**1. How to import csv?**

*import pandas as pd*

*cr_data = pd.read_csv("credit_risk_dataset.csv")*

**2. How do you select columns from dataframe?**

*Selecting the 'description' column from 'reviews' dataframe*

*reviews['description']*

**3. How do you select rows from dataframe?**

*Selecting the first row from 'reviews' dataframe*

*reviews.iloc[0]*

**4. How do you select both rows and columns from dataframe?**

*Selecting the first row of 'description' column from 'reviews' dataframe*

*reviews['description'].iloc[0]*

**5. How do you select rows based on indices?**

*Selecting rows 1, 2, 3, 5 and 8 from 'reviews' dataframe*

*indices = [1, 2, 3, 5, 8]*

*sample_reviews = reviews.loc[indices]*

**6. How do you find the median value?**

*Finding the median of 'points' column from 'reviews' dataframe*

*reviews['points'].median()*

**6. How do you find the unique values?**

*Finding all the unique countries in 'country' column from 'reviews' dataframe*

*reviews['country'].unique()*

**7. How do you find count of unique values?**

*Finding the count of unique countries in 'country' column from 'reviews' dataframe*

*reviews['country'].value_counts()*

**48. How to get the data type of a particular variable?**

*Get the data type of 'points' column from 'reviews' dataframe*

*reviews['points'].dtype*

**49. How do you drop columns?**

*Dropping columns 'points' and 'country' from 'reviews' dataframe*

*reviews.drop(['points', 'country'], axis=1, inplace=True)*

**51. How do you rename a column?**

*Rename 'region_1' as 'region' and 'region_2' as 'locale'*

*reviews.rename(columns=dict(region_1='region', region_2='locale'))*

**52. How do you sort a dataframe based on a variable?**

*Sorting 'region_1' in descending order*

*reviews['region_1'].sort_values(ascending=False)*

Visualization — 8 questions

**53. How do you plot a line chart?**

*import seaborn as sns*

*sns.lineplot(data=loan_amnt)*

**54. How do you plot a bar chart?**

*import seaborn as sns*

*sns.barplot(x=cr_data[‘cb_person_default_on_file’], y=cr_data[‘loan_int_rate’])*

**56. How do you plot scatter plot?**

*import seaborn as sns*

*sns.scatterplot(x=cr_data[‘loan_amnt’], y=cr_data[‘person_income’])*

**57. How do you plot distribution chart?**

*import seaborn as sns*

*sns.distplot(a=cr_data[‘person_income’], label=”person_income”, kde=False)*

**58. How do you add x-label and y-label to the chart?**

*import matplotlib.pyplot as plt*

*plt.xlabel(“cred_hist_length”)*

*plt.ylabel(“loan_amnt”)*

**59. How do you add title to the chart?**

*import matplotlib.pyplot as plt*

*plt.title("Average int_rate")*

**60. How do you add legend to chart?**

*import matplotlib.pyplot as plt*

*plt.legend()*

Data Cleaning — 5 questions

**61. How do you identify missing values?**

*The function used to identify the missing value is through .isnull()*

*The code below gives the total number of missing data points in the data frame*

*missing_values_count = sf_permits.isnull().sum()*

**62. How do you impute missing values value imputation?**

*Replace missing values with zero / mean*

*df['income'].fillna(0)*

*df['income'] = df['income'].fillna((df['income'].mean()))*

**63. What is scaling of data?**

***Scaling convert the data using the formula = (value — min value) / (max value — min value)***

*from sklearn.preprocessing import MinMaxScaler*

*scaler = MinMaxScaler()*

*original_data = pd.DataFrame(kickstarters_2017['usd_goal_real'])*

*scaled_data = pd.DataFrame(scaler.fit_transform(original_data))*

***Original data***

*Minimum value: 0.01*

*Maximum value: 166361390.71*

***Scaled data***

*Minimum value: 0.0*

*Maximum value: 1.0*

**64. What is normalizing of data?**

***Scaling convert the data using the formula = (value — mean) / standard deviation***

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

original_data = pd.DataFrame(kickstarters_2017['usd_goal_real'])

