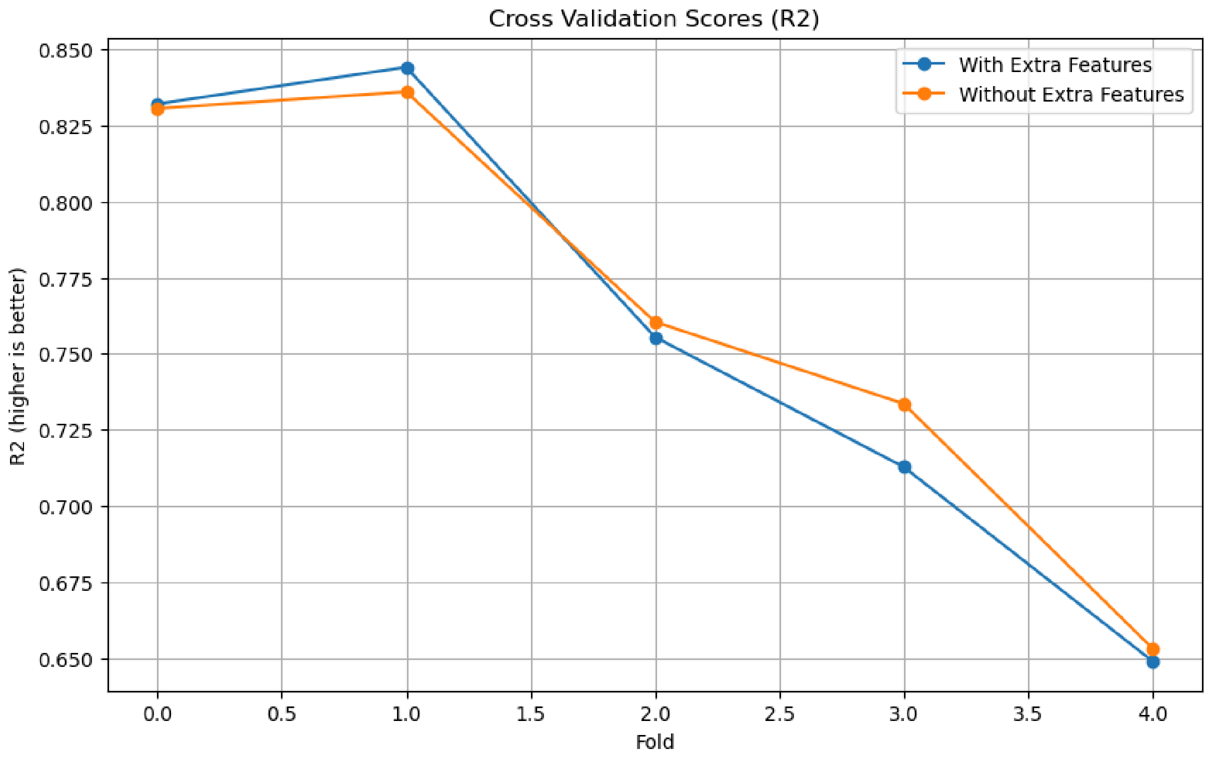

Figure 3, R2 cross validation for model with vs. without volume features

To sum up, both models' mean RMSE and R2 were acceptable making the model valid. However, when it comes to performance ranking both random forests performed slightly worse than KNN, which in itself performed slightly worse than linear regression.

Conclusion:

Finally, when it comes to regression and predicting parameters, there's no rule that says with complexity comes accuracy and that is demonstrated by this paper. Although KNN and Random Forest are much more complicated than linear regression, they failed to produce results as accurate as it. Although, for the sake of completeness, KNN and Random Forest did come extremely close to the linear regression model's performance. It might be possible for both machine learning models to surpass linear regression given more data to train the models. However, as things currently stand, both ML models had acceptable yet not optimal performance.

References:

Al-Sharafi, N. (2024). *Predicting inflation in the United States using S&P 500 Price, Part II* [Project Report]. Sabanci University.

KNeighborsRegressor. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

RandomForestRegressor. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html