Building a Wine Quality Prediction Model Using the SEMMA Methodology

S

Satvik Atmakuri

4 min read · Sep 27, 2024

In this article, we'll explore how to use the SEMMA methodology to build a machine learning model that predicts wine quality. The SEMMA process — Sampling, Exploration, Modification, Modeling, and Assessment — is a widely used data mining framework that guides the workflow from data collection to model evaluation.

Introduction to SEMMA

SEMMA is a data mining methodology that ensures a systematic approach to building machine learning models. Each step focuses on a particular aspect of data processing and modeling, making it a robust framework for solving predictive modeling problems.

The acronym SEMMA stands for:

- Sample: Select a subset of the data, or use the entire dataset if manageable.
- Explore: Analyze the data to find trends, correlations, and anomalies.
- Modify: Clean, transform, and prepare the data for modeling.
- Model: Develop predictive models using appropriate algorithms.
- Assess: Evaluate model performance and ensure generalizability.

Dataset Overview: Wine Quality

For this project, we use a publicly available dataset on wine quality. This dataset contains 11 physiochemical properties of wine (such as acidity, sulfur dioxide, and alcohol content) and the corresponding wine quality rating, a score between 3 and 8, representing the quality.

Our goal is to build a machine learning model to predict the wine quality based on its chemical properties.

Step 1: Sample

In this step, we determine whether to use a sample or the entire dataset. Since the wine quality dataset contains only 1,143 entries, it is small enough to use the full dataset without the need for sampling.

```
# Load the dataset
import pandas as pd
```

```
data = pd.read_csv('WineQT.csv')
data.head()
```

Step 2: Explore

The next step is Exploratory Data Analysis (EDA), where we analyze the data to understand its structure, distribution, and relationships. This involves generating descriptive statistics and visualizing the correlations between variables.

Some key observations:

- The target variable is wine quality, which is ordinal.
- Several features like alcohol, sulphates, and volatile acidity show moderate correlation with quality.

```
# Descriptive statistics and correlation analysis
desc_stats = data.describe()
# Correlation matrix
import seaborn as sns
import matplotlib.pyplot as plt
correlation_matrix = data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

From the heatmap, we can observe moderate positive correlations between alcohol and wine quality, and negative correlations with volatile acidity.

Step 3: Modify

In this phase, we clean the data, handle any outliers, and prepare the features for modeling. We’ll drop the unnecessary `Id` column, normalize the numerical features, and split the data into training and testing sets.

```
# Drop the 'Id' column
data_cleaned = data.drop('Id', axis=1)
# Split data into features (X) and target (y)
X = data_cleaned.drop('quality', axis=1)
y = data_cleaned['quality']
# Split into training and test sets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Step 4: Model

In this step, we train a machine learning model to predict wine quality. We’ll use a Random Forest Classifier, a robust and widely-used algorithm for classification tasks.

```
# Train a Random Forest model
from sklearn.ensemble import RandomForestClassifier
```

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train_scaled, y_train)
# Make predictions on the test set
y_pred = model.predict(X_test_scaled)
```

The Random Forest algorithm works by creating multiple decision trees and aggregating their predictions, providing high accuracy and reducing the likelihood of overfitting.

Step 5: Assess

Finally, we evaluate the model's performance. We'll use accuracy as the primary metric, along with a confusion matrix and classification report to examine the model's performance in predicting different quality levels.