scaled_data = pd.DataFrame(scaler.fit_transform(original_data))
```

***Original data***

*Minimum value: 0.01*

*Maximum value: 166361390.71*

***Scaled data***

*Minimum value: -0.10*

*Maximum value: 212.57*

**65. How do you treat dates in python?**

*To convert dates from String to Date*

*import datetime*

*import pandas as pd*

*df['Date_parsed'] = pd.to_datetime(df['Date'], format="%m/%d/%Y")*

Machine Learning — 15 questions

**66. What is logistic regression?**

*Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.*

**67. What is the syntax for logistic regression?**

***Library:*** *sklearn.linear_model.LogisticRegression*

***Define model:*** *lr = LogisticRegression()*

***Fit model:*** *model = lr.fit(x, y)*

***Predictions:*** *pred = model.predict_proba(test)*

**68. How do you split the data in train / test?**

***Library:*** *sklearn.model_selection.train_test_split*

***Syntax:*** *X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)*

**69. What is decision tree?**

*Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.*

**70. What is the syntax for decision tree classifier?**

***Library:*** *sklearn.tree.DecisionTreeClassifier*

***Define model:*** *dtc = DecisionTreeClassifier()*

***Fit model:*** *model = dtc.fit(x, y)*

***Predictions:*** *pred = model.predict_proba(test)*

**71. What is random forest?**

*Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.*

**72. What is the syntax for random forest classifier?**

***Library:*** *sklearn.ensemble.RandomForestClassifier*

***Define model:*** *rfc = RandomForestClassifier()*

***Fit model:*** *model = rfc.fit(x, y)*

***Predictions:*** *pred = model.predict_proba(test)*

**77. How do you treat categorical variables?**

*Replace categorical variables with the average of target for each category*

| Gender | Y | Gender_Y |
|---|---|---|
| M | 1 | 0.33 |
| F | 1 | 0.67 |
| M | 0 | 0.33 |
| F | 0 | 0.67 |
| M | 0 | 0.33 |
| F | 1 | 0.67 |

*One hot encoding*

| Gender | Y | Gender_M | Gender_F |
|---|---|---|---|
| M | 1 | 1 | 0 |
| F | 1 | 0 | 1 |
| M | 0 | 1 | 0 |
| F | 0 | 0 | 1 |
| M | 0 | 1 | 0 |
| F | 1 | 0 | 1 |

**78. How do you treat missing values?**

***Drop rows having missing values***

*DataFrame.dropna(axis=0, how='any', inplace=True)*

***Drop columns***

*DataFrame.dropna(axis=1, how='any', inplace=True)*

***Replace missing values with zero / mean***

*df['income'].fillna(0)*

*df['income'] = df['income'].fillna((df['income'].mean()))*

**79. How do you treat outliers?**

*Inter quartile range is used to identify the outliers.*

*Q1 = df['income'].quantile(0.25)*

*Q3 = df['income'].quantile(0.75)*

*IQR = Q3 — Q1*

*df = df[(df['income'] >= (Q1–1.5 * IQR)) & (df['income'] <= (Q3 + 1.5 * IQR))]*

**80. What is bias / variance trade off?**

***Definition***

*The Bias-Variance Trade off is relevant for supervised machine learning, specifically for predictive modelling. It's a way to diagnose the performance of an algorithm by breaking down its prediction error.*

***Error from Bias***

*Bias is the difference between your model's expected predictions and the true values.*

*This is known as under-fitting.*

*Does not improve with collecting more data points.*

***Error from Variance***

*Variance refers to your algorithm's sensitivity to specific sets of training data.*

*This is known as over-fitting.*

*Improves with collecting more data points.*

