

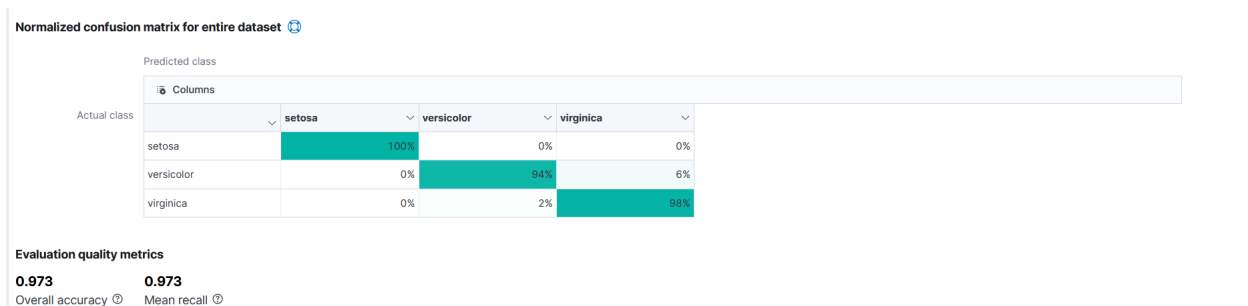
Because we were unable to use the project VM this week, we have developed the following plan to progress the project.

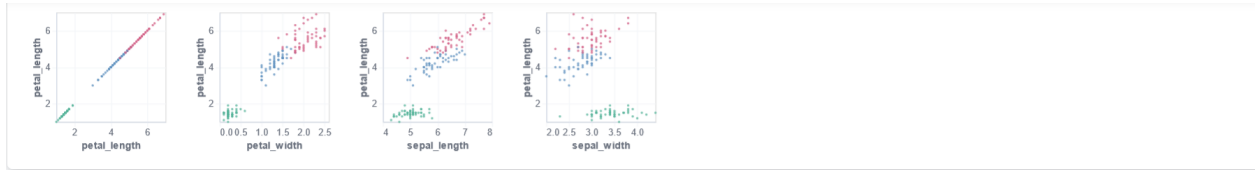
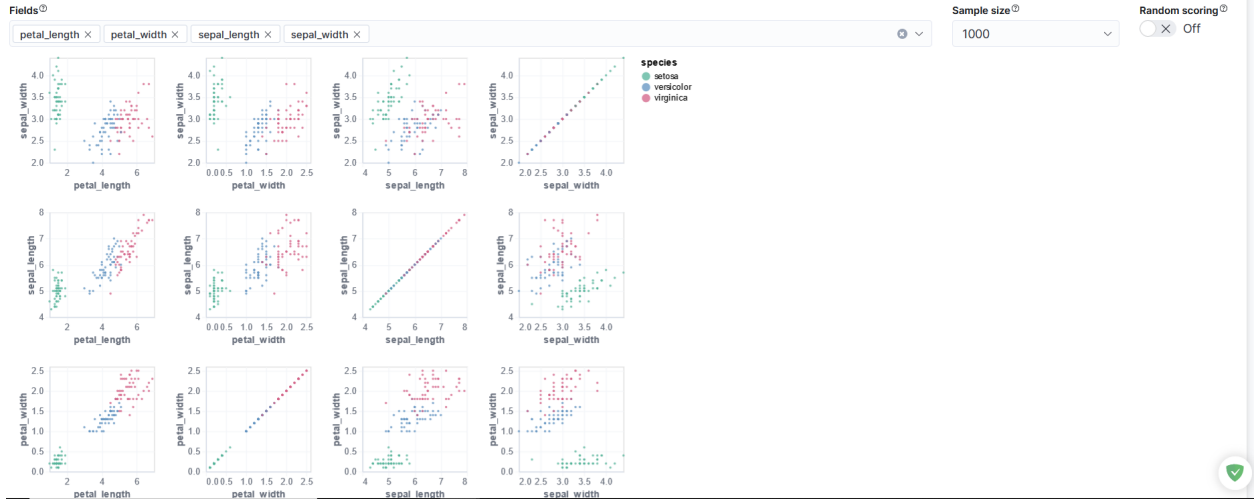
What we would have done:

1. Use Elasticsearch to further develop our machine learning algorithms (specifically decision tree) with data sets to analyze points within the VM.
 - a. Gain further in depth knowledge of the applications and properties of Elasticsearch.
2. Use Kibana to visualize the results.

