Impact of Precipitation and Temperature on Hard Red Winter Wheat Yields in the United States

Level: 6000

Abstract

Hard red winter wheat is the most common variation of wheat grown in the United States. It's a versatile wheat which mostly used for bread, hard rolls, flat breads, all-purpose flour, and Asian style noodles. In this study, I investigate how precipitation and temperature influence hard red winter wheat yield. Leveraging different types of regression model to predict the annual wheat yield based on precipitation and temperature data from 2001 to 2021. The result shows that K-Nearest Neighbors Regression and Support Vector Regression with polynomial kernel achieve the best performance with RMSE value of 4.279 and 4.480 using leave one out cross validation.

Introduction

Precipitation and temperature are 2 important factors that influence wheat yield. Researchers (Pirttioja N, Carter TR, Fronzek S, Bindi M and others, 2015) explored the utility of the impact response surface approach for developing an ensemble model used for detecting the crop yield impact under a large range of changes in climate in Finland, Germany and Spain. Some researchers (Mukherjee A, Wang S-YS, Promchote P, 2019) focus on examining the climate factors that reduce wheat yield in Northwest India. They conclude that higher temperatures coupled with water shortage and irregular irrigation appear to result in wheat yield reduction.

Since hard red winter wheat is the most common wheat in the United States, wheat yield prediction model is needed in terms of monitoring the wheat productivity in the future.

Data Description and Exploratory Data Analytics

1. Dataset 1: Monthly Precipitation

   URL: https://disc.gsfc.nasa.gov/datasets/GPM_3IMERGM_06/

   This precipitation estimates from the various precipitation-relevant satellite passive microwave (PMW) sensors comprising the GPM constellation are computed using the 2017 version of the Goddard Profiling Algorithm (GPROF2017), then gridded, intercalibrated to the GPM Combined Ku Radar-Radiometer Algorithm (CORRA) product, and merged into half-hourly 0.1°x0.1° (roughly 10x10 km) fields.

   Format: HDF5

   Spatial Coverage: -180.0, -90.0, 180.0, 90.0

   Temporal Coverage: 2000-06-01 to 2021-09-30

   File Size: 30 MB per file

   Data Resolution Spatial: 0.1 ° x 0.1 °

   Temporal: 1 month

2. Dataset 2: Daily Temperature

   URL: https://downloads.psl.noaa.gov/Datasets/cpc_global_temp/

   Daily maximum temperature and daily minimum temperature are collected by CPC archive of Global Telecommunication System (GTS) daily report. Reports available from 6000~7000 global stations. The researchers are encouraged to calculate daily average temperature by (tmax + tmin)/2.

   Format: netCDF

   Spatial Coverage: 0.0, -90.0, 360.0, 90.0

   Temporal Coverage: 1979-01-01 to present

   File Size: 89 MB per file

   Data Resolution Spatial: 0.5 ° x 0.5 °

   Temporal: 1 day

3. Dataset 3: Wheat Yield

   URL: https://www.ers.usda.gov/webdocs/DataFiles/54282/Wheat%20Data-All%20Years.xlsx?v=3423

   This data product contains statistics on wheat—including the five classes of wheat: hard red winter, hard red spring, soft red winter, white, and durum—and rye. This product includes data published in the monthly Wheat Outlook and monthly updates

to the Wheat Yearbook tables. Data are monthly, quarterly, and/or annual, depending upon the data series. Most data are shown as marketing years, others as calendar years. Also provided on this page is the Wheat By-Class Quarterly data, which are generally updated each quarter.

Table 1--Wheat: planted acreage, harvested acreage, production, yield, and farm price

Class and marketing year 1/		Planted acreage million acres	Harvested acreage million acres	Production million bushels	Yield bushels per acre	Weighted-average farm price 2/ dollars per bushel
All wheat	1866/67	--	15.408	170	11.0	2.06
	1867/68	--	16.738	211	12.6	2.01
	1868/69	--	19.140	246	12.9	1.46
	1869/70	--	21.194	290	13.7	.92
	1870/71	--	20.945	254	12.1	1.04
	1871/72	--	22.230	272	12.2	1.25
	1872/73	--	22.962	271	11.8	1.24
	1873/74	--	24.866	322	12.9	1.17
	1874/75	--	27.310	356	13.0	.95
	1875/76	--	28.382	314	11.1	1.01

Figure 1: Overview of Wheat Yield Dataset

4. Thoughts

At the beginning of the project, I was only considering using precipitation data and wheat yield data. However, precipitation is the not the only factor that have huge impact on wheat yield. Then I did some research on Internet and found that temperature also have profound influence for crop growth. So, I get the most detailed data that I can find on the internet from Climate Prediction Center (CPC) allowing me to take more variables into consideration.

