Viva Questions and Answers for All 10 ML Lab Experiments

Experiment 1: Basics of NumPy and Arrays

Q1: What is NumPy, and why is it used in Python?

A: NumPy is a Python library for numerical computing, used for working with arrays (ndarray). It provides efficient array operations, mathematical functions, and is widely used in data science and ML due to its speed and ease of handling large datasets.

Q2: How is a NumPy array created in the code?

A: In the code, arrays are created using `np.array()`. For example, `array_from_list = np.array([1, 2, 3, 4, 5])` creates a 1D array from a list, and `array_2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])` creates a 2D array from a nested list.

Q3: What does the `shape` attribute of a NumPy array represent?

A: The `shape` attribute returns a tuple indicating the dimensions of the array (e.g., `(5,)` for a 1D array with 5 elements, or `(3, 3)` for a 3x3 2D array). In the code, `array_2d.shape` outputs `(3, 3)`.

Q4: What operations were performed on arrays in the code?

A: The code performs element-wise addition (`array1 + array2`), multiplication (`array1 * array2`), and statistical operations like `np.mean()`, `np.sum()`, `np.max()`, and `np.min()` on `array_from_list`.

Q5: How is the 2D array visualized in the code?

A: The 2D array is visualized as a heatmap using `plt.imshow(array_2d, cmap='Blues')` with `plt.text()` to display values, and a colorbar is added with `plt.colorbar()`.

Experiment 2: Linear Regression

Q1: What is the purpose of linear regression?

A: Linear regression predicts a continuous dependent variable based on one or more independent variables, modeling the relationship as a linear equation (`y = c + bx`).

Q2: What dataset was used in the code, and how was it described?

A: The code uses the Diabetes dataset, with one feature (BMI) for simplicity. The dataset info is printed using `df.shape` (442 rows, 2 columns) and `df.head()` for the first 5 rows.

Q3: How is the linear regression model implemented in the code?

A: The model is implemented using `LinearRegression()` from scikit-learn. It is trained with `model.fit(X, y)` and used to predict with `model.predict(X)`.

Q4: What performance metrics are calculated in the code?

A: The code calculates Mean Squared Error (`mean_squared_error(y, y_pred)`) and R-squared score (`r2_score(y, y_pred)`) to evaluate the model's performance.

Q5: How is the regression line visualized?

A: A scatter plot of BMI vs. Disease Progression is created with `plt.scatter()`, and the regression line is plotted with `plt.plot(X, y_pred, color='red')`.

Experiment 3: Naive Bayes

Q1: What is Naive Bayes, and what is its key assumption?

A: Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming that features are conditionally independent given the class label.

Q2: What dataset was used, and how was it generated?

A: A synthetic dataset for customer purchase prediction (Age, Income, Purchase) was created with 1000 rows and 3 columns using `np.random.normal()` and logical conditions for the target.

Q3: Which Naive Bayes variant was used in the code, and why?

A: Gaussian Naive Bayes (`GaussianNB`) was used because the features (Age, Income) are continuous and assumed to follow a normal distribution.

Q4: What outputs are printed in the code?

A: The code prints dataset info (`df.shape`, `df.head()`), accuracy (`accuracy_score()`), and a classification report (`classification_report()` with precision, recall, F1-score).

Q5: What visualizations are included?

A: A confusion matrix heatmap (`sns.heatmap(cm, annot=True)`) and a scatter plot (`sns.scatterplot()`) of Age vs. Income colored by Purchase status are included.

Experiment 4: Dimensionality Reduction (PCA)

Q1: What is the goal of dimensionality reduction?

A: Dimensionality reduction reduces the number of features while retaining most of the data's information, simplifying models and visualization.

Q2: What dataset was used in the code?

A: A synthetic student performance dataset (Math Score, Science Score, English Score, Study Hours) with 1000 rows and 4 columns was used, created with `np.random.normal()`.

Q3: How is PCA implemented in the code?

A: PCA is implemented using `PCA(n_components=2)` from scikit-learn, applied to standardized data (`StandardScaler`) to reduce 4 features to 2 principal components.

Q4: What does `explained_variance_ratio_` represent?

A: It shows the proportion of variance explained by each principal component. The code prints this and the total explained variance (`sum(pca.explained_variance_ratio_)`).

Q5: How is PCA visualized?

A: A scatter plot of PC1 vs. PC2 is created with `sns.scatterplot()`, with axis labels showing variance percentages (`pca.explained_variance_ratio_ * 100`).

Experiment 5: Neural Network

Q1: What is the purpose of a neural network?

A: Neural networks model complex relationships in data for tasks like classification or regression by mimicking biological neurons with layers of interconnected nodes.

Q2: What dataset was used, and what does it represent?

A: A synthetic dataset for customer purchase behavior (Age, Income, Browsing Time, Purchase) with 1000 rows and 4 columns was created using `np.random.normal()`.

Q3: How is the neural network structured in the code?

A: The code uses a Keras `Sequential` model with three layers: 16 neurons (ReLU), 8 neurons (ReLU), and 1 neuron (sigmoid) for binary classification.

Q4: What metrics are evaluated in the code?

