EARTHQUAKE SEVERITY PREDICTION

Rahul Bangad
University of Arizona
rahulbangad@arizona.edu

Abstract

This project develops a predictive model for earthquake severity using SVM, Linear Regression, and Random Forest, analysing a dataset of 782 entries with 19 features. The approach includes comprehensive data cleaning, exploratory data analysis (EDA), and data preprocessing. The objective is to compare the effectiveness of these machine learning models in accurately forecasting earthquake impacts. This work is vital for enhancing earthquake response strategies and mitigating risks. Additionally, the project explores the correlation between various seismic indicators and the actual severity, aiming to contribute valuable insights to seismology and disaster risk management.

1 INTRODUCTION

Earthquakes are among the most devastating natural disasters, impacting lives, infrastructure, and economies globally. Their unpredictable nature and potential for massive destruction make accurate prediction of their severity a crucial aspect of disaster management and response. Despite advancements in seismology, predicting the exact time, location, and severity of earthquakes remains a significant challenge. However, with the advent of machine learning, new possibilities have emerged for enhancing earthquake prediction and preparedness. The severity of an earthquake is typically measured by various factors, including its magnitude, depth, and the affected area's geological characteristics. These parameters, when analysed effectively, can provide insights into the potential impact of an earthquake. Machine learning models have the capability to analyse large datasets of seismic activity, extracting patterns and correlations that might be invisible to traditional analysis methods. In this project, we aim to leverage the power of machine learning to predict earthquake severity more accurately. By employing three different machine learning models — Support Vector Machine (SVM), Linear Regression, and Random Forest, we seek to understand which model best predicts the severity of earthquakes. The dataset used in this project encompasses 782 records with 19 distinct features, providing a comprehensive base for analysis. This research is not just an academic exercise; it holds significant practical implications. By improving the accuracy of earthquake severity predictions, we can enhance early warning systems, aid in disaster preparedness, and ultimately save lives. This project stands at the intersection of machine learning and geoscience, contributing to a growing body of knowledge that aims to harness the predictive power of data for the betterment of society in the face of natural disasters. Furthermore, this project also explores the integration of advanced data preprocessing techniques and feature selection methods to refine the predictive models. By doing so, it aims to tackle the complexities and variabilities inherent in seismic data, thus pushing the boundaries of what machine learning can achieve in geoscience. The ultimate goal is to provide actionable insights that can be readily utilized by authorities and disaster response teams, thereby making communities more resilient to the threat of earthquakes.

2 RELATED WORK

This project advances research in the field of predictive analytics, specifically earthquake severity prediction, while also advancing human-machine collaboration and complex problem-solving in

dynamic environments.The major aim is to improve machine learning models' prediction powers in the setting of natural disasters, an area typified by high stakes and unpredictability. This effort, inspired by a Kaggle project, aims to create a robust machine learning model capable of reliably forecasting earthquake intensity. The Kaggle project we're referring to uses a wide collection of input variables, including seismic activity, geographic location, and historical earthquake data, and validates the model using ensemble techniques and cross-validation. This approach is consistent with recent advances in the field, emphasising the importance of dependable, accurate models to aid in earthquake preparedness and reduce the impact on infrastructure and human lives. Our research aims to contribute to this evolving field by providing new insights and strategies for predicting earthquake severity using advanced machine learning techniques.

3 PROCEDURE

3.1 Dataset

The dataset is a reference from Kaggle:
Source: `https://www.kaggle.com/datasets/warcoder/earthquake-dataset`

3.2 Attributes Description

Datasets contain records of 782 earthquakes from 1/1/2001 to 1/1/2023. The meaning of all columns is as follows:

- **Title:** title name given to the earthquake.
- **Magnitude:** The magnitude of the earthquake.
- **Date_Time:** Date and time of the earthquake.
- **CDI:** The maximum reported intensity for the event range.
- **MMI:** The maximum estimated instrumental intensity for the event.
- **Alert:** The alert level - "green", "yellow", "orange", and "red".
- **Tsunami:** "1" for events in oceanic regions and "0" otherwise.
- **Sig:** A number describing how significant the event is.
- **Net:** The ID of a data contributor.
- **NST:** The total number of seismic stations used to determine earthquake location.
- **Dmin:** Horizontal distance from the epicenter to the nearest station.
- **Gap:** The largest azimuthal gap between azimuthally adjacent stations.
- **MagType:** The method or algorithm used to calculate the preferred magnitude.
- **Depth:** The depth where the earthquake begins to rupture.
- **Latitude / Longitude:** Coordinate system for Earth's surface location.
- **Location:** Location within the country.
- **Continent:** Continent of the earthquake-hit country.
- **Country:** Affected country.