Data Preprocessing (Analysis)

1. Select Areas

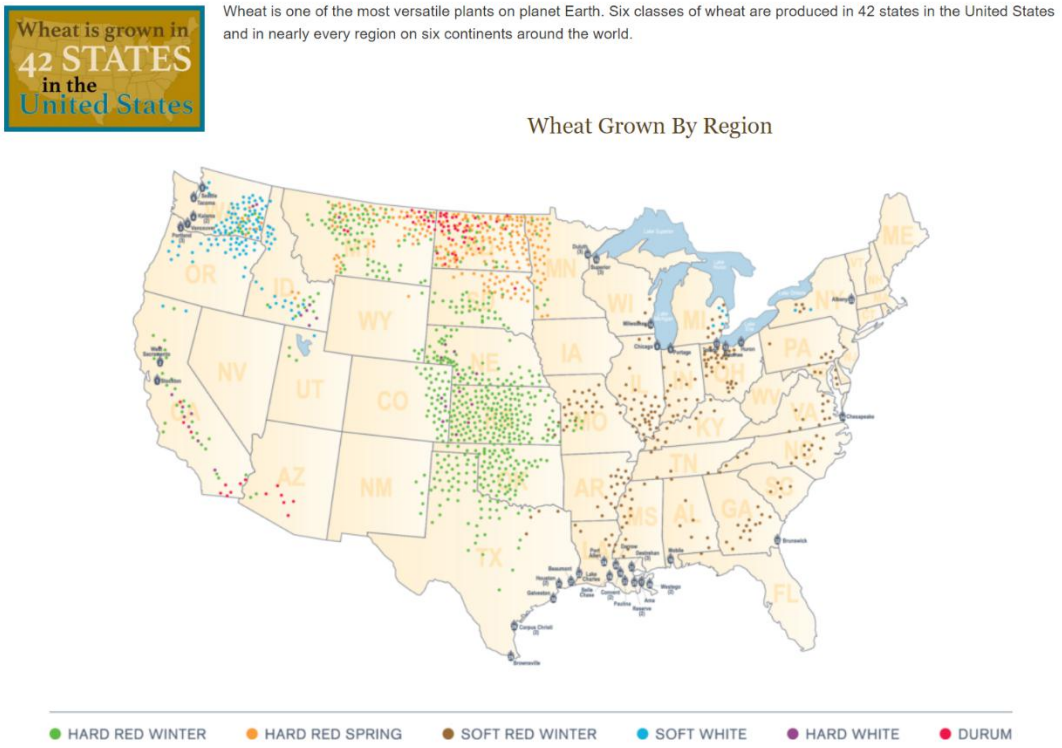

Figure 2: Wheat Map

This study focuses on hard red winter wheat, so I decide to select 4 areas in grid that best include all the green point on the wheat map. the 4 areas are: longitude [-104.05405, -94.61825], latitude [36.99900, 41.00132] in Kansas State, longitude [-103.01280, -97.28346], latitude [33.89811, 36.99900] in Oklahoma State, longitude [-112.05760, -104.05405], latitude [44.99552, 48.99742] in Montana State, longitude [-104.05982, -102.06658], latitude [41.00132, 43.00957] in Nebraska State.

2. Process Raw Precipitation Data

Using h5py package to read HDF5 file. Creating longitude/latitude list and fill it with len(latitude)/len(longitude) repeated original list. Then utilizing flatten function from numpy package to transform the precipitation value. This step is to convert 3-dimension data to 2-dimension data. After conversion, I replace noise with 0 using mask function. Finally, I draw 4 grids as I mention in select areas section and gather the precipitation value only from these 4 grids and get the sum value which is the ultimate monthly precipitation value that I want. To create code automation

processing all precipitation file, I use global function to obtain all file url and leverage for loop to achieve automate precipitation data processing.

