ATM-Machine Project Algorithm:

1. Define an Account class with the following attributes:
    a. name: account holder's name
    b. pin: account holder's PIN
    c. balance: account balance
    d. transaction_history: list of transactions
2. Create an init function to initialize the attributes of the account object.
3. Define a deposit function to deposit the amount into the account balance.
    a. Add the amount to the balance
    b. Append the transaction history with a deposit message and the amount.
    c. Print the deposit amount and new balance.
4. Define a withdraw function to withdraw the amount from the account balance.
    a. Check if the amount to be withdrawn is greater than the account balance.
    b. Check if the amount to be withdrawn is greater than the allowed limit of $500.
    c. Subtract the amount from the account balance.
    d. Append the transaction history with a withdrawal message and the amount.
    e. Print the withdrawal amount and new balance.
5. Define a check_balance function to print the account balance.
6. Define a check_history function to print the transaction history.
7. Define a main function to interact with the user.
    a. Ask the user for their name, pin and starting balance.
    b. Create an account object with the given information.
    c. Display the menu with options to deposit, withdraw, check balance, check history, and exit.
    d. Take user input for the chosen option and execute the corresponding function.
    e. If the number of transactions exceeds 5, break the loop and display the withdrawal limit message.
    f. Exit the loop when the user chooses to exit.
8. Call the main function if the script is being executed as the main program.
9. End of Algorithm.

Theoretical Analysis:

The code defines a class '**Account**' that represents a bank account. It has four instance variables: '**name**' (the name of the account holder), '**pin**' (the PIN of the account holder), '**balance**' (the current balance of the account), and '**transaction_history**' (a list to store the transaction history of the account).

The '**__init__**' method is called when a new '**Account**' object is created. It takes three parameters: '**name**' , '**pin**' , and '**balance**' . It sets these parameters as instance variables and initializes the '**transaction_history**' list as an empty list.

The '**deposit**' method takes an amount as a parameter and adds it to the account balance. It also appends a string describing the transaction to the '**transaction_history**' list. Finally, it prints the new balance of the account.

The '**withdraw**' method takes an amount as a parameter and subtracts it from the account balance if the amount is less than or equal to the account balance and less than or equal to the withdrawal limit of $500. If the withdrawal limit is exceeded, it prints an appropriate message. If there are insufficient funds, it prints another message. Finally, it appends a string describing the transaction to the '**transaction_history**' list.

The '**check_balance**' method simply prints the current balance of the account.

The '**check_history**' method prints the transaction history of the account by iterating over the '**transaction_history**' list and printing each transaction.

The '**main**' function is the entry point for the program. It first prompts the user to enter their name, PIN, and starting balance, and creates a new '**Account**' object with these values. It then enters a loop that allows the user to perform one of five actions: deposit, withdraw, check balance, check transaction history, or exit. It prompts the user to enter a choice and performs the appropriate action based on the user's choice. If the user's transaction history exceeds five transactions, they are notified that the withdrawal limit has been reached and the program exits.

Overall, the code allows a user to interact with an ATM-like system to manage their account. They can deposit or withdraw funds, check their balance, and view their transaction history. The withdrawal limit prevents excessive withdrawals and the transaction history helps the user keep track of their account activity.