**Customer Churn Prediction Analysis Plan**

**Objective**

Analyze the Telco Customer Churn dataset to predict customer churn and identify key factors driving churn, using Principal Component Analysis (PCA) for dimension reduction and K-Nearest Neighbors (KNN) and Random Forest for classification, with logistic regression for comparison. The results will inform retention strategies for a telecommunications company.

**Dataset**

* **Source**: Telco Customer Churn dataset ("WA\_Fn-UseC\_-Telco-Customer-Churn.csv").
* **Description**: Contains customer demographics, service usage, billing details, and a binary churn label (Yes/No), with approximately 7,043 observations and 21 features.

**Software and Libraries**

* **Environment**: Python in Visual Studio Code.
* **Libraries**:
  + pandas: Data manipulation and loading.
  + numpy: Numerical operations.
  + scikit-learn: PCA, KNN, Random Forest, logistic regression, and cross-validation.
  + matplotlib and seaborn: Visualizations (e.g., scree plots, confusion matrices).
  + scipy: Statistical computations.

**Step-by-Step Analysis Plan**

**Step 1: Data Loading and Preprocessing**

* **Objective**: Load the dataset and prepare it for analysis.
* **Tasks**:
  + Load the CSV file using pandas.read\_csv.
  + Inspect the dataset for missing values, data types, and summary statistics.
  + Handle missing values (e.g., impute numerical features with median, categorical with mode, or remove rows if minimal).
  + Encode categorical variables (e.g., gender, InternetService) using one-hot encoding (pandas.get\_dummies).
  + Convert the Churn column to binary (0 for No, 1 for Yes).
  + Standardize numerical features (e.g., tenure, MonthlyCharges) using StandardScaler to ensure compatibility with PCA and KNN.
* **Output**: Cleaned and preprocessed dataset ready for analysis.

**Step 2: Exploratory Data Analysis (EDA)**

* **Objective**: Understand the dataset’s structure and generate descriptive statistics for the report’s introduction.
* **Tasks**:
  + Compute summary statistics (e.g., mean tenure, churn rate) using pandas.describe.
  + Visualize distributions of key features (e.g., tenure, MonthlyCharges) with histograms or boxplots.
  + Create a bar plot of churn rates by categorical features (e.g., Contract, InternetService) using seaborn.countplot.
  + Generate a correlation matrix heatmap for numerical features to identify potential relationships.
* **Output**: Tables and plots (e.g., churn rate by contract type) for the report’s introduction and appendix.

**Step 3: Dimension Reduction with PCA**

* **Objective**: Reduce the dimensionality of numerical features to capture key patterns and aid visualization.
* **Tasks**:
  + Select numerical features (e.g., tenure, MonthlyCharges, TotalCharges) and standardized features from one-hot-encoded categorical variables.
  + Apply PCA using sklearn.decomposition.PCA to transform the feature space.
  + Compute the Proportion of Variance Explained (PVE) for each principal component.
  + Create a scree plot to determine the number of components explaining significant variance (e.g., 80–90% of total variance).
  + Visualize the first two principal components in a scatter plot, colored by churn status, to identify patterns.
* **Output**: Scree plot, PVE table, and biplot for the report’s Methods & Results section and appendix.

