Pokemon Dataset

Regression and Classification

Presentation Layout

- Objectives
- Conclusions
- The Dataset
- The Process
  - Dragon-type Classification
  - HP Regression
- Summary

Objectives

- Classification
    - Can we predict if a Pokemon is a Dragon-type with the given features?
- Regression
    - Can we predict a Pokemon’s HP stat with the given features?

Conclusions

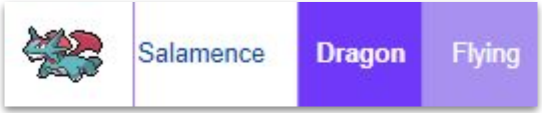
- Dragon-type Classification
  - K-Nearest Neighbors with PCA
    - Test set accuracy = 0.93
    - macro f1-score = 0.63
- HP Stat Regression
  - Ridge Regression with alpha = 20
  - 52% of the variability

The Dataset

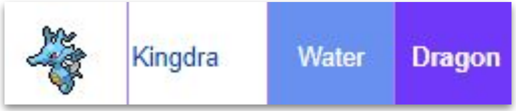
- Includes all Pokemon from the first 8 generations.
  - n = 1032 pokemon
- From Kaggle
  - https://www.kaggle.com/datasets/maca11/all-pokemon-dataset

Dragon-type Classification

- Columns to use
  - HP, Att, Def, Spa, Spd, Spe,
  - Type 1, Type 2
  - Generation, Experience type, Experience to level 100
  - Catch Rate, Height, Weight
- Removed “mega evolutions” for classification
  - Their stats might be different
  - Might skew the data
- Created binary outcome “isDragon”
  - Type 1 and Type 2 Dragons



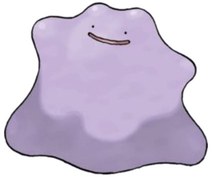
	Salamence	Dragon	Flying



	Kingdra	Water	Dragon

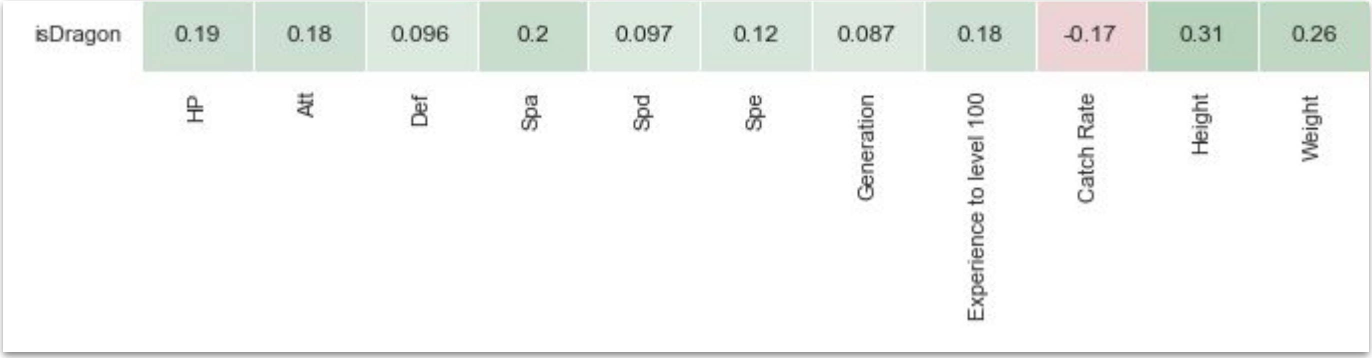
Stratified Train-Test Split

- To make sure the train and test sets have the same proportions of Dragon-types and non-Dragon-types.



Exploratory Data Analysis

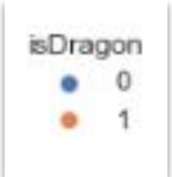
- Correlation Matrix
  - Notable correlations with isDragon
    - Positive: Height, Weight, SPA, HP, ATT, Experience to level 100
    - Negative: Catch Rate