**Q Which all Python libraries have you used for visualization?**

**Matplotlib:** It is the standard data visualization library useful to generate two-dimensional graphs. It helps plot histograms, pie charts, bar or column graphs, scatterplots, and non-Cartesian coordinates graphs. Many libraries are built on top of Matplotlib, and its functions are used in the backend. Also, it is extensively used to create the axes and the layout for plotting.
**Seaborn:** Based on Matplotlib, Seaborn is a data visualization library in Python. It works really well for Numpy and Pandas. It provides a high-level interface for drawing attractive and informative statistical graphics.

**Q. List some of the categorical, distribution plots.**

**Distribution Plots:**

- displot: Figure-level interface for drawing distribution plots onto a Facet Grid.
- histplot: Plots univariate or bivariate histograms to show distributions of datasets.
- kdeplot: Plots univariate or bivariate distributions using kernel density estimation.

**Categorical Plots:**

- Catplot: Figure-level interface for drawing categorical plots onto a FacetGrid.
- Stripplot: Draws a scatter plot where one variable is categorical.
- Swarmplot: Plots a categorical scatter plot with non-overlapping points
- Boxplot: Plots a box plot to show distributions with respect to categories.
- Violinplot: Plots a combination of boxplot and kernel density estimate.
- Boxenplot: Draws an enhanced box plot for larger datasets.
- Pointplot: Shows the point estimates and confidence intervals using scatter plot glyphs.
- Barplot: Shows the point estimates and confidence intervals as rectangular bars.
- Countplot: Show the counts of observations in each categorical bin using bars.

**Q. What is a scatter plot?**

A scatter plot is a two-dimensional data visualization that illustrates the relationship between observations of two different variables. One is plotted along the x-axis, and the other is plotted against the y-axis.

- QWhat is the difference between stripplot() and swarmplot()?

- Strip Plot: It plots a scatter plot where one variable is categorical. A stripplot() can be drawn on its own, but it is also a good complement to a box plot in cases where one wants to show all observations and some representation of the underlying distribution.
- **Swarm Plot:** It is also used to plot a categorical scatter plot; however, it is with non-overlapping points here. swarmplot() is similar to stripplot(), but the points are adjusted (only along the categorical axis) not to overlap. It is a better representation of the distribution of values; however, it does not scale well to large numbers of observations. This style of the plot is sometimes called a "beeswarm."

**Q What is the purpose of density plot or kde plot? Where are these used?**

- A density plot visualizes the distribution of data over a continuous interval or time period. A variation of the histogram density plot uses a kernel for smoothing the plot values. This allows smooth noise. The peaks of a density plot illustrate where the values are concentrated over the interval.

**Q. How is violinplot() different from boxplot()?**

A box plot (or box-and-whisker plot) shows the distribution of quantitative data that helps compare between variables or across levels of a categorical variable. A box plot is the visual representation of the statistical five-number summary of a given data set.

The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be "outliers" using a method that is a function of the inter-quartile range.