**Step 4: Classification with KNN, Random Forest, and Logistic Regression**

* **Objective**: Predict churn using classification methods and compare performance.
* **Tasks**:
  + **Data Preparation**:
    - Split the preprocessed dataset into training (80%) and test (20%) sets using sklearn.model\_selection.train\_test\_split.
    - Optionally, use PCA-transformed features (from Step 3) for classification to reduce dimensionality.
  + **KNN**:
    - Implement KNN using sklearn.neighbors.KNeighborsClassifier.
    - Use k-fold cross-validation (sklearn.model\_selection.cross\_val\_score, k=5) to select the optimal number of neighbors (e.g., test k=3, 5, 7, 10).
    - Train the model on the training set and predict on the test set.
  + **Random Forest**:
    - Implement Random Forest using sklearn.ensemble.RandomForestClassifier.
    - Tune hyperparameters (e.g., number of trees, max depth) using cross-validation.
    - Train the model and extract feature importance scores.
  + **Logistic Regression**:
    - Implement logistic regression using sklearn.linear\_model.LogisticRegression for comparison.
    - Train the model and compute coefficient estimates.
  + **Evaluation**:
    - Compute performance metrics for each model on the test set: accuracy, sensitivity (true positive rate), specificity (true negative rate), and misclassification rate.
    - Generate confusion matrices using sklearn.metrics.confusion\_matrix.
    - Create ROC curves and compute AUC scores using sklearn.metrics.roc\_curve and sklearn.metrics.roc\_auc\_score.
    - Perform k-fold cross-validation to estimate test error for each model.
* **Output**: Confusion matrices, ROC curves, feature importance plots (Random Forest), and performance metric tables for the report’s Methods & Results section.

**Step 5: Interpretation and Business Implications**

* **Objective**: Interpret results and link findings to business strategies.
* **Tasks**:
  + Analyze PCA results to identify which features (e.g., high MonthlyCharges, short tenure) contribute most to principal components associated with churn.
  + Compare classification model performance (e.g., Random Forest vs. KNN vs. logistic regression) based on accuracy, sensitivity, and AUC.
  + Highlight key predictors of churn from Random Forest feature importance (e.g., Contract, tenure).
  + Propose business strategies (e.g., target short-tenure customers with retention offers) based on findings.
* **Output**: Narrative for the report’s Conclusion section, linking statistical results to actionable recommendations.

**Step 6: Report and Presentation Preparation**

* **Objective**: Compile findings into a report and presentation per the project guidelines.
* **Tasks**:
  + **Report Structure** (6 pages main content, 1 cover page, ≤6 pages appendix):
    - **Cover Page**: Include project title, group members, group number, date, and executive summary (≤150 words summarizing key findings, e.g., “Random Forest achieved 85% accuracy in predicting churn, with tenure and contract type as key predictors”).
    - **Introduction**: Describe the dataset, problem (predicting churn), and business relevance, including EDA results (e.g., churn rate by contract type).
    - **Methods & Results**: Detail PCA (scree plot, PVE), classification methods (KNN, Random Forest, logistic regression), and performance metrics (tables, confusion matrices, ROC curves).
    - **Conclusion**: Summarize findings and business implications (e.g., retention strategies).
    - **Appendix**: Include detailed outputs (e.g., full PCA loadings, cross-validation results, additional plots).
  + **Presentation** (15 minutes):
    - Create slides summarizing the problem, dataset, methods, key results (e.g., PCA biplot, Random Forest feature importance), and business recommendations.
    - Use visuals like scree plots, confusion matrices, and ROC curves to engage the audience.
  + **Code Submission**: Submit Python scripts with comments explaining each step.
* **Output**: Report (PDF), presentation slides (PDF or PPT), and Python code files.

**Expected Outputs**

* **Report**: Tables (summary statistics, PVE, classification metrics), plots (scree plot, biplot, confusion matrices, ROC curves, feature importance), and narrative linking results to business strategies.
* **Presentation**: Slides with key visuals and concise explanations of methods and findings.
* **Code**: Python script with data preprocessing, PCA, classification, and visualization steps.

**Timeline (Assuming Deadlines: Slides by March 13, Report by March 17, 2025)**

* **Week 1**: Data loading, preprocessing, and EDA.
* **Week 2**: PCA implementation and visualization.
* **Week 3**: Classification (KNN, Random Forest, logistic regression) and performance evaluation.
* **Week 4**: Result interpretation, report drafting, and presentation preparation.
* **Week 5**: Finalize report, slides, and code for submission.

**Notes**

* Ensure reproducibility by setting random seeds (e.g., np.random.seed(42)) for train-test splits and model training.
* Place large outputs (e.g., PCA loadings, full confusion matrices) in the appendix to stay within the 6-page limit.
* Use cross-validation to robustly estimate model performance, as emphasized in Lecture 2b.
* Save plots as high-resolution images for the report and presentation.