OLD DOMINION UNIVERSITY

Question 1:

Intent : Learn the basics of how web applications are built and what technologies could be used for web applications.

Suppose you were given to build a web application for https://www.boats.com/ from scratch as a startup . Imagine you would be solely responsible to build it's frontend, backend, design database, setup media storage, deployment, ci/cd automation and everything else. Thus, do some research on your part and write down your response to the following questions.

a. Which backend framework would you prefer to use and why?

Ans.: Backend Framework: Django

- Django is an open source, high level backend framework which is easily scalable and contains all the necessary built in functions that a web developer requires.
- It is based on the DRY(Don't Repeat Yourself) code principle which focuses on reusing the code which makes it faster and efficient.
- It uses MVC(Model View Control) for logical grouping. It uses the python features providing the user to perform CRUD(Create, Read, Update, Delete).
- Django has secure authentication so that no other user can easily break through and damage the application.
- It also provides easy integration with third party applications. It has a very huge community support.
- It follows MVT design pattern(Model View Template).
- The latest version of Django is 4.0.3 which was introduced in 2022.
- The popular applications where Django is being used are Google, YouTube, Instagram, Spotify, Pinterest, DropBox, Mozilla and National Geographic.

b. Which frontend framework would you prefer to use and why?

Ans.: Frontend Framework: React Framework

- React is a JavaScript based UI library which provides many tools for developers which helps them build web pages faster, easier and in a standardized way.
- Developed by Facebook and is also maintained by them to create an interactive and reusable UI components.
- Used for handling the presentation layer of the website.
- React creates is own virtual DOM(Document Object Model) where our components live which gives us enormous flexibility and amazing performance. It identifies the changes in DOM. This process provides DOM operations and makes updates in a very systematic manner.
- It uses a special syntax called as JSX which allows us to combine HTML with JavaScript which allows us to write a HTML in the render function without any concatenation of strings. React turns those bits of HTML into functions with JSX transformer. This feature makes me choose React JS over any other frontend framework.
- Debugging our web page can be made much easier with installing the official React chrome extension by providing a virtual DOM system.
- The main disadvantage of any other framework is they are not search engine friendly.
- Search engines generally have problem reading heavy JavaScript heavy applications where as React stands out because we can run it on the server and the virtual DOM will be sent to the browser as a regular web page.

c. Which database would you prefer to use and why?

Ans.: Database: MySQL

- MySQL stands for Structured Query Language.
- Developed and distributed by Oracle Corporation.
- MySQL is easy to use, extremely powerful, fast, secure and scalable.
- Compatible with wide range of operating systems such as UNIX, Linus, Microsoft Windows, Apple Mac OS X and so on.
- Supports both small and large applications.
- It includes data security layers that protect sensitive data from any outsider.
- It stores data in tables which is a collection of related data which is divided into rows and columns.
- The main feature which made me choose MySQL over any other data base is its space to run. It uses 1MB to run.
- Contains many self management features like automatic space expansion, auto-restart and dynamic configuration changes.

d. What would you use for version control of the codebase ?

Ans.: Version Controller: Apache Subversion

- Is an open source software version and revision control system under the Apache license.
- Version control is used to manage changes in computer files and allows easier collaboration between multiple people.
- It has a centralized model.
- There is a central repository that has the copy of the changes made in central repository.
- Is able to handle large number of binary files.
- File locking can be done for safety purpose.
- The merges made can also be tracked which allows automatic merging without the knowledge of Subversion.

e. Which platform would you prefer for media storages?

Ans.: Cloud: Azure

- The main feature for me to use this cloud space is due to its free access to students where we have the opportunity to secure multiple badges and certificates.
- It focuses on Hybrid cloud that is integrates public cloud services and private cloud services.
- Azure provides access to data centers that provide changing architecture.
- It provides data security on default.
- Azure has AI services such as Azure Machine Learning and Azure Data Bricks.
- It also provides IoT services which allows us to connect and monitor IoT devices.

f. Where would you deploy your application and which web server would you use and why

Ans.: Deployment of application: Azure DevOps

Web Server: Apache HTTP Server

- Azure DevOps ensure code quality with branch policies.
- Has many ready enterprises
- It enables team worldwide access so that people can access from any part of the world.

- Can adapt and personalize to meet our requirements by providing various entension options and APIs.

- Apache HTTP server is an open source software which includes administration control panel.
- It allows us to run multiple websites from same server. Provides various third party add-ons.
- Compatible with Linux, Windows, MacOS, Unix and so on.
- Supports programming languages like PHP, Python.
- They are available very easily on multiple websites, this makes me choose apache over any other server.

Question 2:

Intent: We would like you to learn the basics of python and data science to load a dataset, read it and perform some operations to find multiple mathematical metrics such as average, maximum, minimum and such.