A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared.

Unlike a box plot, where all of the plot components correspond to actual data points, the violin plot features a kernel density estimation of the underlying distribution.

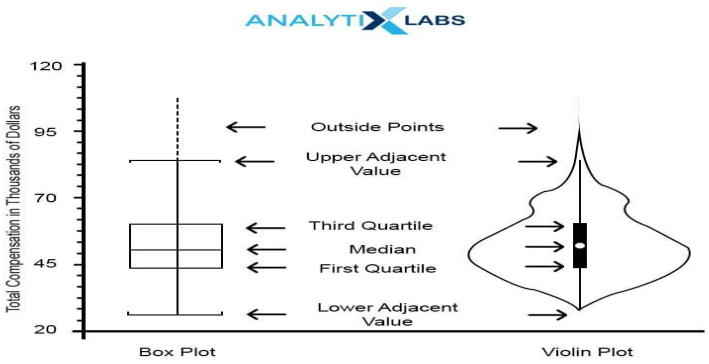



*Figure 1. Common Components of Box Plot and Violin Plot. Total compensation for all academic ranks.*

**1. What do you mean by feature engineering?**

Feature engineering is the process of creating a new feature, transforming a feature, and encoding a feature. Sometimes we also use the domain knowledge to generate new features. It prepares the data that easily input to the model and improves model performance.

2**. What do you mean by feature scaling or data normalization? Explain some techniques for feature scaling?**

In feature scaling, we change the scale of features to convert it into the same range such as (-1,1) or (0,1). This process is also known as data normalization. There are various methods for scaling or normalizing data such as min-max normalization, z-score normalization(standard scaler), and Robust scaler.

Min-max normalization performs a linear transformation on the original data and converts it into a given minimum and maximum range.

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

z-score normalization (or standard scaler) normalizes the data based on the mean and standard deviation.

$$z = \frac{x - \mu}{\sigma}$$

**3. What are the missing values? and How do you handle missing values?**

In the data cleaning process, we can see there are lots of values that are missing or not filled or collected during the survey. WE can handle such missing values using the following methods:

- Ignoring such rows or dropping such records.
- Fill values with mean, mode, and median.
- you can also fill values using mean but for different classes, different means can be used.
- You can also fill the most probable value using regression, Bayesian formula, or decision tree, KNN, and Prebuilt imputing libraries.
- Fill with a constant value.
- Fill values manually.

**4. What is an outlier? How you detect outliers in your data?**

Outliers are abnormal observations that deviate from the norm. Outliers do not fit in the normal behavior of the data. We can detect outliers using the following methods:

- Box plot
- Scatter plot
- Histogram
- Standard Deviation or Z-Score
- Inter Quartile Range(IQR): values out of 1.5 times of IQR
- Percentile: you can select 99 percentile values and remove the
- DBSCAN
- Isolation forest, One-Class SVM

**5. How you treat the outliers in your data?**

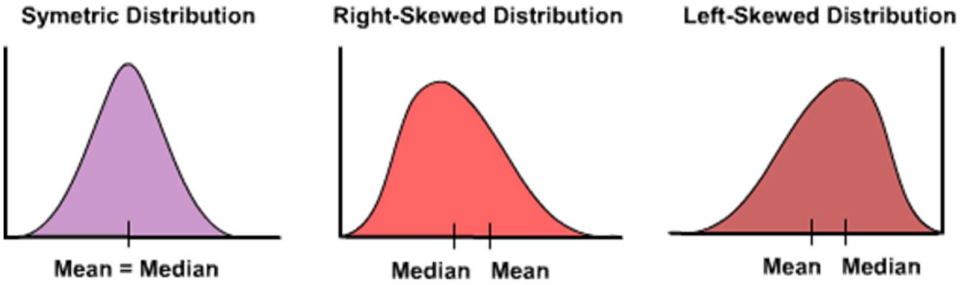
We can treat the outlier by removing it from the data. After detecting outliers we can filter the outliers using Z-score, Percentile, or 1.5 times of IQR.

**6. What do you mean by feature splitting?**

A feature split is an approach to generate a few other features from the existing one to improve the model performance. for example, splitting names into first and last names.

**7. How do you handle the skewed data columns?**

We can handle skewed data using transformations such as square, log, square, square root, reciprocal (1/x), and Box-Cox Transform.




https://medium.com/@TheDataGyan/day-8-data-transformation-skewness-normalization-and-much-more-4c144d370e55

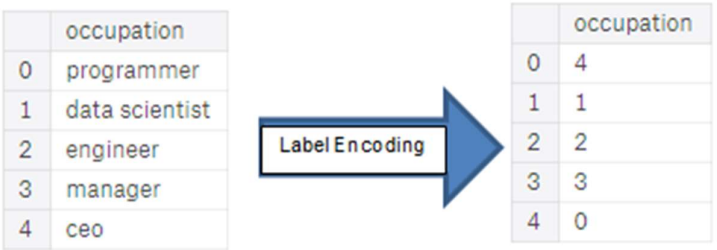
**8. What do you mean by data transformation?**

Data transformation consolidated or aggregate your data columns. It may impact your machine learning model performance. There are the following strategies to transform data:

- Data Smoothing using binning, or clustering
- Aggregate your data
- Scale or normalize your data for example scaling income column between 0 and 1 range.
- Discretize your data for example convert continuous age column into the range 0–10, 11–20, and so on. Or we can also convert the continuous age column into conceptual labels such as youth, middle, and senior.

**4. What are the label and ordinal encodings?**

Label encoding is a kind of integer encoding. Here, each unique category value is replaced with an integer value so that the machine can understand.




https://huntdatascience.wordpress.com/2019/07/26/encoding-categorical-variables/

Ordinal encodings is a label encoding with an order in the encoded values.

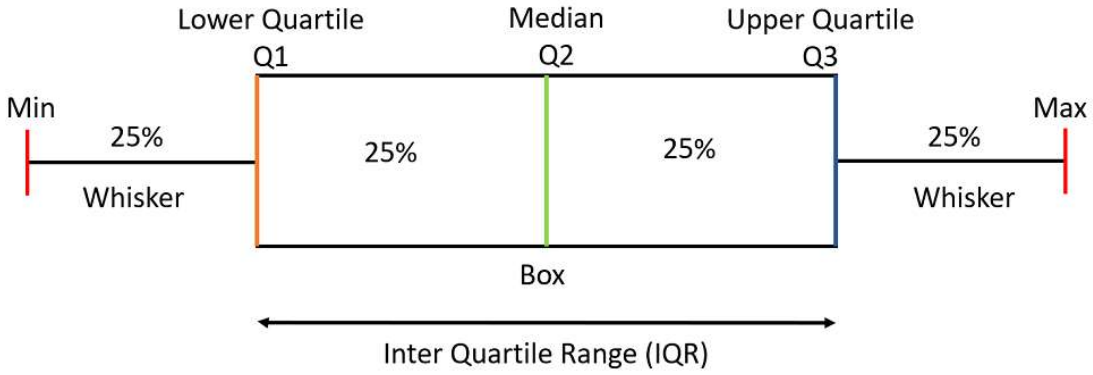
**15. Explain one-hot encoding.**

One hot encoding is used to encode the categorical column. It replaces a categorical column with its labels and fills values either 0 or 1. For example, you can see the “color” column, there are 3 categories such as red, yellow, and green. 3 categories labeled with binary values.

| Color |
|---|
| Red |
| Red |
| Yellow |
| Green |
| Yellow |

| Red | Yellow | Green |
|---|---|---|
| 1 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |

**4: What are the five statistical measures represented in a boxplot?**




*Boxplot with its statistical measures*

- Left Whisker – This statistical measure is calculated by subtracting 1.5 times IQR(Inter Quartile Range) from Q1.
    - IQR = Q3-Q1
    - Left Whisker = Q1-1.5*IQR
- Q1 – This is also known as the 25 percentile.
- Q2 – This is the median of the data or 50 percentile.
- Q3 – This is also known as 75 percentile
- Right Whisker – This statistical measure is calculated by adding 1.5 times of IQR(Inter Quartile Range) in Q3.
    - Right Whisker = Q3 + 1.5*IQR

**15: Why do we perform normalization?**

To achieve stable and fast training of the model we use normalization techniques to bring all the features to a certain scale or range of values. If we do not perform normalization then there are chances that the gradient will not converge to the global or local minima and end up oscillating back and forth. Read more about it here.

**16: What is the difference between precision and recall?**
Precision is simply the ratio between the true positives(TP) and all the positive examples (TP+FP) predicted by the model. In other words, precision measures how many of the predicted positive examples are actually true positives. It is a measure of the model's ability to avoid false positives and make accurate positive predictions.

But in the case of a recall, we calculate the ratio of true positives (TP) and the total number of examples (TP+FN) that actually fall in the positive class. recall measures how many of the actual positive examples are correctly identified by the model. It is a measure of the model's ability to avoid false negatives and identify all positive examples correctly.

**28: Explain the difference between Normalization and Standardization.**
The difference between Normalization and Standardization is as follows:

| Normalization | Standardization |
|---|---|
| It uses the minimum and maximum values of features for scaling the data. | It uses the mean and standard deviation for scaling. |
| After normalization results are in the range of [0, 1] | After standardization, the results are not bounded to any certain range. |