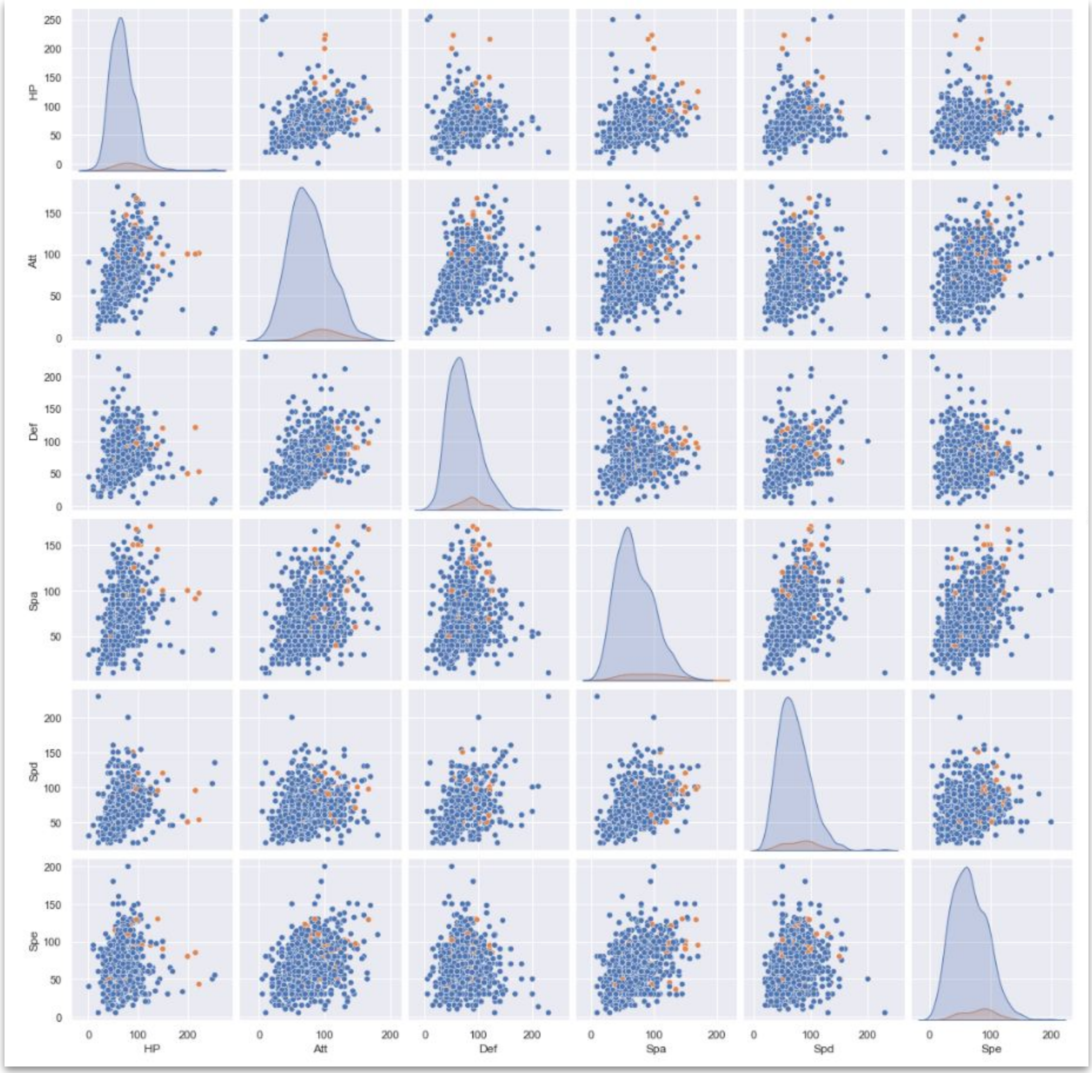


	HP	Att	Def	Spa	Spd	Spe	Generation	Experience to level 100	Catch Rate	Height	Weight
isDragon	0.19	0.18	0.096	0.2	0.097	0.12	0.087	0.18	-0.17	0.31	0.26

