Group 1
Final Project

Kyle Schneider, Carolina Roca

Topic Selection, Source & Reasoning

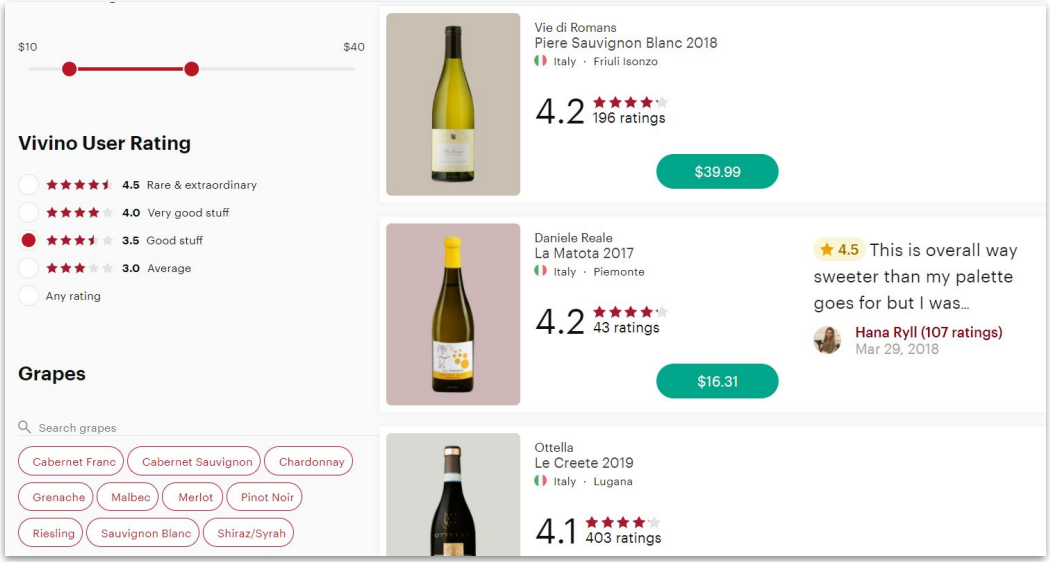
Wine Not?

The topic we chose to pursue was wine as we both enjoy it, and Kyle works in the CPG industry. The dataset we chose originated from Vivino and was hosted on Kaggle for extraction. It contains information about specific wines, where they come from, and how the public has rated them.

As wine consumers, we understand that vintage can greatly affect the quality, price, and taste of finished wine. As data scientists and machine learning practitioners, we want to understand if the features available in the dataset can lead to an accurate prediction of rating.

Our Data Source

vivino



kaggle

~5K rows of wine vintage ratings along with pricing, geographical, and other data

Question to answer...

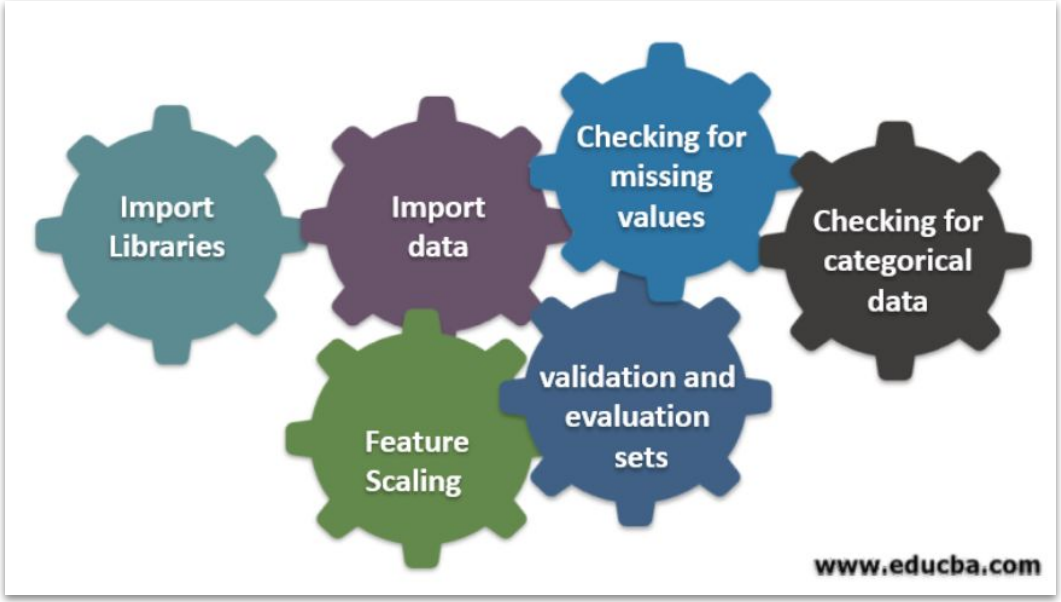
Using the available data from Vivino wine ratings, can we accurately predict a given vintage's rating?