Here is a dataset for autos.

https://drive.google.com/file/d/1QP21K5tiJAjt5NA7W2FxSe9Wam9tIcQ/view?usp=sharing

Flow:

1. Download this dataset.
2. Write basic python script to load csv and read it as dataframe
3. Use the dataframe to perform following:

a. Find Average price of autos (using **price** column of dataset)
b. Print the list of different possible types of **VehicleType** found in dataset
c. Calculate and print lowest **yearOfRegistration** and highest **yearOfRegistration**
d. Find and print standard deviation of column **kilometer**
e. Draw a bar graph to represent count of different type of column **brand**
f. Find out which **VehicleType** is sold minimum and maximum
g. Create a pie chart to represent different types of **gearbox** count

Ans.:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy
import stats from sklearn
import model_selection, feature_selection, linear_model from sklearn.linear_model
import LinearRegression from sklearn.preprocessing
import StandardScaler,MinMaxScaler,OneHotEncoder,OrdinalEncoder from
matplotlib
import pyplot from sklearn.linear_model
import Ridge , Lasso,RidgeCV from sklearn.metrics
import mean_squared_error, r2_score from scipy.stats
import shapirofrom statsmodels.stats.diagnostic
import normal_ad from statsmodels.stats.outliers_influence
import variance_inflation_factor
import statsmodels.api as sm from statsmodels.stats.diagnostic
import het_goldfeldquandt
import warningswarnings.filterwarnings('ignore') from sklearn.compose
import ColumnTransformer , make_column_transformer,make_column_selector
from sklearn.pipeline
import Pipeline from sklearn.feature_selection
import mutual_info_regression, SelectKBest,f_regression,RFE
df["normalized-losses"].fillna(df["normalized-losses"].astype("float").mean(),
inplace=True)
df["bore"].fillna(df["bore"].astype("float").mean(), inplace=True)
df["stroke"].fillna(df["stroke"].astype("float").mean(), inplace = True)
df["peak-rpm"].fillna(df["peak-rpm"].astype("float").mean(), inplace = True)
df['horsepower'].fillna(df['horsepower'].astype("float").mean(), inplace=True)
 category = ["make","fuel-type","aspiration","num-of-doors","body-style",
 "drive-wheels","engine-location","engine-type","num-of-cylinders","fuel-
 system"]fig, axes = plt.subplots(nrows=5, ncols=2, figsize=(14, 18))axe =
 axes.ravel()for i, category in enumerate(df_obj[category]):
 df_obj[category].value_counts().plot(kind="bar",
 ax=axe[i]).set_title(category)fig.show()fig.tight_layout()

sns.regplot(x = 'engine-size', y = 'price', data = df)plt.ylim(0,) # y axis starts from
zeroprint("correlation between engine-size and price:")df[["engine-size",
"price"]].corr()
sns.regplot(x = 'highway-mpg', y = 'price',data = df)print('correlation between
highway-mpg and price:')df[['highway-mpg', 'price']].corr()
sns.regplot(x = 'peak-rpm', y = 'price', data = df)print('correlation between peak-rpm
and price:')df[['peak-rpm','price']].corr()
print('correlation between stroke and prce :')
sns.regplot(x = 'stroke', y = 'price', data = df)
.df[['stroke','price']].corr()
stdscaler=StandardScaler()numeric_features = ["symboling","normalized-
losses","wheel-base", "length","width",
"height","curb-weight", "engine-size","bore","stroke","compression-
ratio","horsepower",
"peak-rpm","city-mpg","highway-
mpg"]num_transformed=stdscaler.fit_transform(X_numeric)num_transformed=pd.Da
taFrame(num_transformed, columns=numeric_features)num_transformed.head()
def residual(model,feature,label):
pred=model.predict(feature)
df2=pd.DataFrame({"Actual":label,"Predicted":pred})
df2["Residuals"]=df2["Actual"]-df2["Predicted"]
return df2
def linear_assumption(model, features, label):
print('Linearity Check:', '\n')

print('Checking with a scatter plot of actual vs. predicted.',
'Predictions should follow the diagonal line.')
df_results = residual(model, features, label)
```

```
sns.lmplot(x='Actual', y='Predicted', data=df_results, fit_reg=False, height=7)
# sns.residplot(x="Actual", y="Predicted", data=df_results)
# line_coords = np.arange(df_results.min().min(), df_results.max().max())
line_coords = np.arange(df_results.iloc[:,0:2].min().min(),
df_results.iloc[:,0:2].max().max())
plt.plot(line_coords, line_coords,  # X and y points
color='darkorange', linestyle='--')
plt.title('Actual vs. Predicted')
    plt.show()
```

Question 3:

Intent: Check you problem solving approach on machine learning

Consider the dataset on Question 2. Now,

A client has solicited your services to develop a machine-learning model that can forecast the approximate value of their customers' used cars. The objective is to provide accurate quotations to customers on the price to offer for the purchase of their used cars. You have been furnished with a dataset of used cars, and your task is to:

1. conduct exploratory data analysis to identify crucial features that will be utilized in the model.
2. Please justify the selection of these features and aim to incorporate as many as possible.
3. kindly identify any potential challenges or limitations you anticipate/encounter during the feature selection process. (if any)
4. (Bonus) Try to propose a good model you feel would be able to best fit the features you have selected to make predictions.

Ans.:

```
def normal_errors_assumption(model, features, label, p_value_thresh=0.05):

print('Normality Check:', '\n')

