

# YIELD FORECAST<sup>BETA</sup>

API Documentation 2020

[API Portal](#)

[GitHub Repo](#)

[Jupyter Notebook](#)

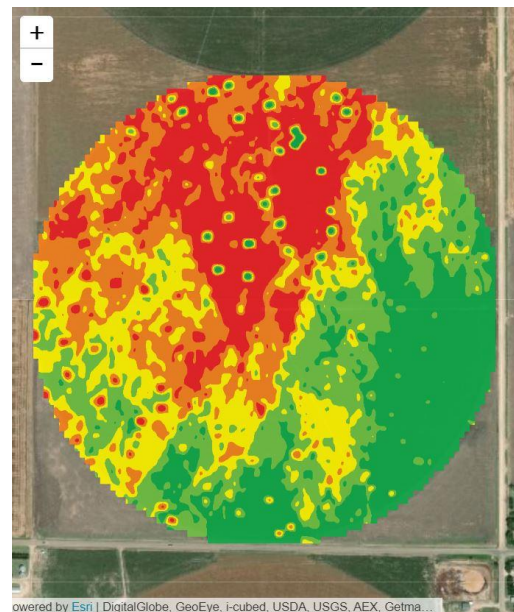
[Purchase](#)

## Service Overview

The Ag-Analytics® Yield Forecast API uses Artificial Intelligence algorithms to forecast the yield on a given field, based on geospatial data. The Yield Model API provides service by considering various factors like soil, vegetation index, location of the field, planting varieties to forecast the yield for a given field.

Crop yield is a function of a large set of parameters, many of which are outside the control of the farmer. The forces of nature are unpredictable and can make or break a growing season, while chemical applications can be rendered ineffective by new weeds or pests.

The current version of the yield model is relatively simple as compared to the number of factors that influence yield, however, it still provides insight and predictive power. Current yield model factors are location, crop season, seeding characteristics, planting date, application characteristics, soil characteristics, past growing history, weather data, and satellite imagery.



*YieldAI shown in FarmScope*

## Model Specifications

The current model requires only six inputs to predict the yield for a given field. This may seem like a small amount, but a large amount of external data is collected based on these input values. The inputs are the bare minimum information that is required, and is information that would be impossible to obtain otherwise. Below are listed the types of data that inform the model and their origins, along with how the inputs help obtain the external data.

1. **Location:** the location of the field plays a role in the predicted yield due to the geospatial influence on growing season, type of climate, and possible farming practices. The SHAPE parameter provides latitude and longitude coordinates that can be used to find these patterns.

2. **Season:** for back-filling applications (i.e., determining what the yield was in the past for unknown fields), crop season plays a big role in keying the model in to factors such as adverse weather or unusual conditions specific to that year.
3. **Weather conditions:** the crop season also helps us find the weather data for that particular growing season. Without it, it would be impossible to know what weather patterns to apply to predict the yield. In future models, the user will be able to select different weather scenarios in order to see the effect of average or extreme conditions.
4. **Soil characteristics:** soil characteristics are obtained from both the SSURGO and POLARIS public soil datasets. The model uses the location of the field and the Soil API to find its specific soil properties.
5. **Past growing history:** past growing history is obtained from the Cropland Data Layer (CDL) database. The influence of crop rotation practices or lack thereof can factor in to how well the current crop performs.
6. **Weather data:** weather data is obtained through the PRISM dataset provisioned by Oregon State University. This requires the location and crop season for the field.
7. **Satellite imagery:** satellite imagery can illuminate the growing pattern of a crop throughout the growing season. Using multispectral bands over a period of many weeks throughout the year can help the model determine whether a crop is thriving.

## POST Request

### Request Parameters

Parameter	Data Type	Required?	Default	Options	Description
<b>MODELNAME</b>	Text	Yes	--	"NN" (Neural Network)	The type of AI Model to be used.
<b>SHAPE</b>	Text	Yes	--	GeoJSON	The shape of the desired area-of-interest. See Fig. 1 for example.
<b>ScalarVariables</b>	Text	Yes	--		The constants for the given Shape – includes all parameters below.
<b>CropSeason</b>	Text	Yes	--	2013-2019	Growing season year for prediction. Ex. "2018"
<b>PlantingDay1</b>	Text	Yes	--	"mm-dd-yyyy"	Date when planting occurred for crop of interest. Ex. "05/20/2018"
<b>SeedingDensity</b>	Integer	Yes	--	Any number	The number of seeds planted per acre. Ex. 10,000



<b>HarvestDay</b>	Integer	Yes	--		Date when harvest is expected. Ex. "10/20/2018"
-------------------	---------	-----	----	--	---

