Here's the complete analysis and viva questions in a structured .txt format:

```

DSBDA Practical Code Analysis and Viva Questions

=================================================

File: logistic\_regression\_social\_network.py

-------------------------------------------

1. CODE ANALYSIS:

-----------------

Line-by-line Explanation:

1. import pandas as pd

- Imports Pandas library for data manipulation

2. from sklearn.linear\_model import LogisticRegression

- Imports Logistic Regression model from scikit-learn

3. from sklearn.model\_selection import train\_test\_split

- Imports train\_test\_split function for splitting dataset

4. from sklearn.metrics import confusion\_matrix,accuracy\_score,precision\_score,recall\_score

- Imports evaluation metrics for classification

5. df=pd.read\_csv(r"C:\Users\sumit\Downloads\456Social\_Network\_Ads.csv")

- Loads social network ads dataset from CSV file

6. df.head()

- Displays first 5 rows of the dataframe

7. df['Gender'].replace({"Male":0,"Female":1},inplace=True)

- Encodes Gender column (Male=0, Female=1) using label encoding

8. df.head()

- Displays first 5 rows after encoding

9. df.columns

- Shows all column names in the dataframe

10. x=df[['User ID', 'Gender', 'Age', 'EstimatedSalary']]

- Selects feature columns (independent variables)

11. y=df[['Purchased']]

- Selects target variable (dependent variable)

12. x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.25,random\_state=42)

- Splits data into 75% training and 25% testing sets with fixed random state

13. model=LogisticRegression()

- Creates Logistic Regression model instance

14. model.fit(x\_train,y\_train)

- Trains the model on training data

15. y\_pred=model.predict(x\_test)

- Makes predictions on test data

16. y\_pred

- Displays predicted values

17. model.score(x,y)

- Calculates and displays model accuracy on entire dataset

18. cf=confusion\_matrix(y\_test,y\_pred)

- Computes confusion matrix

19. cf

- Displays confusion matrix

20. tn,fp,fn,tp=confusion\_matrix(y\_test,y\_pred).ravel()

- Extracts true negatives, false positives, false negatives, true positives

21. print(tn,fp,fn,tp)

- Prints the confusion matrix components

22. accuracy\_score(y\_test,y\_pred)

- Calculates and displays accuracy score

23. precision\_score(y\_test,y\_pred)

- Calculates and displays precision score

24. recall\_score(y\_test,y\_pred)

- Calculates and displays recall score

25. acc=(tp+tn)/(tp+tn+fp+fn)

- Manual calculation of accuracy

26. print(acc)

- Prints manually calculated accuracy

2. POTENTIAL VIVA QUESTIONS:

----------------------------

Basic Concepts:

Q1. What is logistic regression?

A1. Logistic regression is a statistical model used for binary classification that predicts the probability of a binary outcome using a logistic function.

Q2. Why do we split data into training and testing sets?

A2. To evaluate model performance on unseen data and prevent overfitting.

Q3. What is the purpose of random\_state in train\_test\_split?

A3. It ensures reproducible results by fixing the random seed for data shuffling.

Code-Specific Questions:

Q4. Why did we replace Male/Female with 0/1?

A4. Machine learning algorithms require numerical input, so we performed label encoding on categorical data.

Q5. What do the components of confusion matrix represent?

A5:

- TP: Correctly predicted positive cases

- TN: Correctly predicted negative cases

- FP: Negative cases predicted as positive

- FN: Positive cases predicted as negative

Q6. How is accuracy different from precision?

A6:

- Accuracy = (TP+TN)/(TP+TN+FP+FN) - Overall correctness

- Precision = TP/(TP+FP) - Correct positive predictions among all positive predictions

Advanced Questions:

Q7. What are the assumptions of logistic regression?

A7:

- Binary outcome variable

- No multicollinearity among predictors

- Large sample size

- Linear relationship between logit of outcome and independent variables

Q8. How would you handle missing values in this dataset?

A8: Options include:

- Dropping rows with missing values

- Imputing with mean/median/mode

- Using advanced imputation techniques

Q9. What feature engineering techniques could improve this model?

A9:

- Feature scaling (standardization/normalization)

- Creating interaction terms

- Polynomial features

- Handling outliers

Q10. What alternatives exist to logistic regression for this problem?

A10:

- Decision trees

- Random forest

- SVM

- Neural networks

Syllabus-Related Questions:

Q11. Explain big data characteristics (4Vs)

A11: Volume, Velocity, Variety, Veracity

Q12. Compare supervised vs unsupervised learning

A12:

Supervised uses labeled data, unsupervised finds patterns in unlabeled data

Q13. What is MapReduce?

A13: A programming model for processing large datasets in parallel across clusters

Q14. Explain any two preprocessing techniques

A14:

- Normalization: Scaling features to [0,1] range

- Standardization: Transforming to mean=0, std=1

Q15. What is overfitting and how to prevent it?

A15:

Overfitting is when model learns training data too well including noise.

Prevention: Regularization, cross-validation, pruning, more data

```

Here’s a simplified version of the answers in easy-to-understand language with examples:

---

### \*\*DSBDA Practical Viva Questions (Simple Answers)\*\*

#### \*\*Basic Concepts:\*\*

\*\*Q1. What is logistic regression?\*\*

\*\*A1.\*\* Logistic regression is like a "yes/no" predictor. It takes input data (like age, salary) and gives the probability (0% to 100%) of an event happening (e.g., buying a product).