3. Process Raw Temperature Data

The overall procedure is similar to the step I process raw precipitation data. The different part is using netCDF4 package to read file. Using average value instead of sum. Using num2date function to transform the date column. Calculate average value by the mean of tmax plus tmin and finally group daily data to monthly data.

4. Dataset Merging and Transformation

I merge the precipitation, temperature and wheat yield dataset by year number using concat function. Since the winter type wheat are seeded by fall and then move to dormancy through winter and harvested at late May or June based on different area. I decide only use data from January to May. By using pivot function to munge the data, the dataset format is extremely close to the data can be used for model. The independent variable is monthly precipitation and temperature from January to May each year. The target variable is wheat yield, unit of it is bushels per acre.

5. Analysis

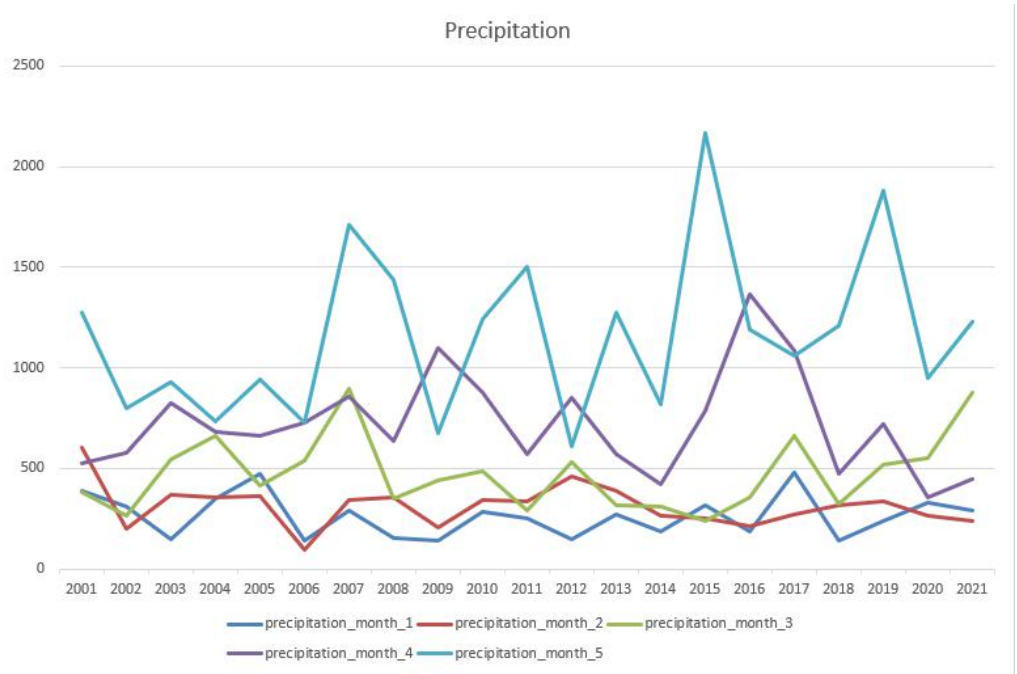

Figure 3: Precipitation Time Series Analysis

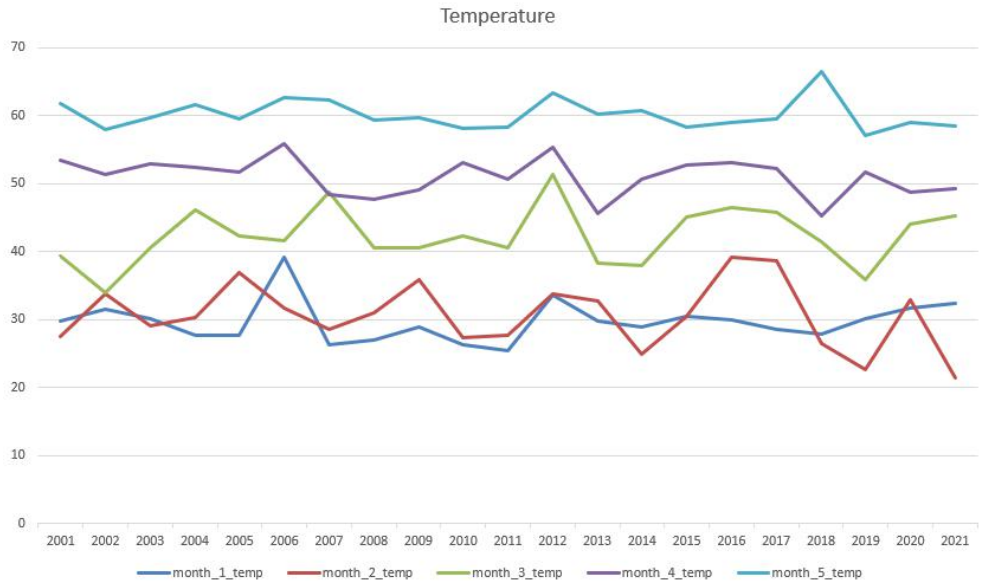

Figure 4: Precipitation Time Series Analysis

In order to explore the trend of precipitation and temperature, I conduct time series analysis using monthly data from 2001 to 2021 and January to May each year. As shown from figure 3, the precipitation value remains the same until February and start to increase from March to May. As time move on to March, April and May, precipitation become more unstable when we compare precipitation for each month. It appears that the average value among each year is more stable, there is no obvious spike or significant drop.

6. Data Standardization

Obviously, precipitation and temperature are not the same order of magnitude. The precipitation value is way greater than temperature value. So, I utilize StandScaler function to transform data making sure each variable takes the same weight.

Year	precipitation_month_1	precipitation_month_2	precipitation_month_3	precipitation_month_4	precipitation_month_5	month_1_temp	month_2_temp	month_3_temp	month_4_temp	month_5_temp	yield