Understanding our data...

Tableau snips

Building the database...

Machine Learning Model - Preprocessing




```python
#Import initial dependencies

import pandas as pd
import pyodbc

#Establish connection to SQL database

conn = pyodbc.connect('Driver={ODBC Driver 17 for SQL Server};'
                      'Server=tcp:group1-owner-nu.database.windows.net,1433;'
                      'Database=final-project;'
                      'Persist Security Info=False;'
                      'Uid=GROUPDB1NU;'
                      'Pwd=NU02282021!;'
                      'MultipleActiveResultSets=False;'
                      'Encrypt=Yes;'
                      'TrustServerCertificate=No;'
                      'Connection Timeout=30;')


cursor = conn.cursor()

#Read table from SQL

initial_df = pd.read_sql("SELECT * FROM dbo.final_table", conn)

initial_df.head()
```

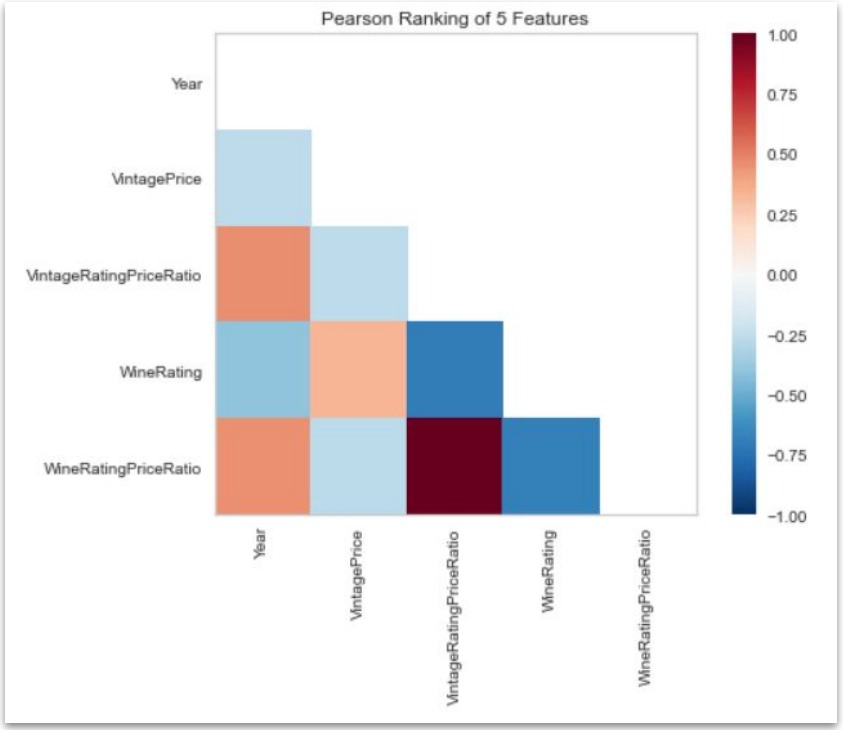
```python
from sklearn.compose import ColumnTransformer
from sklearn.compose import make_column_transformer
from sklearn.compose import make_column_selector

ct = make_column_transformer(
    (StandardScaler(), make_column_selector(dtype_include=np.float64)),
    (OrdinalEncoder(), make_column_selector(dtype_include=object))
)
ct.fit_transform(X)
```

Machine Learning Model - Feature Selection

FullName	Winery	Year	Vintage Price	Vintage Rating Price Ratio	Wine Rating	Wine Rating Price Ratio

Description of Analysis Phase




```
# visualize pipeline
from sklearn import set_config
set_config(display="diagram")
model
```

```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('standardscaler',
                                                  StandardScaler(),

ColumnTransformer(transformers=[('standardscaler', StandardScaler(),

StandardScaler()
OrdinalEncoder()
LinearRegression()
```