## Header Parameters

**content-type:** "application/json"

## Request Example

*POST Request Example – application/json*

```
{
  "MODELNAME": "NN",
  "SHAPE": "{ \"type\": \"Feature\", \"properties\": {}, \"geometry\": { \"type\": \"Polygon\",
  \"coordinates\": [[ [-100.953840994, 38.5946753571], [-100.953832008,
  38.5948720599], [-100.953876941, 38.5952162884], [-100.953957821, 38.5955324152],
  [-100.953984781, 38.5955745654], [-100.954029714, 38.5957361407], [-100.954245394,
  38.5961716896], [100.954452087, 38.5964807873], [100.95473966, 38.5968179832], [-
  100.954910406, 38.596965506], [-100.954910406, 38.5969795557], [-100.962998393,
  38.5947877588], [-100.962989406, 38.5947666835], [-100.962917513, 38.5947526333], [-
  100.962935486, 38.5947245329], [-100.962926499, 38.5946894073], [-100.962881566,
  38.5947034575], [-100.962665886, 38.5946823822], [100.958541013, 38.5946753571], [-
  100.9584152, 38.5947175077], [-100.958316347, 38.5947175077], [-100.958298374,
  38.5947034575], [-100.958154587, 38.5946753571], [-100.953840994,
  38.5946753571]]] } }",
  "ScalarVariables": {
    "CropSeason": "2018"
    "PlantingDay1": "05/20/2018",
    "SeedingDensity": "30000",
    "HarvestDay": "10/20/2018"
  }
}
```



## POST Response

### Response Parameters

Parameter	Data Type	Description
<b>raster_filename</b>	String	URL to use in GET request to retrieve predicted raster file.
<b>rasterinfo</b>	List of Dictionaries	Container for the features and metadata information for the raster.
<b>rasterinfo.attributes</b>		
<b>CellSize</b>	List	Size of a single cell in the raster in degrees. (0.0001, -0.0001) roughly corresponds to an 8 meter by 8-meter square on the Earth's equator.
<b>CoordinateSystem</b>	String	Information about the coordinate system being used for calculations.
<b>Extent</b>	String	Extents of the result raster. Specifies the bottom left and top right corners of the field raster in degrees.
<b>Legend</b>	List of Dictionaries	Legend gives the following details for each range of values: Area: I. <b>Area</b> : Area covered II. <b>AreaUnit</b> : Unit of Area covered III. <b>AreaPercent</b> : Area covered in percentage. IV. <b>Count</b> : # of pixels from the result raster in that range V. <b>CountAllPixels</b> : Total # of pixels in the result raster VI. <b>CropID</b> : Code for the crop identified by model. See Figure 4 VII. <b>CropName</b> : Crop name identified by the model VIII. <b>Color</b> : Hex color used for the crop type
<b>Matrix</b>	List	Rows and columns containing the attributes below
<b>Max</b>	Number	Maximum value from the result raster
<b>Min</b>	Number	Minimum value from the result raster
<b>Mean</b>	Number	Average value from the result raster
<b>Percentile5</b>	Number	5 <sup>th</sup> percentile value from the result raster
<b>Percentile95</b>	Number	95 <sup>th</sup> percentile value from the result raster
<b>pngb64</b>	Link	Base64png image of the result raster with legend entries.



## Response Example

*POST Response Example – application/json*

```
{
  "raster_filename": "result_yieldraster_20191121_222140_9894.tif",
  "rasterinfo": [{
    "attributes": {"CellSize": [0.0001, -0.0001],
    "CoordinateSystem": "GEOGCS[\"WGS 84\",DATUM[\"WGS_1984\",SPHEROID[\"WGS
    84\",6378137,298.257223563,AUTHORITY[\"EPSG\",\"7030\"]],AUTHORITY[\"EPSG\",\"6326\"
    ]],PRIMEM[\"Greenwich\",0],UNIT[\"degree\",0.0174532925199433],AUTHORITY[\"EPSG\"
    ,\"4326\"]]",
    "Extent": "-89.209236, 40.969983000000006, -89.199536000000001, 40.977383",
    "Legend": [
      {
        "Area": "100.0 %",
        "Count": 6511,
        "CountAllPixels": 6511,
        "Value": 1.0,
        "Variety": "Variety 1.0",
        "color": "#6e9a9e"
      }
    ],
    "Matrix": [74, 97],
    "Max": 1.0,
    "Mean": 1.0,
    "Min": 1.0,
    "OID": 0,
    "Percentile5": 1.0,
    "Percentile95": 1.0,
    "Variety": "Variety",
    "pngb64": "data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAGEAAABKCAYAAA
    CrbTpWAAAB6k1EQVR4nO2ZMZKDMAxFP5lcgQtSUOYwlFvsBfcS26xnM14bDDHwJf1XZZIikp6Qw
    AwwZvrr++fuGD5lmafX6SER6zzuDkBIAGwSQIAkECAJBEGCAZJAgCQQIAkESA=="
  }
  ]
}
```



## GET Request

### Request Example

The GET request to retrieve the tif image using the file name from the POST response.

```
https://ag-analytics.azure-api.net/yield-forecast/?filename=result_yieldraster_20191126_183736_2356.tif
```

### Request Parameters

Parameter	Data Type	Required?	Default	Options	Description
<b>filename</b>	text	Yes	--	.tif file	file name returned by POST request

### Response Parameters

Parameter	Data Type	Description
<b>file</b>	.tif	Tiff file will be download to the computer of the caller with the name that was used to call the API.



Please contact [support@analytics.ag](mailto:support@analytics.ag) or [josh@ag-analytics.org](mailto:josh@ag-analytics.org) with any comments or questions.

[Terms of Use and Privacy](#)

