Predicting Red Wine Quality

Sakshi Jawarani

Sj8em

DS 6003 - Spark Assignment

Motivation

- Being a Red Wine enthusiast myself, I decided to pick a Red Wine quality Dataset from Kaggle.
- I tried to predict the quality of red wine based on variables provided – which included the following:

winequality-red.csv (98.58 KB)

12 of 12 columns Views

	# fixed acidity	# volatile acidity	# citric acid	# residual sugar	# chlorides	# free sulfur dioxide	# total sulfur dioxide	# density
	most acids involved with wine or fixed or nonvolatile (do not evaporate readily)	the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste	found in small quantities, citric acid can add 'freshness' and flavor to wines	the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and	the amount of salt in the wine	the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents	amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2	the density of water is close to that of water depending on the percent alcohol and sugar content
	4.6 – 15.9	0.12 – 1.58	0 – 1	0.9 – 15.5	0.01 – 0.61	1 – 72	6 – 289	0.99 – 1
1	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978
2	7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968
3	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997
4	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998
5	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978
6	7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978
7	7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964
8	7.3	0.65	0.0	1.2	0.065	15.0	21.0	0.9946
9	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968

Code Snippet

- Reading data uploaded to S3 bucket
- Wrote data to spark data frame from parquet
- Exploratory Analysis
- Vectorization
- Machine Learning (Linear Regression)
- Model Evaluation & Visualization

```
In [20]: # renaming
trainingDF = trainingDF.withColumnRenamed("quality", "label").withColumnRenamed("alcohol", "features")
testDF = testDF.withColumnRenamed("quality", "label").withColumnRenamed("alcohol", "features")
```

ML

1. Train
2. Predict
3. Evaluate

```
In [21]: from pyspark.ml.regression import LinearRegression, LinearRegressionModel

lr = LinearRegression()
lrModel = lr.fit(trainingDF)
```

```
In [22]: type(lrModel)

Out[22]: pyspark.ml.regression.LinearRegressionModel
```

```
In [23]: predictionsAndLabelsDF = lrModel.transform(testDF)

print(predictionsAndLabelsDF.orderBy(predictionsAndLabelsDF.label.desc()).take(5))

[Row(label=8, features=DenseVector([11.4]), prediction=5.989037634520461), Row(label=7, features=DenseVecto
prediction=6.308128519154092), Row(label=7, features=DenseVector([11.7]), prediction=6.09540126273167), Ro
features=DenseVector([11.5]), prediction=6.02449217725753), Row(label=7, features=DenseVector([12.8]), pre
85401232839442)]
```

Visualization

- Precited values

vs residual plot

```
import matplotlib.pyplot as plt

plt.scatter(predicted, residual)
plt.xlabel("predicted values")
plt.ylabel("residuals")
```

```
Text(0,0.5,'residuals')
```