Pairplot

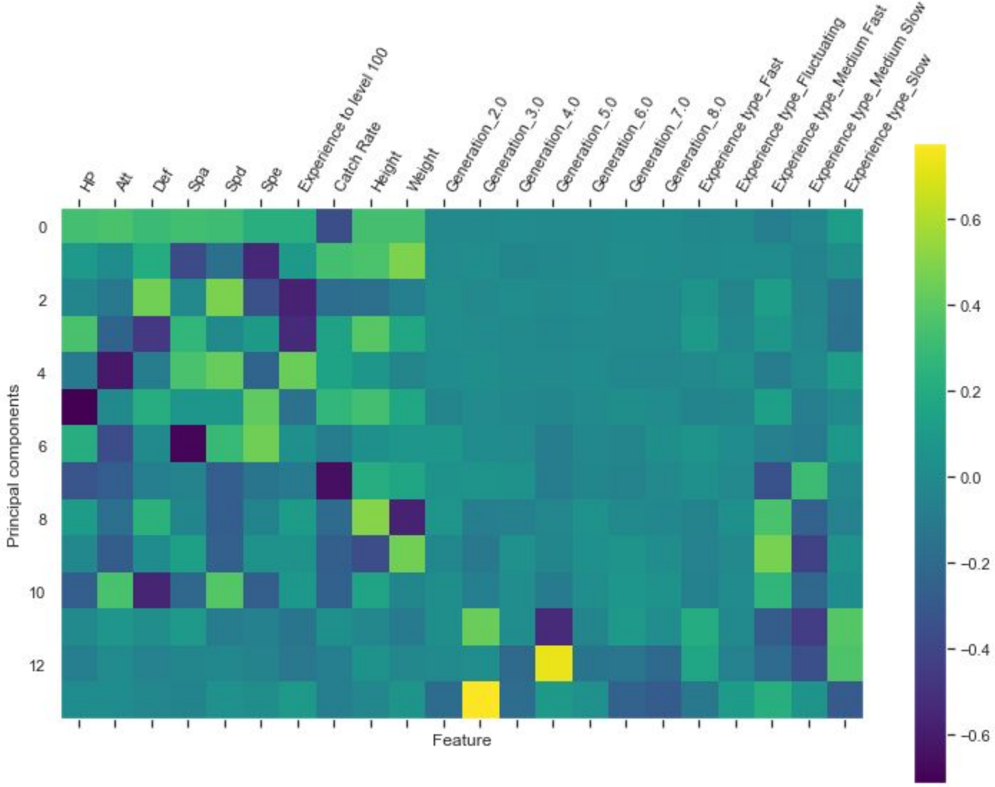
- 6 main stats





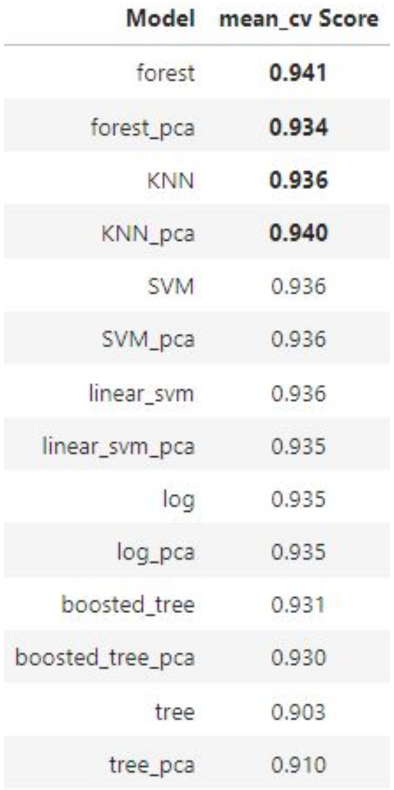
PCA Heatmap

- First component's more extreme feature weights:
  - Positive:
    - The six main stats
    - Experience to level 100
    - Height
    - Weight
  - Negative:
    - Catch Rates
- The stronger, bigger, and the more effort it takes to train the Pokemon
- The harder it is to catch



Classification Models

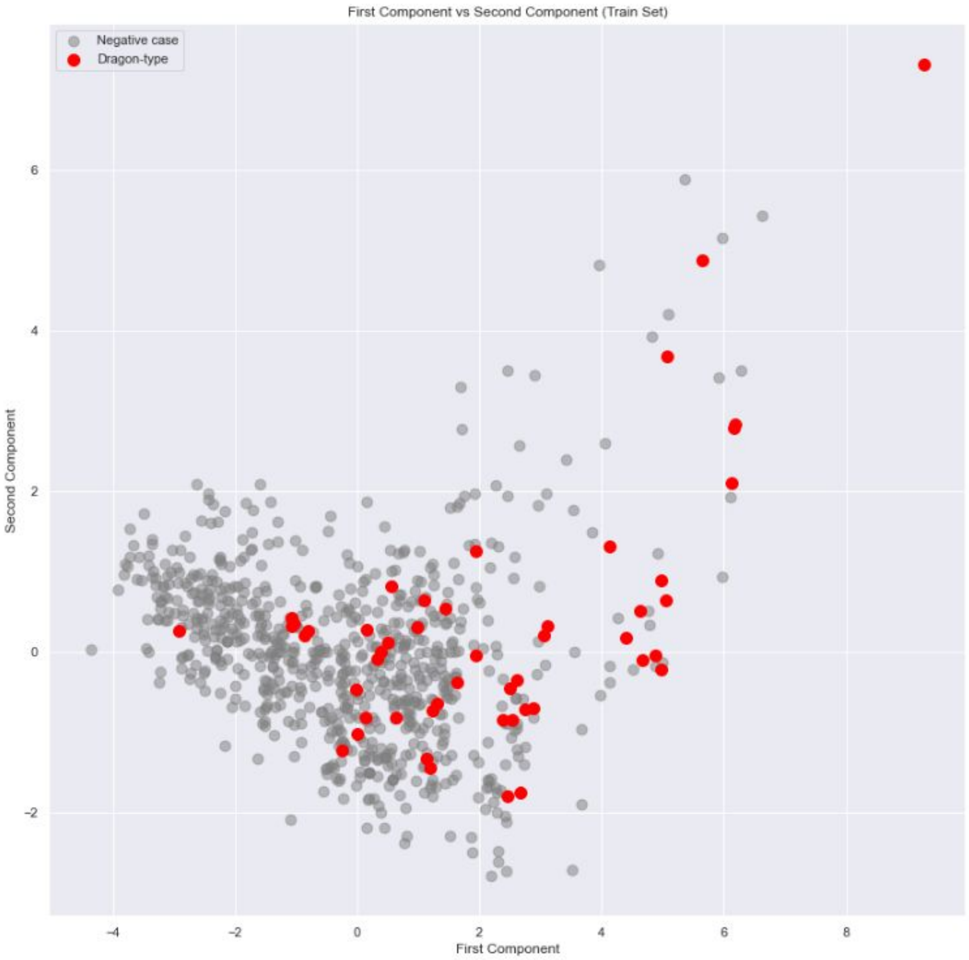
- Mean cross validation with stratified shuffle splits
  - Evaluate the models on the training set
  - Tested the models with and without PCA
- The two best models were Random Forest and KNN