2001	385.3383484	604.3039551	381.2904053	525.6271973	1274.29541	29.725	27.525	39.35	53.4	61.8	36.72491
2002	312.6609192	199.784668	262.1911316	574.1824951	797.0206299	31.5	33.825	33.975	51.325	58.025	31.12846
2003	145.6664124	371.8045349	545.852356	827.8270874	930.5014648	30.15	29	40.6	52.9	59.775	41.79469
2004	348.4841919	352.9444885	662.8456421	682.885376	733.6508789	27.625	30.325	46.225	52.4	61.55	36.59688
2005	471.7041016	362.3030396	413.5306091	663.098999	943.59198	27.7	36.95	42.25	51.75	59.6	37.76711
2006	138.0708771	97.94351196	540.0499268	727.3969727	726.5213013	39.15	31.75	41.575	55.925	62.625	31.98654
2007	289.0488892	342.5772705	895.8411255	859.0713501	1708.040527	26.375	28.6	48.775	48.425	62.25	37.15511
2008	153.7771149	354.473938	350.6074829	635.7884521	1439.599243	26.925	30.975	40.65	47.625	59.35	40.0039
2009	140.1841125	207.9692993	437.2670898	1096.2854	674.1495972	28.875	35.95	40.55	49.075	59.65	38.10186
2010	286.5023804	340.3811646	484.8961792	877.4642944	1242.444946	26.275	27.425	42.275	53.15	58.225	42.09007
2011	253.0499573	333.4958191	291.8121033	572.1126709	1504.691406	25.45	27.625	40.55	50.625	58.35	36.363
2012	149.4436188	462.4129639	532.2808228	848.5889893	608.6260376	33.575	33.775	51.325	55.45	63.35	40.57524
2013	271.6402588	386.1445313	318.8605042	570.9605103	1273.187988	29.8	32.725	38.275	45.625	60.175	36.64671
2014	188.7514038	263.8212891	311.5839539	422.9547119	816.34729	28.875	24.85	37.925	50.675	60.8	33.69293
2015	316.9320679	251.6775055	238.4590607	785.5271606	2168.823975	30.425	30.5	45.075	52.775	58.4	35.76888
2016	189.0938416	209.5256195	356.8054199	1364.109009	1189.544678	29.9	39.2	46.425	53.15	59	49.47439
2017	479.1775513	267.6893616	660.647522	1088.121338	1056.332275	28.475	38.6	45.775	52.2	59.6	42.53172
2018	143.0857849	317.3755493	322.0111694	470.9925842	1206.549927	27.8	26.5	41.425	45.225	66.425	39.07765
2019	235.6571655	333.6958008	517.9634399	719.3614502	1883.379639	30.175	22.625	35.9	51.725	57.175	48.17807
2020	327.9013672	267.0498047	553.2293091	358.3137207	946.6182861	31.7	32.975	44.025	48.75	59.075	42.21505
2021	289.7973328	241.196991	876.1647339	444.5087891	1225.991821	32.425	21.425	45.2	49.325	58.575	43.60283