# Calculating residuals for the Anderson-Darling test
df_results = residual(model, features, label)

print('Using the Anderson-Darling test for normal distribution')

# Performing the test on the residuals
p_value = normal_ad(df_results['Residuals'])[1]

print('p-value less than 0.05, non-normal')
print('p-value more than 0.05, normal')
print('p-value is: ', p_value)

# Plotting the residuals distribution
plt.title('Distribution of Residuals')
sns.distplot(df_results['Residuals'])
```

```
plt.show()

print()
if p_value > p_value_thresh:
print('Normally distributed')
else:
print('Not Normally distributed')
def multicollinearity(X,y,name=None):
plt.figure(figsize=(16,10))
sns.heatmap(pd.DataFrame(X,columns=name).corr(),annot=True)
VIF=[variance_inflation_factor(X,idx) for idx in range(X.shape[1])]
for i, j in enumerate(VIF):
print(f"{name[i]}----->{j}")
print(f"cases of Multicollinearity---->{sum(map(lambda x: x>10,VIF))}")
numeric_features2 = ["symboling","normalized-losses","wheel-base",
"length","width",
"height","curb-weight", "engine-size","bore","stroke","compression-
ratio","horsepower",
"peak-rpm","highway-mpg"]
num_transformed2 = num_transformed[numeric_features2]num_transformed2.head()
numeric_features3 = ["symboling","normalized-losses","wheel-base",
"length","width",
"height", "engine-size","bore","stroke","compression-ratio","horsepower",
"peak-rpm","highway-mpg"]
num_transformed3 = num_transformed[numeric_features3]num_transformed3.head()
def autocorrelation_assumption(model, features, label):
from statsmodels.stats.stattools import durbin_watson
print('Autocorrelation Check:', '\n')

# Calculating residuals for the Durbin Watson-tests
df_results = residual(model, features, label)

durbinWatson = durbin_watson(df_results['Residuals'])
print('Durbin-Watson:', durbinWatson)
if durbinWatson < 1.5:
print('Signs of positive autocorrelation', '\n')
elif durbinWatson > 2.5:
print('Signs of negative autocorrelation', '\n')
else:
print('Little to no autocorrelation', '\n')
numeric_features = ["symboling","normalized-losses","wheel-base", "length","width",
"height", "engine-size","bore","stroke","compression-ratio","horsepower",
"peak-rpm","highway-mpg"]
df1 = df[numeric_features]
# print(df1.head())
## Feature Scalingnumeric_transformer = Pipeline(
steps=[("std_scaling",StandardScaler())])
categorical_features = ["make", "fuel-type", "aspiration",'num-of-doors', 'body-style',
'drive-wheels', 'engine-location','engine-type',
'num-of-cylinders','fuel-system']
```

```
df2 = df[categorical_features]# print(df2.head())
## Categorical Feature Encodingcategorical_transformer = Pipeline(
steps=[('onehot', OneHotEncoder(drop="first",handle_unknown='ignore'))])
df = pd.concat([df1, df2], axis=1)df["price"] = df_numeric["price"]# print(df.head())
print('Number of features before encoding =
',len(numeric_features)+len(categorical_features))
preprocess=ColumnTransformer(
transformers=[ ("num", numeric_transformer, numeric_features),
("cat", categorical_transformer, categorical_features)
],
remainder="passthrough",
n_jobs=-1,
verbose=True
)
X_transformed=preprocess.fit_transform(df.iloc[:,:-1])
print('Number of features after encoding = ',X_transformed.shape[1])
lr=LinearRegression()lr_model=lr.fit(X_train,y_train)pred=lr.predict(X_test)print(f"tr
aining score--->{lr_model.score(X_train,y_train)}")print(f"testing score---
>{lr_model.score(X_test,y_test)}")
```

Question 4:

Given a string, find the length of the longest repeating subsequence, such that the two subsequences don’t have the same string character at the same position, i.e. any ith character
in the two subsequences shouldn’t have the same index in the original string.

Examples:

Input: str = "abc"
Output: 0
There is no repeating subsequence
Input: str = "aab"
Output: 1
The two subsequence are 'a'(first) and 'a'(second).
Note that 'b' cannot be considered as part of subsequence
as it would be at same index in both.
Input: str = "aabb"
Output: 2
Input: str = "axxxy"
Output: 2

```
Ans.:import java.util.Scanner;

public class Main
{
public static void main(String[] args)
{
String str;
char ch;
int strLen, i, count, j, k, repChars=0;
Scanner s = new Scanner(System.in);
System.out.print("Enter the String: ");
str = s.nextLine();

strLen = str.length();
char[] arr = new char[strLen];

for(i=0; i<strLen; i++)
arr[i] = str.charAt(i);

for(i=0; i<strLen; i++)
{
ch = arr[i];
count = 0;
for(j=(i+1); j<strLen; j++)
{
if(ch==arr[j])
{
count++;
for(k=j; k<(strLen-1); k++)
arr[k] = arr[k+1];
strLen--;
j--;
}
}
if(count>0)
repChars++;
}

System.out.println("\nTotal Number of Repeated Characters = " +repChars);
}
}
```