What we have done:

1. Used Elasticsearch on free trial of elastic to analyze Iris data set





Results ^

Total docs

150

Showing documents for which predictions exist

Histogram charts 3 columns hidden Sort fields

ml.is_training	ml.species_prediction	species	ml.prediction_probability	ml.prediction_score	ml.top_classes	ml_incremental_id	petal_length
true	setosa	setosa	0.956	0.314	[{"class_score": 0.31427...}]	0	1.4
true	setosa	setosa	0.956	0.314	[{"class_score": 0.31427...}]	1	1.4
true	setosa	setosa	0.956	0.314	[{"class_score": 0.31427...}]	3	1.5
true	setosa	setosa	0.956	0.314	[{"class_score": 0.31427...}]	4	1.4
true	setosa	setosa	0.956	0.314	[{"class_score": 0.31427...}]	7	1.5

kddjson.txt - Notepad

File Edit Format View Help

```
{
  "description": "",
  "source": {
    "index": "kdd20percent",
    "query": {
      "match_all": {}
    }
  },
  "dest": {
    "index": "kdddataset"
  },
  "analyzed_fields": {
    "includes": [
      "Column1",
      "Column10",
      "Column11",
      "Column12",
      "Column13",
      "Column14",
      "Column15",
      "Column16",
      "Column17",
      "Column18",
      "Column19",
      "Column2",
      "Column20",
      "Column21",
      "Column22",
      "Column23",
      "Column24",
      "Column25",
      "Column26",
      "Column27",
      "Column28",
      "Column29",
      "Column3",
      "Column30",
      "Column31",
      "Column32",
      "Column33",
      "Column34",
      "Column35",
      "Column36",
      "Column37",
      "Column38",
      "Column39",
      "Column4",
      "Column40",
      "Column41",
      "Column42",
      "Column43",
      "Column5",
      "Column6",
      "Column7",
      "Column8",
      "Column9"
    ]
  },
  "analysis": {
    "classification": {
      "dependent_variable": "Column42",
      "num_top_feature_importance_values": 0,
      "training_percent": 80,
      "num_top_classes": -1
    }
  },
  "model_memory_limit": "909mb",
  "max_num_threads": 1
}
```

JSON code above:

Results

Total docs
>10000

Showing documents for which predictions exist

📊 Histogram charts 📄 41 columns hidden ⚙️ Sort fields

ml.is_training	ml.Column42_predicti...	Column42	ml.prediction_probabi...	ml.prediction_score	ml.top_classes	Column1	Column10
true	normal	normal	0.994	0.365	[{"class_score":0.36537...	0	0
true	normal	normal	0.944	0.347	[{"class_score":0.34703...	0	0
true	neptune	neptune	1	0.324	[{"class_score":0.32375...	0	0
true	normal	normal	1	0.368	[{"class_score":0.36758...	0	0
true	neptune	neptune	0.999	0.324	[{"class_score":0.32362...	0	0
true	neptune	neptune	1	0.324	[{"class_score":0.32373...	0	0
true	neptune	neptune	1	0.324	[{"class_score":0.32374...	0	0
true	neptune	neptune	1	0.324	[{"class_score":0.32374...	0	0
true	neptune	neptune	0.999	0.324	[{"class_score":0.32362...	0	0
true	neptune	neptune	1	0.324	[{"class_score":0.32374...	0	0
true	normal	normal	1	0.368	[{"class_score":0.36758...	0	0
true	warezcilent	warezcilent	0.968	0.346	[{"class_score":0.34605...	0	0
true	neptune	neptune	1	0.324	[{"class_score":0.32370...	0	0
true	neptune	neptune	1	0.324	[{"class_score":0.32372...	0	0

Model evaluation [↗](#) [Classification evaluation docs](#)

Job status: stopped docs evaluated: 25192

Normalized confusion matrix for entire dataset [📄](#)

Actual class	Predicted class							
	back	ipsweep	neptune	nmap	normal	portsweep	4 more	
back	100%	0%	0%	0%	0%	0%	0%	0%
ipsweep	0%	100%	0%	0%	0%	0%	0%	0%
neptune	0%	0%	100%	0%	0%	0%	0%	0%
nmap	0%	0%	0%	100%	0%	0%	0%	0%
normal	0%	0%	0%	0%	100%	0%	0%	0%
portsweep	0%	0%	0%	0%	0%	0%	100%	0%

Evaluation quality metrics

0.999 **0.948**
 Overall accuracy [📄](#) Mean recall [📄](#)

Next Step: Using code in elastic to analyze IRIS data set using python