Figure 5: Dataset before Standardization

Model Development and Application

1. Model Selection

Our target variable is wheat yield which is a continuous variable, the model type should be regression model. I create a hypothesis using commonsense, which is too high or too low value for precipitation or temperature will cause deduction of wheat yield. Under this context, multivariate linear regression will not work. For multivariate polynomial regression, I have to predefine the formulation. However, I cannot come up with a formulation without any supporting knowledge, also grid search will not work on this one because the parameter is unlimited. After careful consideration, I decide to use Random Forest regression model, Support Vector machine regression model. eXtreme Gradient Boosting regression model and K-Nearest Neighbors regression model.

2. Model Validation

I only have 21 rows of data in the model output dataset, because each year is a row of data and the monthly precipitation data from NASA only trace back to June 2000. The best validation method is leave one out cross validation. Compared to k-fold cross validation, leave one out cross validation can test on each datapoint taking each data into account as test data.

We choose root mean square error as the validation metrics for all regression model.

Below is the validation code:

```
def LeaveOneOut_validation(model,X,y):
    cv = LeaveOneOut()
    rmse_scores = cross_val_score(model, X, y, scoring='neg_mean_squared_error',
                                  cv=cv, n_jobs=-1)
    rmse = np.sqrt(np.mean(np.absolute(rmse_scores)))
    return rmse
```

3. Random Forest Regression

Random forest regression is a very powerful regression that ensemble multiple trees, the result is generated by getting the average of each tree result. I create lists for

model tuning, a n_estimators list contains 100, 300, 500, 700, and a max_depth list contains 3, 4, 5, 6, 7. Result shows in figure 6.

```
n_estimators:100 max_depth:3 rmse:4.806867211974307
n_estimators:100 max_depth:4 rmse:4.81029593912702
n_estimators:100 max_depth:5 rmse:4.794550511632239
n_estimators:100 max_depth:6 rmse:4.801527290644424
n_estimators:100 max_depth:7 rmse:4.810359993914521
n_estimators:300 max_depth:3 rmse:4.840345307645903
n_estimators:300 max_depth:4 rmse:4.818055545303003
n_estimators:300 max_depth:5 rmse:4.840582696127657
n_estimators:300 max_depth:6 rmse:4.82782185042032
n_estimators:300 max_depth:7 rmse:4.81907908466899
n_estimators:500 max_depth:3 rmse:4.855101422654656
n_estimators:500 max_depth:4 rmse:4.8331274737229375
n_estimators:500 max_depth:5 rmse:4.8390188174418185
n_estimators:500 max_depth:6 rmse:4.832987534511582
n_estimators:500 max_depth:7 rmse:4.823375878238689
n_estimators:700 max_depth:3 rmse:4.845068268430644
n_estimators:700 max_depth:4 rmse:4.835543650387363
n_estimators:700 max_depth:5 rmse:4.840449581085746
n_estimators:700 max_depth:6 rmse:4.836090345495909
n_estimators:700 max_depth:7 rmse:4.826672724198097
```

Figure 6: Random Forest Optimization

100 n_estimators and 5 max_depth get the best result. So, I retrain the model with this hyperparameter combo and use this model to do the comparison between different types of model.