3.3 Exploratory Data Analysis

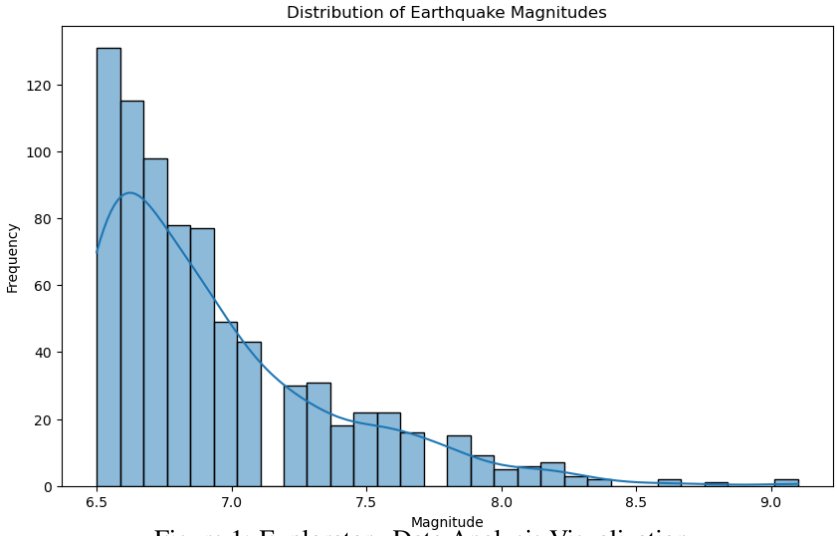


Figure 1: Exploratory Data Analysis Visualization

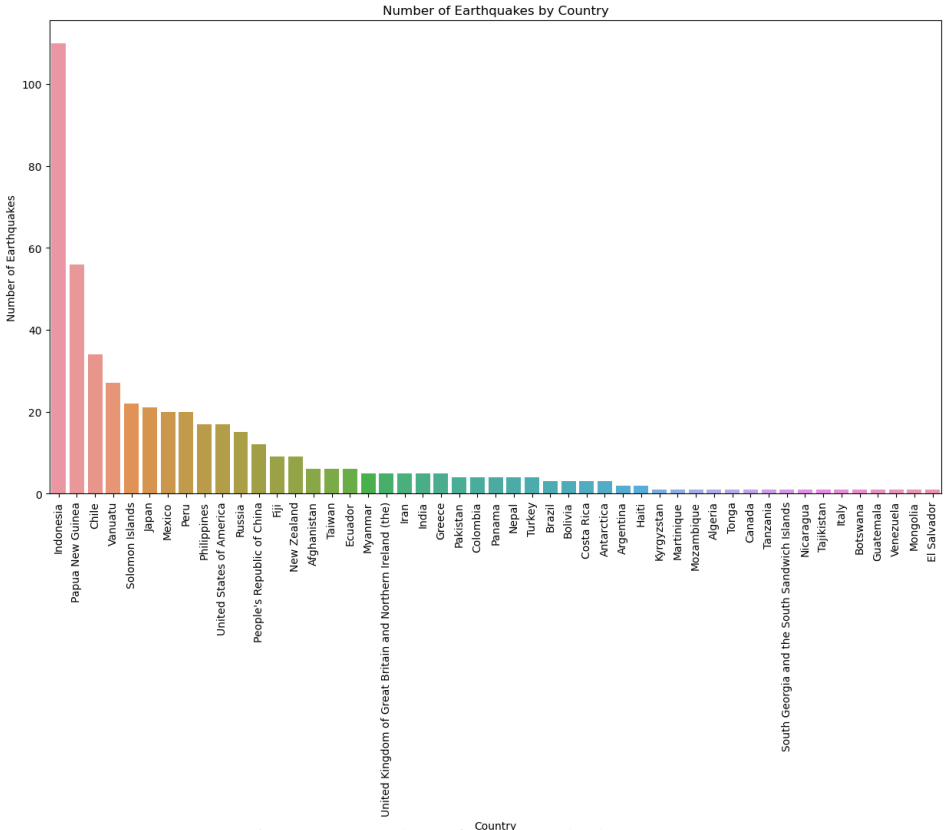


Figure 2: Number of Earthquake by country

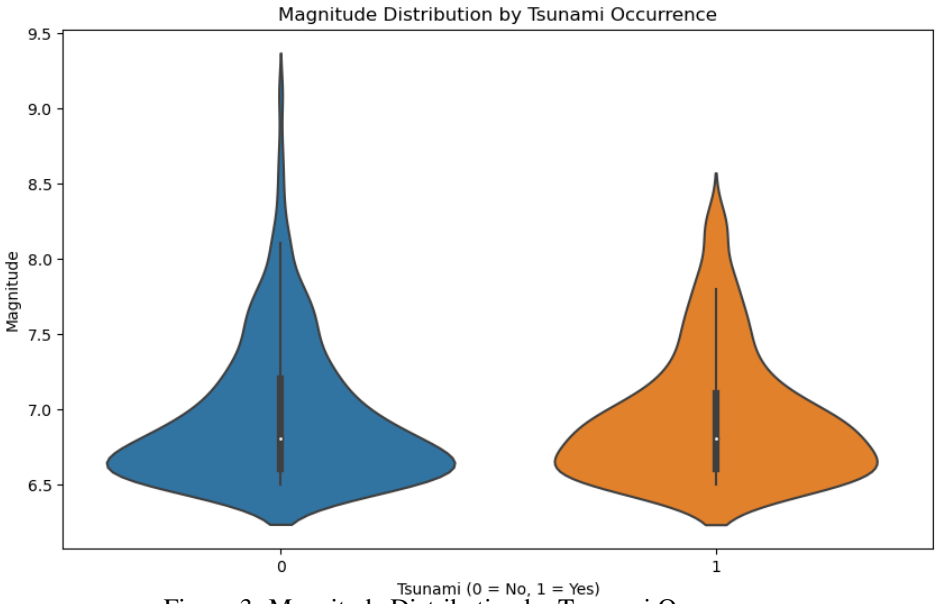


Figure 3: Magnitude Distribution by Tsunami Occurrence

3.4 Machine Learning Model Used

Logistic Regression: This model is a fundamental technique in statistical modelling, assuming a linear relationship between dependent and independent variables. It operates by fitting a best-fit line or regression line through the data. Linear Regression is versatile, being applicable in simple (single variable) and complex (multiple variables) scenarios. In the context of this project, it helps to understand the linear relationships between various seismic features and the severity of earthquakes. The equation for Logistic Regression is given as:

$$P(Y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)}}$$

Support Vector Machine (SVM): SVM is a powerful machine learning algorithm used for both classification and regression tasks. It works by finding the best hyperplane that separates the data into classes or predicts a continuous output. In earthquake severity prediction, SVM can model complex nonlinear relationships between the features and the target variable, potentially capturing intricate patterns in seismic data. The equation for SVM is given as:

$$f(x) = \text{sign}(w \cdot x + b)$$

Random Forest Classification: This is an ensemble learning method, particularly effective for regression and classification problems. Random Forest operates by constructing a multitude of decision trees during training and outputting the average prediction of the individual trees for regression tasks. Its strength lies in its ability to handle large datasets with higher dimensionality and provides estimates of feature importance, which can be invaluable in understanding the contributing factors to earthquake severity. The equation for Random Forest Classification is given as:

$$f(x) = \text{mode}(f_1(x), f_2(x), ..., f_N(x))$$

4 Data Pre-Processing

In our Earthquake Severity Prediction project, the dataset, comprising 782 records with 19 variables, underwent thorough data preprocessing to ensure the reliability and accuracy of the machine learning models. Key preprocessing steps included:

4.1 Missing Value Analysis

In [4]:
```
# Checking for missing values in each column
missing_values = dataset.isnull().sum()
missing_values_percentage = (missing_values / len(dataset)) * 100

missing_data_info = pd.DataFrame({
    'Missing Values': missing_values,
    'Percentage (%)': missing_values_percentage
})
missing_data_info.sort_values(by="Missing Values", ascending=False)
```

Out[4]:

	Missing Values	Percentage (%)
continent	576	73.657289
alert	367	46.930946
country	298	38.107417
location	5	0.639386
dmin	0	0.000000
longitude	0	0.000000
latitude	0	0.000000
depth	0	0.000000
magType	0	0.000000
gap	0	0.000000
title	0	0.000000
magnitude	0	0.000000
net	0	0.000000
sig	0	0.000000
tsunami	0	0.000000
mmi	0	0.000000
cdi	0	0.000000
date_time	0	0.000000
nst	0	0.000000

4.2 Duplicate Record Check

Check for Duplicates

In [8]:
```
# Check for duplicate rows
duplicates = dataset[dataset.duplicated()]

# Print the number of duplicates and the first few duplicate rows
print("Number of duplicate rows:", duplicates.shape[0])
print(duplicates.head())
```

```
Number of duplicate rows: 0
Empty DataFrame
Columns: [title, magnitude, date_time, cdi, mmi, alert, tsunami, sig, net, nst, dmin, gap, magType, depth, latitud
e, longitude, location, continent, country]
Index: []
```

In [9]:
```
# There is no Duplicate Rows in the Dataset
```

4.3 Correlation Analysis

Correlation Matrix

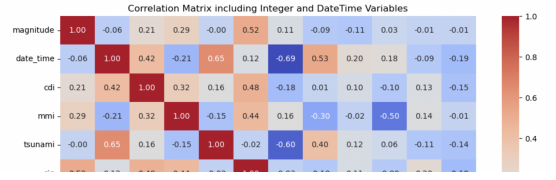
In [12]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Convert date_time to datetime format
dataset['date_time'] = pd.to_datetime(dataset['date_time'], errors='coerce')

# Select numeric and datetime columns
numeric_and_datetime_cols = dataset.select_dtypes(include=[np.number, 'datetime64']).columns

# Compute the correlation matrix including datetime converted to ordinal
corr_matrix = dataset[numeric_and_datetime_cols].apply(lambda x: x if np.issubdtype(x.dtype, np.number) else x.map(p

# Plot the correlation matrix
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Correlation Matrix including Integer and DateTime Variables")
plt.show()
```



4.4 Chi-Square Test for Categorical Values

Chi Square Test

In [16]:
```
from scipy.stats import chi2_contingency

# Convert 'magnitude' into a categorical variable for the chi-square test
# This can be done by binning the magnitude into ranges
magnitude_bins = pd.qcut(dataset['magnitude'], q=4, labels=False, duplicates='drop')

# Perform Chi-square tests for 'country', 'continent', 'location', 'magType', 'net', and 'alert'
categorical_columns = ['country', 'continent', 'location', 'magType', 'net', 'alert']
chi_square_results = {}

for col in categorical_columns:
    contingency_table = pd.crosstab(magnitude_bins, dataset[col])
    chi2, p, dof, expected = chi2_contingency(contingency_table)
    chi_square_results[col] = {'Chi-Square Statistic': chi2, 'p-value': p}

chi_square_results
```

Out[16]:
```
{'country': {'Chi-Square Statistic': 153.02819649852083,
  'p-value': 0.287617262135679},
 'continent': {'Chi-Square Statistic': 20.24529103519705,
  'p-value': 0.16272360266318026},
 'location': {'Chi-Square Statistic': 1300.9504357367991,
  'p-value': 0.09723120430473117},
 'magType': {'Chi-Square Statistic': 39.987622249968055,
  'p-value': 0.021452360785815697},
 'net': {'Chi-Square Statistic': 50.294453186272605,
  'p-value': 0.011556099000382939},
 'alert': {'Chi-Square Statistic': 29.451833366086046,
  'p-value': 0.0005435500921643 23}}
```

In [17]:
```
del dataset["country"]
```

4.5 Label Encoding

label Encoding

In [23]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
alert_le = LabelEncoder()
magtype_le = LabelEncoder()
net_le = LabelEncoder()
dataset["alert"]=alert_le.fit_transform(dataset["alert"])
dataset["magType"]=magtype_le.fit_transform(dataset["magType"])
dataset["net"]=net_le.fit_transform(dataset["net"])
dataset
```