```
# Evaluate the model
from sklearn.metrics import accuracy_score, confusion_matrix, classification_rep
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\\n{conf_matrix}')
print(f'Classification Report:\\n{class_report}')
```

Model Performance

- The model achieved an accuracy of approximately 69.87%.
- The confusion matrix shows that the model performs well in predicting quality levels 5 and 6 but struggles with less frequent classes such as 3

and 8.

- The classification report highlights that the model performs better on the most frequent labels (precision and recall for class 5 are higher compared to class 4 or 8).

Conclusion

The SEMMA methodology provides a structured approach to developing machine learning models. By following the process step-by-step, we built a Random Forest classifier that predicts wine quality with decent accuracy. There are potential improvements, such as handling class imbalance, using hyperparameter tuning, or testing other algorithms like Gradient Boosting or Support Vector Machines (SVM).

This approach can be applied to a variety of machine learning tasks, ensuring a thorough understanding of the data and a solid foundation for modeling and evaluation.

Further Reading

If you'd like to dive deeper into SEMMA or explore alternative algorithms, check out:

- SEMMA and CRISP-DM: Key Differences
- Random Forest in Machine Learning

Thank you for reading! If you found this article helpful, feel free to leave a comment or share your own experience using SEMMA for data mining and machine learning projects.

Semma Machine Learning AI Python Sklearn

Written by Satvik Atmakuri

Edit profile

0 Followers

More from Satvik Atmakuri

Satvik Atmakuri

Building an Income Prediction Model with Machine Learning: A...

Satvik Atmakuri

Predicting Heart Attack Risk Using Machine Learning: A CRISP-DM...

Introduction

Machine learning projects are all about exploration, analysis, and iteration. In this…

Sep 27

Satvik Atmakuri

Building a Fraud Detection Model Using Random Forest and KDD…

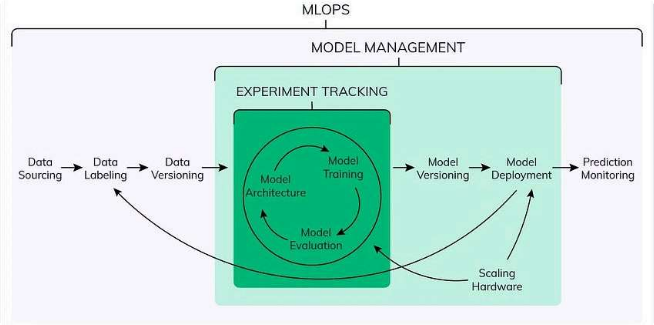
Introduction

Sep 27

Sep 27

See all from Satvik Atmakuri

Recommended from Medium



Stephen Echessa

Stacking Ensembles: Combining XGBoost, LightGBM and CatBoost...

In the ever-evolving world of machine learning, where numerous algorithms vie for...

Jul 29 24 1

Prashant Shinde in Data And Beyond

From Zero to MLOps Hero: Your First Steps in Building an MLOps...

Just a few months ago, I was where you are—thrilled by the possibilities of machine...

Sep 15 56

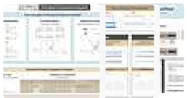
Lists

Predictive Modeling w/ Python

20 stories · 1628 saves

Coding & Development

11 stories · 880 saves

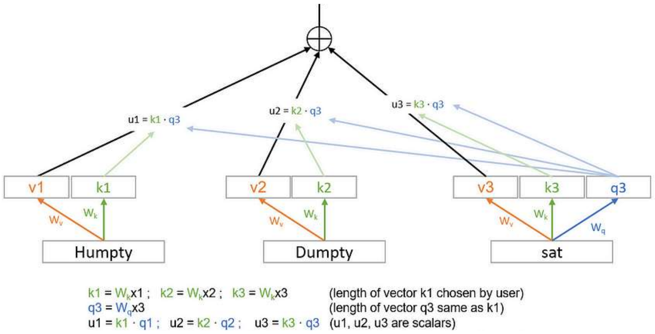


Practical Guides to Machine Learning

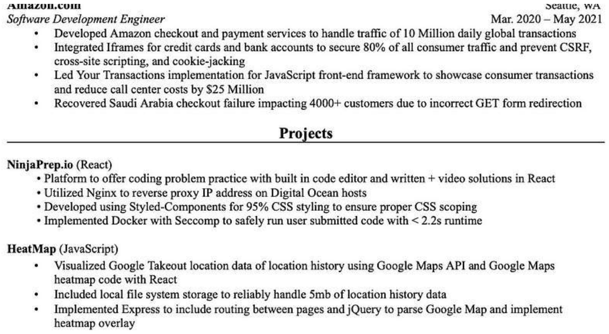
10 stories · 1989 saves

Natural Language Processing

1788 stories · 1391 saves


u1 = k1 · q3
u2 = k2 · q3
u3 = k3 · q3
v1
k1
v2
k2
v3
k3
q3
Wv
Wk
Wv
Wk
Wv
Wk
Wq
Humpty
Dumpty
sat
k1 = Wkx1 ; k2 = Wkx2 ; k3 = Wkx3 (length of vector k1 chosen by user)
q3 = Wqx3 (length of vector q3 same as k1)
u1 = k1 · q1 ; u2 = k2 · q2 ; u3 = k3 · q3 (u1, u2, u3 are scalars)

Rohit Patel in Towards Data Science



Amazon.com Seattle, WA

Software Development Engineer Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by $25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)

- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)

- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

Alexander Nguyen in Level Up Coding

Understanding LLMs from Scratch Using Middle School Math

In this article, we talk about how LLMs work, from scratch—assuming only that you know...

Oct 19 2.1K 24

The resume that got a software engineer a $300,000 job at Google.

1-page. Well-formatted.

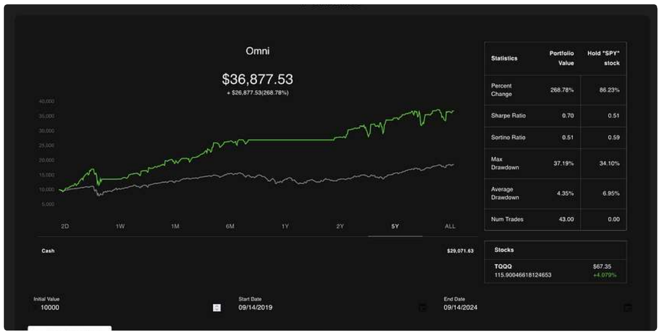
May 31 25K 478

Ayomitan Adesua in The Deep Hub

Predicting and Explaining Customer Churn: A Data Science...

How Data Science and Causal Inference Can Help Predict and Reduce Customer Churn to...

Oct 1 209 2



Statistics	Portfolio Value	Hold "SPY" stock
Percent Change	268.78%	86.23%
Sharpe Ratio	0.70	0.51
Sortino Ratio	0.51	0.59
Max Drawdown	37.19%	34.10%
Average Drawdown	4.35%	6.95%
Num Trades	43.00	0.00

Cash $29,071.63

Stocks

TQQQ $67.35
115.90046618124653 +4.079%

Initial Value 10000

Start Date 09/14/2019

End Date 09/14/2024

Austin Starks in DataDrivenInvestor

I used OpenAI's o1 model to develop a trading strategy. It is...

It literally took one try. I was shocked.

Sep 15 5.3K 138

See more recommendations