- \*\*Example:\*\* Predict if a customer will buy (1) or not buy (0) based on their age and salary.

---

\*\*Q2. Why split data into training and testing sets?\*\*

\*\*A2.\*\* Imagine studying for an exam:

- \*\*Training set (70-80%) = Your study notes\*\* → You learn from this.

- \*\*Testing set (20-30%) = The actual exam\*\* → Checks if you really understood.

If you only memorize notes (overfit), you’ll fail new questions. Testing ensures you actually learned.

---

\*\*Q3. What does `random\_state=42` do?\*\*

\*\*A3.\*\* It’s like fixing a dice roll to always get the same number.

- Without it: Every time you split data, you get different train/test sets (annoying for debugging).

- With it: Same split every time (reproducible results).

---

#### \*\*Code-Specific Questions:\*\*

\*\*Q4. Why replace "Male"/"Female" with 0/1?\*\*

\*\*A4.\*\* Computers don’t understand words, only numbers.

- \*\*Male = 0, Female = 1\*\* → Now the model can use gender in calculations.

- \*\*Alternative:\*\* One-hot encoding (e.g., Male=[1,0], Female=[0,1]).

---

\*\*Q5. What is a confusion matrix?\*\*

\*\*A5.\*\* A report card for your model:

| | Predicted \*\*No (0)\*\* | Predicted \*\*Yes (1)\*\* |

|----------------|----------------------|-----------------------|

| \*\*Actual No (0)\*\* | ✅ Correct (TN) | ❌ Wrong (FP) |

| \*\*Actual Yes (1)\*\* | ❌ Wrong (FN) | ✅ Correct (TP) |

- \*\*TN = 50\*\* (Correctly predicted "No")

- \*\*FP = 10\*\* (Said "Yes" but was "No" → False alarm)

- \*\*FN = 5\*\* (Said "No" but was "Yes" → Missed opportunity)

- \*\*TP = 35\*\* (Correctly predicted "Yes")

---

\*\*Q6. Accuracy vs. Precision?\*\*

\*\*A6.\*\*

- \*\*Accuracy = % of total correct guesses\*\*

`= (TN + TP) / Total = (50 + 35) / 100 = 85%`

- \*\*Precision = % of "Yes" predictions that were right\*\*

`= TP / (TP + FP) = 35 / (35 + 10) = 77.7%`

- \*\*Recall = % of actual "Yes" cases caught\*\*

`= TP / (TP + FN) = 35 / (35 + 5) = 87.5%`

\*\*Example:\*\*

- \*\*High precision (but low recall):\*\* A spam filter that rarely marks good emails as spam (few FP) but misses many actual spams (many FN).

---

#### \*\*Advanced Questions:\*\*

\*\*Q7. What if data has missing values?\*\*

\*\*A7.\*\* Options:

1. \*\*Delete rows\*\* (if few missing values).

2. \*\*Fill with average\*\* (e.g., mean age).

3. \*\*Predict missing values\*\* (using other data).

\*\*Example:\*\* If "Salary" is missing, fill it with the average salary of similar users.

---

\*\*Q8. How to improve the model?\*\*

\*\*A8.\*\* Try:

1. \*\*Scale features\*\* (e.g., salary from 0 to 1).

2. \*\*Add interactions\*\* (e.g., "Age × Salary").

3. \*\*Remove useless features\*\* (e.g., "User ID").

---

\*\*Q9. Alternatives to logistic regression?\*\*

\*\*A9.\*\*

1. \*\*Decision Trees\*\* (like a flowchart of questions).

2. \*\*Random Forest\*\* (many decision trees voting).

3. \*\*Neural Networks\*\* (for complex patterns).

---

#### \*\*Syllabus-Related:\*\*

\*\*Q10. What is big data (4Vs)?\*\*

\*\*A10.\*\*

1. \*\*Volume\*\* (huge data, e.g., Facebook posts).

2. \*\*Velocity\*\* (fast data, e.g., live tweets).

3. \*\*Variety\*\* (different types, e.g., text + images).

4. \*\*Veracity\*\* (trustworthiness, e.g., fake news vs. real).

---

\*\*Q11. Supervised vs. Unsupervised Learning?\*\*

\*\*A11.\*\*

- \*\*Supervised (with labels):\*\* Teacher gives answers (e.g., spam/not spam).

- \*\*Unsupervised (no labels):\*\* Find hidden patterns (e.g., grouping similar customers).

---

\*\*Q12. What is overfitting?\*\*

\*\*A12.\*\* When your model \*\*memorizes\*\* the training data but fails on new data.

- \*\*Example:\*\* A student who memorizes answers but can’t solve new questions.

- \*\*Fix:\*\* Simplify the model, use more data, or cross-validation.

---

### \*\*Summary:\*\*

- Logistic regression = Yes/No predictor.

- Train/test split = Study notes vs. real exam.

- Confusion matrix = Model’s report card.

- Big data = Huge, fast, mixed, and messy data.

This version avoids jargon and uses everyday examples. Let me know if you'd like any part simplified further!