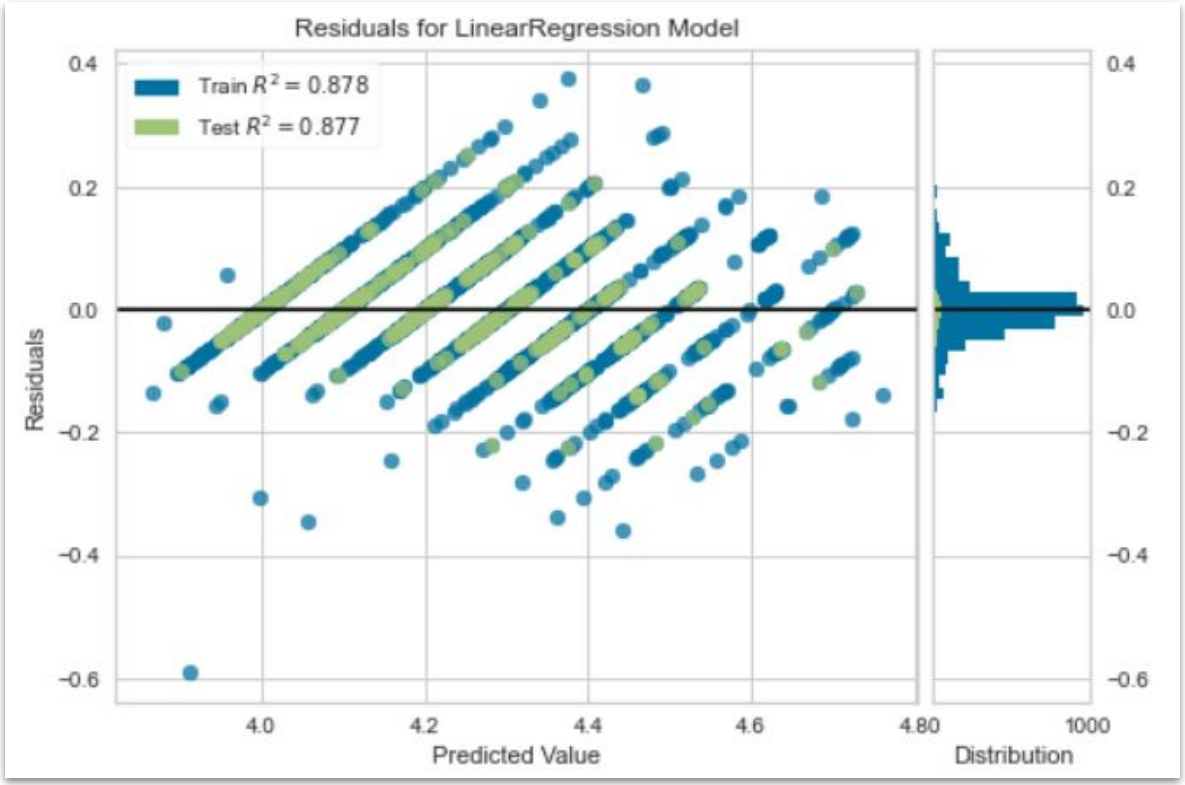
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
model = make_pipeline(ct, LinearRegression())
model.fit(X, y)
model.score(X_test, y_test)
```

```
0.8640005437048607
```

```
X_encoded = ct.fit_transform(X)
y_encoded = y
lr = LinearRegression()
lr.fit(X_encoded, y_encoded)
lr.score(X_encoded, y_encoded)
```

```
0.877734987463609
```

Results

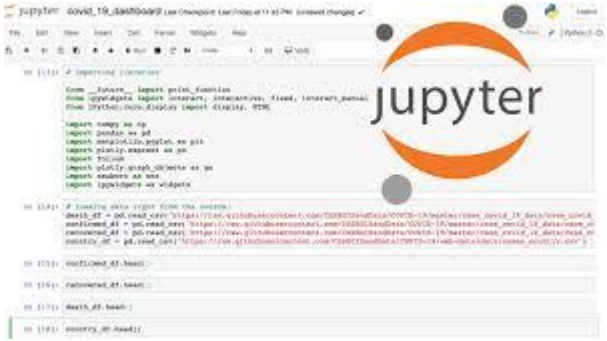



```
1  from sklearn.metrics import mean_squared_error
2
3  predictions = model.predict(X_test)
4  MSE = mean_squared_error(y_test, predictions)
5  r2 = model.score(X_test, y_test)
```

```
1  print(f"MSE: {MSE}, R2: {r2}")
```

```
MSE: 0.003395387554488735, R2: 0.877407772833871
```

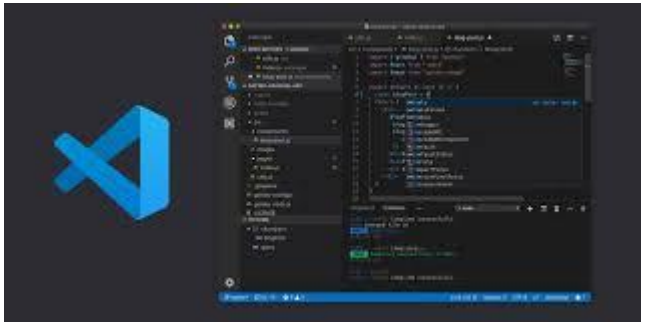
Tools and Resources



Microsoft SQL Server Management

Google Slides

GitHub



Appendix