A: The code evaluates test accuracy using `accuracy_score(y_test, y_pred)` after predicting with `model.predict(X_test)`.

Q5: What visualizations are provided?

A: A loss curve (`plt.plot(history.history['loss'])`) for training and validation loss and a confusion matrix heatmap (`sns.heatmap(cm, annot=True)`) are included.

Experiment 6: Decision Tree

Q1: What is a decision tree, and how does it work?

A: A decision tree is a flowchart-like model where internal nodes test attributes, branches represent outcomes, and leaf nodes indicate class labels, used for classification or regression.

Q2: What dataset was used in the code?

A: A synthetic dataset for employee promotion eligibility (Years of Experience, Performance Score, Education Level, Promoted) with 1000 rows and 4 columns was used.

Q3: How is the decision tree implemented?

A: The code uses `DecisionTreeClassifier(max_depth=3)` from scikit-learn, trained with `model.fit(X_train, y_train)`.

Q4: What outputs are shown?

A: The code prints dataset info (`df.shape`, `df.head()`), accuracy (`accuracy_score()`), and a classification report (`classification_report()`).

Q5: How is the decision tree visualized?

A: The tree structure is plotted with `plot_tree(model, feature_names=X.columns)` and a confusion matrix heatmap is shown with `sns.heatmap(cm, annot=True)`.

Experiment 7: Random Forest

Q1: What is Random Forest, and how does it differ from a single decision tree?

A: Random Forest is an ensemble method that builds multiple decision trees on random subsets of data and features, aggregating their predictions to improve accuracy and reduce overfitting.

Q2: What dataset was used?

A: A synthetic dataset for loan approval decisions (Credit Score, Income, Loan Amount, Approved) with 1000 rows and 4 columns was created.

Q3: How is Random Forest implemented in the code?

A: The code uses `RandomForestClassifier(n_estimators=100, max_depth=3)` from scikit-learn, trained with `model.fit(X_train, y_train)`.

Q4: What does `feature_importances_` represent?

A: It shows the relative importance of each feature in the Random Forest model. The code plots this using `sns.barplot()`.

Q5: What visualizations are included?

A: A feature importance bar plot (`sns.barplot()`) and a confusion matrix heatmap (`sns.heatmap(cm, annot=True)`) are included.

Experiment 8: K-Means Clustering

Q1: What is K-Means clustering, and when is it used?

A: K-Means is an unsupervised learning algorithm that groups unlabeled data into K clusters based on feature similarity, used for discovering patterns in data.

Q2: What dataset was used in the code?

A: A synthetic dataset for customer shopping behavior (Annual Spending, Visit Frequency) with 1000 rows and 2 columns was created to form natural clusters.

Q3: How is K-Means implemented in the code?

A: The code uses `KMeans(n_clusters=3)` from scikit-learn, applied to standardized data with `kmeans.fit_predict(X_scaled)`.

Q4: How is the optimal number of clusters determined?

A: The elbow method is used, plotting inertia (`kmeans.inertia_`) for K=1 to 10 to identify the “elbow” point.

Q5: What visualizations are provided?

A: A scatter plot of clusters with centroids (`sns.scatterplot()` and `plt.scatter()`) and an elbow plot (`plt.plot(K_range, inertia)`) are included.

Experiment 9: Principal Component Analysis (PCA)

Q1: What are Principal Components?

A: Principal Components are uncorrelated variables derived from the original features, capturing the maximum variance in the data. The first component explains the most variance, followed by subsequent components.

Q2: How is PCA used in social sciences?

A: In social sciences, PCA is used to reduce the dimensionality of datasets (e.g., survey responses) to identify underlying patterns or factors, such as grouping related questionnaire items.

Q3: What dataset was used in the code?

A: A synthetic dataset for employee performance (Productivity Score, Team Collaboration, Task Completion, Overtime Hours) with 1000 rows and 4 columns was used.

Q4: How is standardization handled in the code?

A: The code uses `StandardScaler()` to standardize features before PCA, ensuring all features have zero mean and unit variance (`X_scaled = scaler.fit_transform(df)`).

Q5: What visualizations are included?

A: A scatter plot of PC1 vs. PC2 (`sns.scatterplot()`) and a bar plot of explained variance ratios (`plt.bar()`) are included.

Experiment 10: Support Vector Machine (SVM)

Q1: What is SVM?

A: SVM is a supervised learning algorithm for classification that finds a hyperplane in a high-dimensional space to separate classes with the maximum margin.

Q2: What is the objective of SVM?

A: The objective of SVM is to maximize the margin between the separating hyperplane and the nearest data points (support vectors) to improve classification accuracy.

Q3: What dataset was used in the code?

A: A synthetic dataset for customer churn prediction (Monthly Charges, Tenure, Satisfaction Score, Churn) with 1000 rows and 4 columns was used.

Q4: Why is a linear kernel used in the code?

A: A linear kernel (`SVC(kernel='linear')`) is used for simplicity and because the synthetic data is designed to be linearly separable after standardization.

Q5: What visualizations are provided?

A: A scatter plot with the decision boundary for two features (`plt.contourf()` and `sns.scatterplot()`) and a confusion matrix heatmap (`sns.heatmap(cm, annot=True)`) are included.