4. eXtreme Gradient Boosting regression

First, I train the model with different loss function changing the objective parameter, one is 'reg:linear', another one is 'reg:tweedie'. Linear one gets the rmse value of 4.75, whereas tweedie just get the rmse of 11.9 which is a really terrible result.

Then, I tune the model using different max_depth and different min_child_weight (usually used to prevent overfitting problem). Result shows in figure 7:

The default hyperparameter already get the best performance. So, I deicide to stick with the model with default parameter.

```
max_depth:4 min_child_weight:1 rmse:4.75396270800269
max_depth:4 min_child_weight:2 rmse:4.882592074549768
max_depth:4 min_child_weight:3 rmse:5.145121406092007
max_depth:4 min_child_weight:4 rmse:5.018096239924533
max_depth:4 min_child_weight:5 rmse:5.002360674629478
max_depth:5 min_child_weight:1 rmse:4.752590023120308
max_depth:5 min_child_weight:2 rmse:4.880864964200236
max_depth:5 min_child_weight:3 rmse:5.145121406092007
max_depth:5 min_child_weight:4 rmse:5.018096239924533
max_depth:5 min_child_weight:5 rmse:5.002360674629478
max_depth:6 min_child_weight:1 rmse:4.752590023120308
max_depth:6 min_child_weight:2 rmse:4.880864964200236
max_depth:6 min_child_weight:3 rmse:5.145121406092007
max_depth:6 min_child_weight:4 rmse:5.018096239924533
max_depth:6 min_child_weight:5 rmse:5.002360674629478
max_depth:7 min_child_weight:1 rmse:4.752590023120308
max_depth:7 min_child_weight:2 rmse:4.880864964200236
max_depth:7 min_child_weight:3 rmse:5.145121406092007
max_depth:7 min_child_weight:4 rmse:5.018096239924533
max_depth:7 min_child_weight:5 rmse:5.002360674629478
max_depth:8 min_child_weight:1 rmse:4.752590023120308
max_depth:8 min_child_weight:2 rmse:4.880864964200236
max_depth:8 min_child_weight:3 rmse:5.145121406092007
max_depth:8 min_child_weight:4 rmse:5.018096239924533
max_depth:8 min_child_weight:5 rmse:5.002360674629478
```

Figure 7: XGBoost Optimization

5. Support Vector Regression

I try support vector regression with different kernel and different parameter. It has different result. The parameter of each model as below shown:

svr_rbf = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.1)

svr_lin = SVR(kernel="linear", C=100, gamma="auto")

svr_poly = SVR(kernel="poly", C=100, gamma="auto", degree=3, epsilon=0.1, coef0=1)

rbf kernel: The RBF kernel on two samples **x** and **x'**, represented as feature vectors in some input space, is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

poly kernel: For degree-*d* polynomials, the polynomial kernel is defined as:

$$K(x, y) = (x^\mathsf{T} y + c)^d$$

The model performance comparison will display on latter section.

6. KNN regression

Implementing model with different k value, rmse value with different k shows in figure 8.

	k	RMSE
0	2	5.199644
1	3	4.884486
2	4	4.278526
3	5	4.592731
4	6	4.602784
5	7	4.662829
6	8	4.618196
7	9	4.844086

Figure 8: KNN Optimization

The best k is k = 4. So, stick with 4.

7. Feature Importance Analysis

Feature Importance refers to techniques that calculate a score for all the input features for a given model - the scores simply represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict target variable.

There are mainly 3 types of feature importance. The first one is coefficient as feature importance come from linear regression and related variation model. The second one is feature importance from tree-based model such random forest and XGBoost. The third one is permutation feature importance, I have not used this

method in this study.

I choose random forest regression, XGBoost regression and support vector regression with linear kernel to perform feature importance.