| Formula: (x-x_min)/(x_max – x_min) | Formula: (x-x_mean)/standard_deviation of X |
| It is also known as Scaling Normalization | It is also known as Z-Score Normalization. |

**45: How can you conclude about the model's performance using the confusion matrix?**
confusion matrix summarizes the performance of a classification model. In a confusion matrix, we get four types of output (in case of a binary classification problem) which are TP, TN, FP, and FN. As we know that there are two diagonals possible in a square, and one of these two diagonals represents the numbers for which our model's prediction and the true labels are the same. Our target is also to maximize the values along these diagonals. From the confusion matrix, we can calculate various evaluation metrics like accuracy, precision, recall, F1 score, etc.

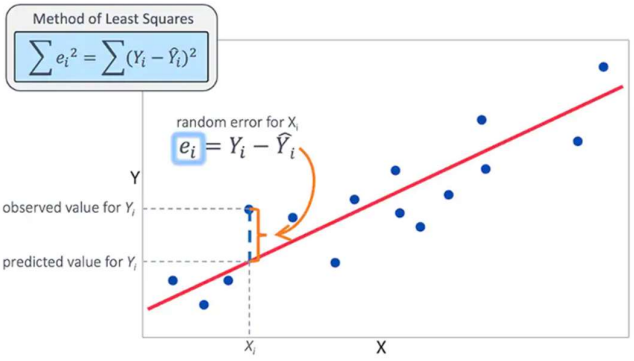
**Q1:** What is *Linear Regression*?
**Linear Regression** is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a *continuous range*, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).

Q2 How is the *Error* calculated in a Linear Regression model?
**Answer**

1. Measuring the distance of the observed *y-values* from the predicted *y-values* at each value of *x*.
2. Squaring each of these distances.
3. Calculating the *mean* of each of the squared distances.

MSE = (1/n) * Σ(actual – forecast)2

4. The smaller the **Mean Squared Error**, the closer you are to finding the *line of best fit*
5. How *bad* or *good* is this final value always depends on the context of the problem, but the main goal is that its value is so minimal as possible.




**Q6:** What is the difference between *Mean Absolute Error (MAE)* vs *Mean Squared Error (MSE)*?

**Answer**

- The **Mean Squared Error** measures the variance of the residuals and is used when we want to punish the outliers in the dataset

- The **Mean Absolute Error** measures the average of the residuals in the dataset. Is used when we don't want outliers to play a big role. It can also be useful if we know that our distribution is multimodal, and it's desirable to have predictions at one of the modes, rather than at the mean of them.

**Q7:** Compare *Linear Regression* and *Decision Trees*
**Answer**

- **Linear regression** is used to predict *continuous* outputs where there is a linear relationship between the features of the dataset and the output variable.
- **Decision trees** work by splitting the dataset, in a tree-like structure, into smaller and smaller subsets and make predictions based on which subset the new example falls into.

- **Linear regression** is used for *regression* problems where it predicts something with infinite possible answers such as the price of a house.
- **Decision trees** can be used to predict both *regression* and *classification* problems.

- **Linear regression** is prone to *underfitting* the data. Switching to *polynomial regression* will sometimes help in countering underfitting.
- **Decision trees** are prone to *overfit* the data. *Pruning* helps with the overfitting problem.

**Q15:** What is the difference between *Linear Regression* and *Logistic Regression*?

**Linear regression output as probabilities** In linear regression, the outcome (dependent variable) is continuous. It can have any one of an infinite number of possible values. In logistic regression, the outcome (dependent variable) has only a limited number of possible values.