Out[23]:

	magnitude	cdi	mmi	alert	tsunami	sig	net	nst	dmin	gap	magType	depth	latitude	longitude
0	7.0	8	7	0	1	768	9	117	0.509	17.0	8	14.000	-9.7963	159.596
1	6.9	4	4	0	0	735	9	99	2.229	34.0	8	25.000	-4.9559	100.738
2	7.0	3	3	0	1	755	9	147	3.125	18.0	8	579.000	-20.0508	-178.346
3	7.3	5	5	0	1	833	9	149	1.865	21.0	8	37.000	-19.2918	-172.129
4	6.6	0	2	0	1	670	9	131	4.998	27.0	8	624.464	-25.5948	178.278
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
777	7.7	0	8	2	0	912	9	427	0.000	0.0	7	60.000	13.0490	-88.660
778	6.9	5	7	2	0	745	0	0	0.000	0.0	5	36.400	56.7744	-153.281
779	7.1	0	7	2	0	776	9	372	0.000	0.0	6	103.000	-14.9280	167.170
780	6.8	0	5	2	0	711	9	64	0.000	0.0	7	33.000	6.6310	126.899
781	7.5	0	7	2	0	865	9	324	0.000	0.0	7	33.000	6.8980	126.579

782 rows × 14 columns

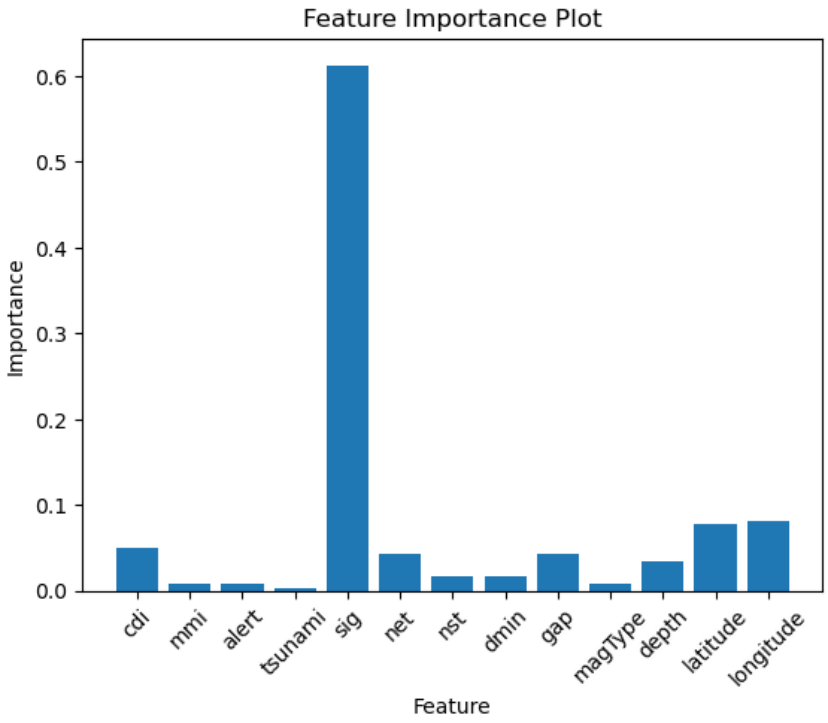
4.6 Feature Selection and Removal

Removing the following features with low importance:

- country
- continent
- location
- title
- Date_Time

This is done with the help of correlation matrix and p-values.

4.7 Summary figure



5 Model Training

Based on how well they fit regression tasks, models like Random Forest, SVM, and Linear Regression are chosen. The training dataset is used to train each model.

6 Data Splitting

Using the standard 80/20 split method, the dataset is initially divided into training and test sets. This indicates that while 20 percent of the data is set aside for testing, the remaining 80 percent is utilized to train the models, enabling them to gain knowledge from a sizable amount of the dataset. This test set serves as fresh, untested data to assess the generalization capabilities and performance of the models.