Model	mean_cv Score
forest	**0.941**
forest_pca	**0.934**
KNN	**0.936**
KNN_pca	**0.940**
SVM	0.936
SVM_pca	0.936
linear_svm	0.936
linear_svm_pca	0.935
log	0.935
log_pca	0.935
boosted_tree	0.931
boosted_tree_pca	0.930
tree	0.903
tree_pca	0.910

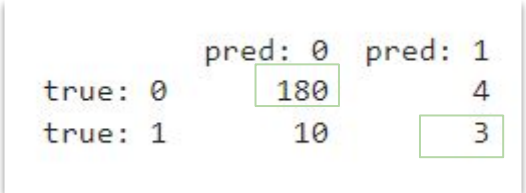
Plot Train Set

- Why KNN and Random Forest performed the best?
- Dragon-types
  - Lean toward the top right corner
  - Little groupings of them all over the plot, which might hint as to why K-Nearest Neighbors worked well.

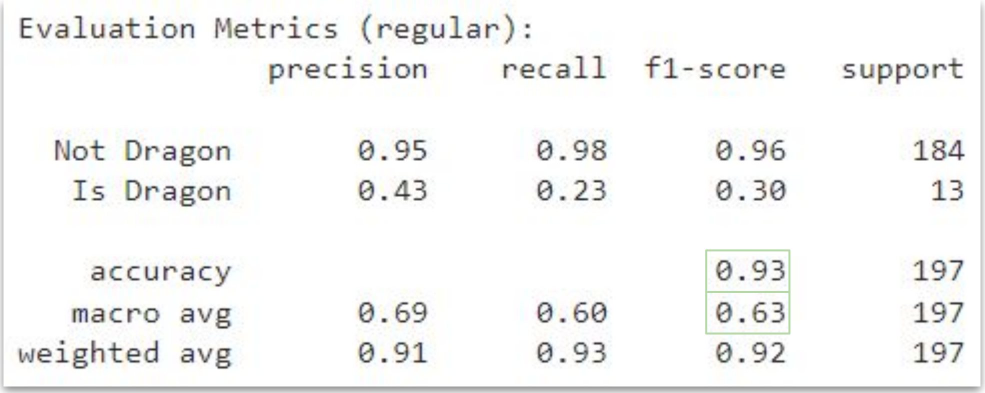


Evaluation Metrics with Test Set

- K-Nearest Neighbors with PCA



```
          pred: 0  pred: 1
true: 0       180        4
true: 1        10        3
```



