Predictive Analytics & Data Science Program

Advanced 5-Day Training Programme - Data Science and Predictive Analytics 5 Days | 8 Hours/Day | 40 Hours Total

Tools Used: Python, pandas, numpy, matplotlib, seaborn, scikit-learn, XGBoost, SHAP, statsmodels, SQLite, Excel

Content Coverage :

Day 1: Advanced Data Wrangling and Exploratory Data Analysis

Module 1: Advanced Data Cleaning and Transformation with Pandas

- Handling missing data at scale: advanced imputation strategies (mean, median, mode, KNN imputation) and when to drop vs fill
- Reshaping data: pivot_table, melt, stack, unstack for converting between wide and long formats
- Merging and joining multiple datasets: inner, outer, left, right joins and handling duplicate keys
- Working with datetime columns: parsing, resampling, time-based grouping, and lag/lead features

Lab Exercises:

1. Load a messy real-world dataset and apply a full cleaning pipeline using pandas.
2. Merge multiple datasets, reshape using pivot_table, and produce a summary DataFrame grouped by month and category.

Module 2: Exploratory Data Analysis and Statistical Profiling

- Univariate analysis: distributions, skewness, kurtosis, and outlier detection
- Bivariate and multivariate analysis: correlation matrices, pair plots, covariance analysis
- Visualising distributions and relationships: histograms, box plots, violin plots, heatmaps
- Feature profiling: identifying high-cardinality columns, zero-variance features, and highly correlated predictors

Lab Exercises:

1. Perform a full EDA on a dataset, compute descriptive statistics, plot distributions, and build a correlation heatmap.
2. Detect and handle outliers using IQR and Z-score methods, visualise before/after distributions, and document impact.

Day 2: SQL for Data Analysis and Feature Engineering

Module 3: Advanced SQL for Data Extraction and Analysis

- Window functions: ROW_NUMBER, RANK, DENSE_RANK, LAG, LEAD, running totals
- Common Table Expressions (CTEs) and recursive queries
- Aggregation and grouping patterns: ROLLUP, CUBE, GROUPING SETS
- Integrating SQL with Python using sqlite3 and pandas

Lab Exercises:

1. Write queries using window functions to calculate running totals, rank customers, and find growth rates.
2. Connect to SQLite from Python, run a CTE-based query, and produce a summary chart.

Module 4: Feature Engineering for Machine Learning

- Encoding categorical variables: one-hot, label, target, ordinal encoding
- Numerical transformations: log, Box-Cox, binning, polynomial features
- Creating interaction and domain-driven features
- Feature scaling: StandardScaler, MinMaxScaler, RobustScaler

Lab Exercises:

1. Engineer new features from a raw dataset, document reasoning.
2. Build a full feature engineering pipeline using scikit-learn ColumnTransformer and Pipeline.

Day 3: Supervised Machine Learning: Regression and Classification

Module 5: Regression Models

- Linear regression: assumptions, OLS estimation, interpreting coefficients
- Polynomial regression and interaction terms
- Regularisation: Ridge (L2) and Lasso (L1) regression
- Evaluation: R-squared, MAE, MSE, RMSE, residual analysis

Lab Exercises:

1. Build a linear regression model, check assumptions, interpret coefficients.
2. Apply Ridge and Lasso regression, tune hyperparameters, compare performance.

Module 6: Classification Models

- Logistic regression for binary/multi-class classification
- Decision trees: splitting criteria, depth control, visualisation
- Random forests: ensemble logic, feature importance, hyperparameter tuning

- Evaluation: confusion matrix, precision, recall, F1, ROC-AUC, handling imbalance

Lab Exercises:

1. Build logistic regression and decision tree classifiers, compare metrics.
2. Train a random forest, tune parameters, evaluate with ROC curve and AUC.

Day 4: Advanced ML: Gradient Boosting, Clustering, and Model Validation

Module 7: Gradient Boosting Models

- Gradient boosting intuition
- XGBoost with Python: parameters, early stopping
- Hyperparameter tuning with grid search
- SHAP values for explainability

Lab Exercises:

1. Train an XGBoost classifier, tune hyperparameters, compare with random forest.
2. Use SHAP to explain predictions, plot summary and waterfall plots.

Module 8: Unsupervised Learning

- K-Means clustering: elbow method, silhouette score
- Hierarchical clustering: dendrograms, linkage methods
- PCA: dimensionality reduction, explained variance, visualisation
- Practical clustering workflow

Lab Exercises:

1. Apply K-Means clustering, profile clusters, label with business meaning.
2. Apply PCA, plot reduced data, compute explained variance ratio.

Day 5: Model Evaluation, Time Series, and Capstone Project

Module 9: Robust Model Validation

- Cross-validation strategies: k-fold, stratified, time-series split
- Data leakage: prevention with Pipelines
- Saving/loading models with joblib
- Building reproducible ML workflows

Lab Exercises:

1. Wrap preprocessing and model into a Pipeline, run stratified k-fold CV, check for leakage.
2. Save and reload trained pipeline, run predictions, simulate deployment.

Module 10: Time Series Analysis and Capstone Project

- Fundamentals: trend, seasonality, stationarity, autocorrelation
- Forecasting with ARIMA models
- Visualisation with pandas and matplotlib
- Capstone: end-to-end data science project

Lab Exercises:

1. Fit ARIMA model on time series data, plot forecasts.
2. Complete a capstone project: EDA, feature engineering, model training, evaluation, and presentation.