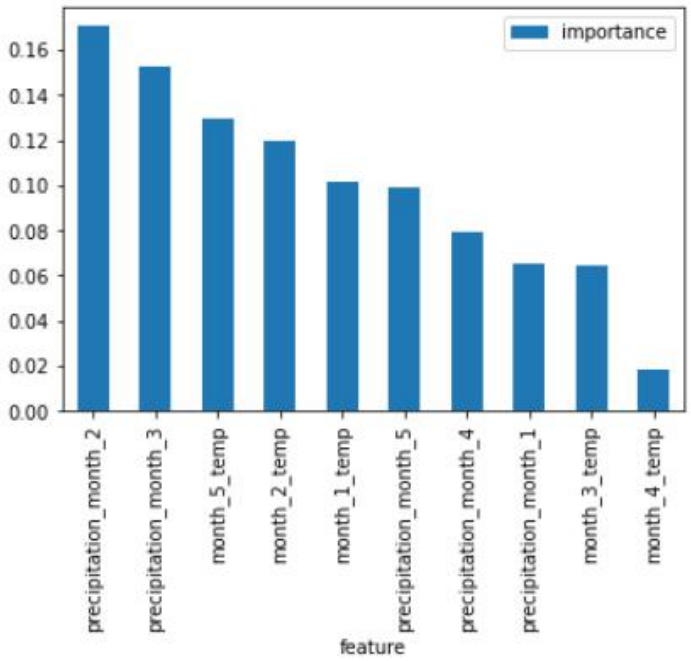

Figure 9: Feature Importance from Random Forest

From random forest feature importance plot, I am able to say that February and March precipitation impact the wheat yield most.

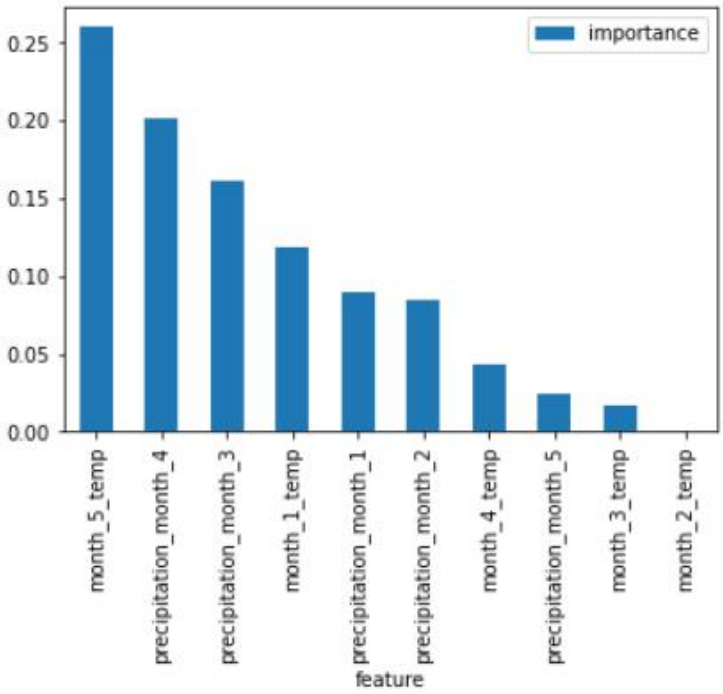

Figure 10: Feature Importance from XGBoost Regression

However, the plot of XGBoost regression feature importance shows different result. The top 3 most influencing variables are May temperature, April precipitation and March precipitation.