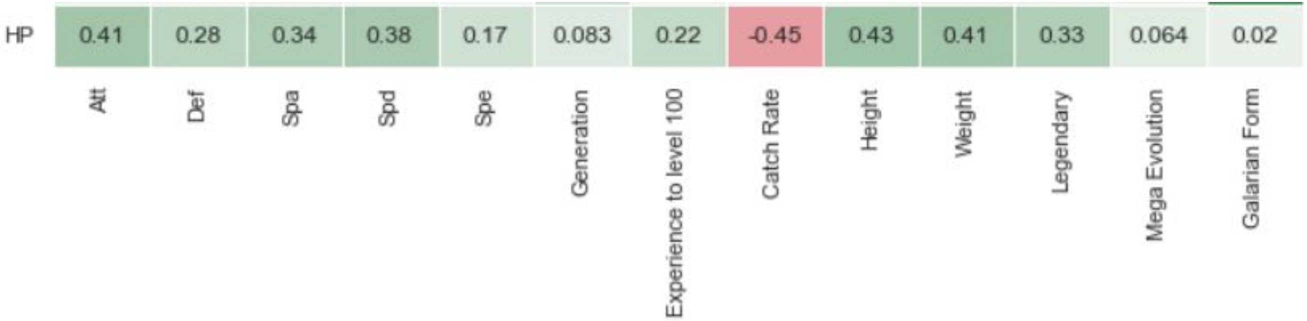
```
Evaluation Metrics (regular):
              precision    recall  f1-score   support

  Not Dragon       0.95      0.98      0.96       184
   Is Dragon       0.43      0.23      0.30        13

    accuracy                           0.93       197
   macro avg       0.69      0.60      0.63       197
weighted avg       0.91      0.93      0.92       197
```

Part 2: Regression for HP

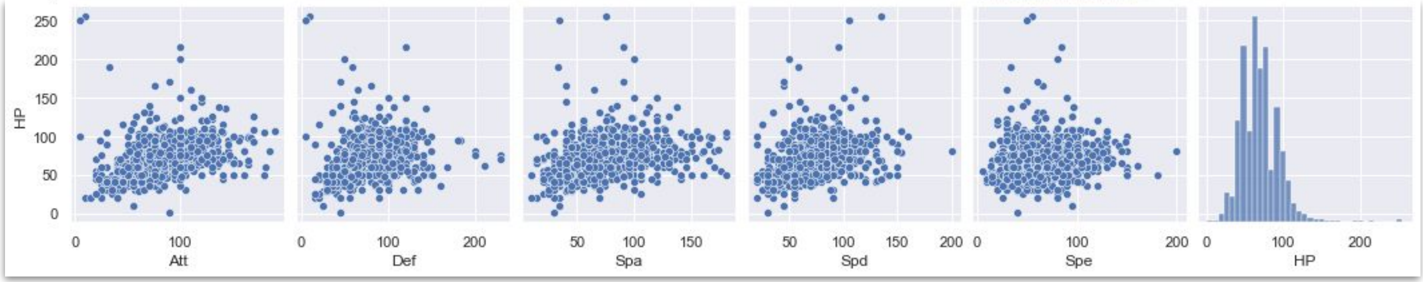
- Features most correlated with HP:
  - Positive
    - Height
    - Weight
    - Att
  - Negative
    - Catch Rate



	Att	Def	Spa	Spd	Spe	Generation	Experience to level 100	Catch Rate	Height	Weight	Legendary	Mega Evolution	Galarian Form
HP	0.41	0.28	0.34	0.38	0.17	0.083	0.22	-0.45	0.43	0.41	0.33	0.064	0.02

Pairplot of Main Stats

- Some correlation with all
- Most correlation with HP
    - Att
    - Spa
    - Spd



Linear Regression (Backwards Approach)

- Model 1
  - Full model
    - All 57 features
    - $R^2 = 0.47$
    - P-value = 0
- Model 2
  - Drop all features with above 0.10 p-value
  - Feature p-values may change as others are removed
  - 0.10 p-value cut-off point gives features that might drop below 0.05 p-value a chance
    - 19 Features
    - $R^2 = 0.41$
    - P-value = 0
- Model 3
  - Drop features with above 0.05 p-value
    - 15 Features
    - $R^2 = 0.41$
    - P-value = 0

Ridge and Lasso + GridSeachCV

- Ridge
    - alpha = 20
- Lasso
    - alpha = 0
        - No regularization
        - Same as linear regression

Final Results

- Full Model Linear Regression
  - Training set R^2: 0.47
  - Test set R^2: (negative value)
- 15 Feature Linear regression model
  - Training set R^2: 0.41
  - Test set R^2: 0.47
- Ridge Regression model (alpha = 20)
  - Training set R^2: 0.46
  - Test set R^2: 0.52
- Lasso (default parameters)
  - Training set R^2: 0.35
  - Test set R^2: 0.47
  - Features Included: 10

Summary

- Best Dragon-type classification model
  - KNN with PCA
- Best HP stat regression model
  - Ridge Regression with alpha = 20