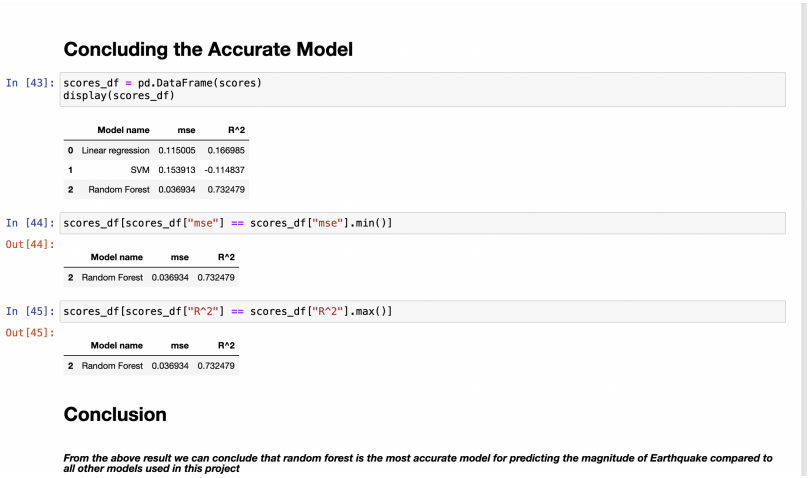
7 Performance Evaluation

The test set is used to validate the models . The models' performance is evaluated. Although Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are often used metrics, Mean Squared Error (MSE) and the R-squared value are especially used for evaluation in this project.

Models	Linear Regression	SVM	Random Forest
R^2	0.17	-0.11	0.7324789111372308
MSE	0.12	0.15	0.03693351719745237

Table 1: Performance Evaluation of All models

8 Results



Concluding the Accurate Model

```
In [43]: scores_df = pd.DataFrame(scores)
         display(scores_df)
```

	Model name	mse	R^2
0	Linear regression	0.115005	0.166985
1	SVM	0.153913	-0.114837
2	Random Forest	0.036934	0.732479

```
In [44]: scores_df[scores_df["mse"] == scores_df["mse"].min()]
```

Out[44]:

	Model name	mse	R^2
2	Random Forest	0.036934	0.732479

```
In [45]: scores_df[scores_df["R^2"] == scores_df["R^2"].max()]
```

Out[45]:

	Model name	mse	R^2
2	Random Forest	0.036934	0.732479

Conclusion

From the above result we can conclude that random forest is the most accurate model for predicting the magnitude of Earthquake compared to all other models used in this project

9 Conclusion

In this project, we aimed to develop a model for predicting earthquake severity, a critical factor in disaster preparedness and response. We implemented and compared three machine learning models: SVM, Linear Regression, and Random Forest, to determine the most effective approach for this task. Our comparison criteria were based on two key performance metrics: Mean Squared Error (MSE) and R-squared (R^2).

The results indicated that the Random Forest model outperformed SVM and Linear Regression in predicting earthquake severity. Random Forest achieved the lowest MSE, implying its predictions were closest to the actual data, and the highest R^2 value, indicating a strong fit to the data. This superiority can be attributed to Random Forest's ability to handle complex datasets and capture nonlinear relationships more effectively than SVM and Linear Regression.

The success of the Random Forest model in this project highlights its potential as a reliable tool in seismology for predicting earthquake impacts. Such predictive capabilities are essential for enhancing early warning systems, aiding in disaster preparedness, and potentially saving lives by allowing for timely and effective responses.

Future research directions could include refining the Random Forest model further, incorporating more diverse and extensive seismic data, and exploring advanced machine learning techniques. Additionally, integrating real-time data and deploying the model in a live environment could significantly impact earthquake preparedness and risk mitigation strategies. The ultimate goal remains to develop increasingly accurate and reliable predictive models, contributing to the broader field of disaster risk management and response planning.

References

[1] https://www.kaggle.com/code/shrinivas2402/earthquake-prediction/notebook#2.-Developing-the-Model

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html

[3] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.crosstab.html

[4] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.qcut.html

[5] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[6] https://realpython.com/linear-regression-in-python/