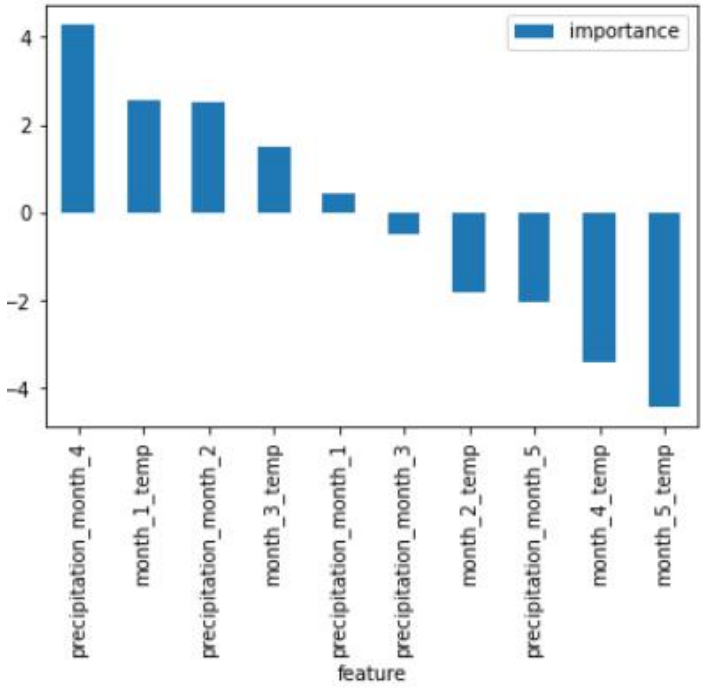

Figure 11: Feature Importance from Support Vector Regression

Support vector regression is capable of extracting coefficient value when I use linear kernel. Despite this model retain a relatively bad result, I am able to dig some insight form it because unlike true feature importance, coefficient value can display not only the importance score, but also detect the direction of impact (positive or negative). However, my initial hypothesis do not encourage a linear relation between dependent and independent variable, so I have to be careful when we try to interpret the plot.

I have been thinking that the feature importance analysis not only help me understand the contribution of each variable, but also served as second model validation method besides rmse especially in a small dataset. If the importance of feature are against our commonsense, the model could not perform good even though it get a lower rmse value because when I apply those model on a larger dataset to predict the target, the performance would drop down significantly.

Conclusion and Discussion

The result of all model after hyperparameter optimization shows in figure 12:

	ML_type	RMSE
0	Random Forest	4.794551
1	XGBoost_linear	4.752590
2	XGBoost_tweedie	11.903273
3	SVM(RBF)	4.652871
4	SVM(linear)	10.314442
5	SVM(polynomial)	4.480130
6	KNN	4.278526

Figure 12: Model Validation Comparison

XGBoost regression with tweedie kernel have the worst performance. SVM with poly kernel also perform not good, this is mainly because the relationship between dependent variable and independent variable is not a simple linear relationship. As hypothesis says, the wheat yield will only achieve the highest with moderate or suitable range of precipitation and temperature. Too high or too low in temperature will lead to reduce of wheat yield. The hypothesis is proved to some extent when we compare the result of SVM with different kernels. RBF and polynomial kernel will drastically improve the model performance. However, the XGBoost result is contradicted to my hypothesis, I guess this is because XGBoost is a type of ensemble model, more reliable and robust than SVM. Random Forest perform great as well. The last model is KNN regression model. The result KNN kind of surprise me, it gets the best performance among all other model. Normally, the performance of KNN is not the best model, even always be the worst model. But I cannot guarantee that KNN is the most reliable model especially the model training set is very small, so we should be cautious when we implement the model to try to predict the next couple years' wheat yield.

For future work, I am able to continue tuning the polynomial SVM model, I believe it can give me better result in this context. Also, despite temperature and precipitation is 2 important factors, we still are able to take fertilizer (the concentration of carbon dioxide), and different types of terrain in different geographic area into consideration. What's more, concentrating on different type of wheat such as hard red spring, soft red winter, white, durum wheat is a great topic to work on in the

future.

Reference

Pirttioja N, Carter TR, Fronzek S, Bindi M and others (2015) Temperature and precipitation effects on wheat yield across a European transect: a crop model ensemble analysis using impact response surfaces. Clim Res 65:87-105. https://doi.org/10.3354/cr01322

Mukherjee, A.; Wang, S.-Y.S.; Promchote, P. Examination of the Climate Factors That Reduced Wheat Yield in Northwest India during the 2000s. *Water* **2019**, *11*, 343. https://doi.org/10.3390/w11020343

Huffman, G.J., E.F. Stocker, D.T. Bolvin, E.J. Nelkin, Jackson Tan (2019), GPM IMERG Final Precipitation L3 1 month 0.1 degree x 0.1 degree V06, Greenbelt, MD, Goddard Earth Sciences Data and Information Services Center (GES DISC), 10.5067/GPM/IMERG/3B-MONTH/06

CPC Global Temperature data provided by the NOAA/OAR/ESRL PSL, Boulder, Colorado, USA. https://downloads.psl.noaa.gov/Datasets/cpc_global_temp/